The Cloudcast (00:01.263)
Good morning, good evening, WebVR, and welcome back to the Cloudcast. We're coming to you live from our massive Cloudcast studios here in Raleigh, North Carolina. And I have both a kind of returning guest from a company standpoint, as well as an ex-co-worker as well. And so we're gonna dig into all of this. We're gonna do really kind of a state of cloud native for mid 2025. And for that, we have Toby Canup.

VP and GM of Cloud Native at Nutanix. So first of all, Toby, welcome to the show and give everyone a quick introduction.

Tobi Knaup (00:35.618)
Hey everybody, thanks so much for having me on. Yeah, my name is Tobi Knell. Work at Nutanix now. I joined the company about a year and a half ago when Nutanix acquired the startup that I co-founded, which some of you might know. It was called Mesosphere. We later changed the name to Day2IQ. And Mesosphere has been around since 2013. We've been doing cloud native stuff for a very long time.

we launched the company around an open source project called Apache Mesos. That's why it was called Mesosphere. And it's, you know, that's a container orchestration platform that predates Kubernetes by about five years. So did that for, you know, the first five or six years of the company. And then, you know, as Kubernetes showed up on the scene and became sort of the industry standard, we pivoted to Kubernetes. So been doing that for a very long time too. And yeah, excited to be here.

The Cloudcast (01:32.38)
Fantastic. Yeah. And, and yeah, full disclosure, cause I kind of mentioned it, you know, I actually worked for Toby at Nutanix, once, once upon a time. and like I said, this show is a bit of a throwback for our long-term listeners. Cause we actually had your co-founder, Ben Hindman on episode two 11. So to give everyone an idea that's almost 10 years ago, when day two IQ was Mesosphere. And we also had

Dave Lester, who was at Twitter back in 2014 as well. And so I'm not gonna ask you to, hey, catch everyone up on the last 10 years of history, because I think that we might be here all day if we did that. But I will say this, I will encourage everyone to go back and listen to this, because I actually re-listened to that one recently. That episode 211, it's an excellent snapshot of...

Tobi Knaup (02:13.71)
That's right.

The Cloudcast (02:24.751)
I'll call it the early days of cloud native and containers. Like as you mentioned, you know, there was Mesos and there was Kubernetes and it was really, really interesting to go back and kind of think about that and think about where we were in the state of the industry. And there's also a lot of lessons that you can kind of pull forward into that. And that's what I kind of wanted to talk

about with you Toby is, you know, with the most recent Kube County, you, gosh, was, you know, it'll be a few months ago now and,

So like, what is your thoughts around KubeCon, CNCF, and kind of the current state of the cloud native in the industry, if you will. Let's kind of start at a highlight.

Tobi Knaup (03:07.83)
Yeah, so I did go to KubeCon London and it was the biggest KubeCon ever. And the first time actually that KubeCon EU was larger than the US version. that was quite a big show. It always feels like the whole industry comes together at these events. And it was awesome to see the continued momentum. It's, for me having...

been part of cloud native for such a long time. And, you know, it's kind of this weird feeling where you've been doing this for over a decade. like in a lot of ways, you're like, this is old now, right? Like we've been doing this forever, but that's not the experience that everybody has, right? For most people, it's still new technology. So if you look at mainstream adoption, enterprise adoption of Kubernetes and other cloud native technologies, it's still early days, right?

And so you look at, what's the percentage of workloads that are running in containers, for example, at an enterprise, it's going to be a small number, right? And so when you look at the state of things, I think it's fair to say that it continues, all these cloud native technologies continue their march to become mainstream and to adopt more mainstream enterprise workloads in the containers.

And that, think, was definitely a theme at KubeCon is when you look at what's new, what are the new technologies, the new offerings from the vendors on the show floor. I would say the common theme was all these new things were around making it a resilient platform, making it scalable, being this rock solid foundation for ever more mission critical workloads.

The Cloudcast (04:54.041)
Yeah, yeah. And actually, that's a perfect segue into our next question here, because I've been meaning to ask you this because, as I mentioned, I don't dabble in the cloud native space as much as I used to now. But I feel like the talk track, like for those that have been around for a while, it kind of sort of hasn't changed a whole lot over the years. And let me explain what I mean by that. I think Kubernetes at times

It just keeps getting in its own way. Because you keep hearing it's complex and you hear it's it's more about, you know, the classic example was it's a platform to build platforms. But as you mentioned, as we, it matures, as we get more people into the space, as we get newer folks into the space, all of that has to evolve. And so tell everyone a little bit about, has it changed? And if it has changed, what's that evolution looking?

Tobi Knaup (05:53.986)

Yeah. So I do like that, you know, a platform to build platforms line. think Joe Bader probably first said it. He called Kubernetes a platform platform. And that is true, right? If you look at just the APIs in Kubernetes, they are very low level APIs, right? They're low level stuff to continue to control deployments and volumes and your networking, all these low level constructs. And most people will need some higher level

layer on top of that, that gives them something more application centric or closer to the business needs. And I think there is actually a lot of opportunity right now and a lot of changes happening, also driven by AI that can help people make this more easy to use. So I've always been looking for a layer on top of Kubernetes where I can express

my business needs or just my application centric desire. And then there's something that translates it to Kubernetes API calls because, know, just pick a very concrete example. If I want to launch an app, right? And I have a certain SLO for this app. Like let's say I want, you know, the P99 response time to be under 50 milliseconds. And I want to give it four nines or five nines or something like that.

Taking that business requirement and translating it into all the yaml that you need to put out there to make it so, that's super, super difficult for a human to do. Even for a Kubernetes expert, you have to do some thinking like how do I exactly do this? But if you think about it, we have all the pieces in the system to actually build some kind of automation, some kind of software that makes it so. We now have fantastic observability stacks where I can monitor P99.

where I can look at uptime. And then, you know, we have all the APIs there to deploy more instances, to deploy it in more availability zones, more geographic locations, to get to that response time target. And so, you know, what I would really like to see is AI doing that work for me, right?

Tobi Knaup (08:18.316)
the same way people are vibe coding right now and just saying like, you know, I want to build this website, it should do X, Y, Z, right? Just do it for me. That's, I think to me, that's the same kind of pattern, right? Like I have a high level idea that I express in natural language and then I have a system that translates it into code or API calls or YAML configuration, right? So, you know,

That's something I would really like to see. think that'd be a great use of AI to make Kubernetes easier because that is still the number one thing that's keeping it back. There is a steep learning curve. It's a complex system. It's pretty low level. So if you're just someone who wants to deploy an app, there's a lot of learning and there's a lot of folks out there that are trying to make that easier now.

The Cloudcast (09:05.625)
Yeah, yeah. And it's funny too, because I mean, I'm going to describe a visual which is the worst for podcasting there possibly is. But for you and I, know, when we were working together, Day 2

IQ had this awesome slide, you know, we always call it the wheel slide, right? Of like, here's Day 2 IQ at the center, and then here's all of the various projects and plugins and everything that can go around. Because, and the reason why I really like that, and that analogy, and when we're talking about kind of design patterns,

is this whole idea of I think a lot of us in the industry tend to think in stacks. yes, while there is absolutely we're building stacks, but this wheel analogy and this wheel diagram, I'll see if I can like put a screen, you know, a screenshot of it in the show notes or something like that if everyone wants to check it out. But the whole idea of like, by the way, there's all of these other things like, know, the networking technologies and the multi cloud technologies and the disk technologies and observability. Like there's all these things that goes into yes, it might.

eventually be building a stack. But I think the wheel analogy also really opened everyone's eyes of like, wow, that is like so much more I have to do at times. Like it's not just, hey, here's the one product, right? It's all these building pieces, right? And it kind of would go out. So go ahead, Toby.

Tobi Knaup (10:16.054)
Yeah, absolutely.

That's right.

Tobi Knaup (10:25.454)
Yeah, and what's going to add to that, there's I think multiple ways in which complexity keeps increasing in the space, right? Kubernetes itself is fairly stable now, right? And it has been for some time in terms of its capabilities and its APIs. The innovation is really happening around it, right? And that's where also a lot of the complexity gets added. A few years ago, we added a service mesh.

That's a pretty complex technology. It's super powerful. It's super important, but very complex. And so you get complexity added from, you know, these things that exist around Kubernetes that you need to do to make it production grade. And we talked earlier about, you know, more mission critical apps moving to the platform. So you need these advanced technologies that give you better security, better stability and so forth. But the other thing that's driving complexity too is people are running it in more locations.

in more types of infrastructure. They may have started originally on the public cloud to run Kubernetes, and now they're adding it in their data centers too. And we're seeing an increase in Kubernetes getting deployed at the edge too, simply because that's where a lot of the new data exists, right? And in the industry, we've always been saying data has gravity. It's generally...

easier to move compute to your data than the other way around. So you get Kubernetes in all these places. And so that's posing new challenges, right? How do you manage that? How do you manage a lot of different clusters in a lot of different locations? So, you know, we need to

also be looking at management approaches. And I think, you know, we talked earlier about, you know, it's been a decade with Kubernetes and CloudAid. One of the things I see changing too in the second decade is

people are using a different management approach to Kubernetes. I would say the first decade was really DevOps oriented. Lots of small clusters managed by individual teams. And in the second decade, I think we're moving to more centralized management because doing things like security, like observability, governance, it's easier to do that once for all of your clusters than doing it individually.

The Cloudcast (12:33.977)
Yeah, yeah. And I actually love that you went there because in a lot of the guests we talked to on this show, mean, we talked to a lot of sometimes super early stage startups and it almost seems like just as there's always waves in the industry, but to talk about that management plane specifically for one second, the whole idea of a centralized management plane and a SaaS based management plane has almost become table stakes.

I mean, it almost has become like if it isn't people kind of scratch their head and go, wait a second. What what's going on here? Like that just seems a lot harder to manage. I think the light bulb has kind of gone off in the industry in general. So let's let's talk about another area of like, while we're kind of talking about some some complexities and areas we can improve on with Kubernetes. So so back in the day to IQ days, we had DKP, the Kubernetes platform, and it has merged or our

Tobi Knaup (13:14.411)
Absolutely.

The Cloudcast (13:31.752)
morphed into Nutanix Kubernetes platform NKP. And then we also worked on another one, was Nutanix data services for Kubernetes. But I don't necessarily want to plug those per se, but I did want to talk about the idea of this platform and data services and how the need for it, because I feel like this was one of the super underrated things that kind of no one liked to talk about.

If Kubernetes being a platform for platforms was messy, data services for that platform of platforms was even more messy at times. And so give everyone, give everyone who's not familiar, like a little bit of overview of this problem, the challenges, especially when you're talking about edge or our multi cloud environments that we're finding.

Tobi Knaup (14:10.2)
That's right.

Tobi Knaup (14:22.946)
Yeah, absolutely. Everything's always nice and beautiful until you start introducing state. And so, in terms of state and data management, it's kind of been a long journey in cloud native too. you

might remember that when Kubernetes actually first showed up on the scene, it had no support for stateful workloads.

It was stateless microservices only. You couldn't run something like a database on there easily. It was very, very difficult. There was no support for volumes, for instance. CSI, which has become the industry standard, the Container Storage Interface, which for the first time allowed containers to provision claim volumes.

actually something that the Mesosphere engineers worked on back in the day in 2017 along with Google. That for the first time allowed us to run stateful workloads, but it was still a pretty low level API, right? It's a volumes API, I can create a volume, et cetera. But if I want to do something application centric, right? If I have...

you know, some application that stores state and I need to think through, all right, like what's my disaster recovery plan for this application? How do I actually survive losing a cluster, for instance? How do I replicate that data into a second physical location, whether that's on cloud or on prem, and then come up with an automated procedure to restore not just the volume, but the whole application.

using all the Kubernetes API manifests that are needed to that, bringing all the pods, not just the volume, in a different physical location.

Tobi Knaup (16:06.636)
that's still something that's very difficult, right? And there isn't anything in Kubernetes at the moment that would give you an application-centric construct to say, all right, here's this application. I want this snapshot policy, and I want to synchronously replicate this data to another location so I can very quickly bring it back up.

should my primary location fail. And that's just one use case, right? It's, want HA for this application across two different physical sites. But you know, one thing that's interesting too is really, and I see this a lot over in Europe, folks are really a lot more concerned about data governance and data sovereignty right now, especially when they're putting things into the public cloud.

Right. So effects less in the U S right. Because the hyperscale clouds that we used, you know, in the U S state, they are U S companies. But in Europe, there aren't hyperscale clouds, right. You use U S companies. And so they're really a lot more concerned around around taking control of their data, owning their data. And so those folks also now thinking, okay, well, if I put my data in a public cloud,

Can I also put it somewhere else? Or can I at least have some kind of off ramp? Should I decide to take it with me, take it out of the cloud later? And so that's a problem that we've addressed here at Nutanix with NDK, the Nutanix Data Services for Kubernetes. So it builds on the lower

level Kubernetes APIs, the CSI, the volumes, but gives you these higher level application level constructs where you can say, all right,

Here is my data replication strategy for this application. And here's my disaster recovery strategy for this application.

Tobi Knaup (17:55.866)
And you do that at a very high level and you can do it independent of the underlying infrastructure. So you can do this on a public cloud, replicate to an on-prem environment or do it on a private cloud and replicate to a public cloud. So it doesn't really matter what the underlying substrate is. And so I think that's another, we talked about platform, platform earlier. I think that's another example of that. Maybe that's the data platform, platform.

The Cloudcast (18:22.627)
Yeah, I love that. I love that. So final topic, because I feel like I'd be remiss. We touched on AI briefly, but I want to talk a little bit more about AI and its impacts in the space, because I feel like to a certain amount, AI kind of came along and sucked all the air out of the room for at least a time there. How do you think about AI today? And now that the dust has settled a bit, is it?

Tobi Knaup (18:43.512)
Yeah.

The Cloudcast (18:51.011)
Is it just a quote unquote app that's running on top of the platform? Or do you think there is, as you mentioned earlier, some longer term impacts to say operations or some other things like that? Like I guess there's almost like a how do you see it running on top and how do you see it being integrated in?

Tobi Knaup (19:10.07)
Yeah, I mean, it's a super exciting time, right? Just turns out that Kubernetes is just the perfect fit for AI workloads for, you know, the dynamic nature of training jobs that need to, you know, spin up and were, you know, at least if you're doing AI development on a smaller scale, you typically have, you know, very expensive hardware that is shared among different people. And so you want to share it efficiently. And that's something that Kubernetes is very good at. Also, you know,

on the other end of AI. there's training, there's inferencing, right? On the inferencing side, as these AI systems are becoming more important, more mission critical, they're treating important data, dealing with important data, and they need to be always available. Well, it turns out all the best practices that we've built around running production systems and containers, observability and CI, CD and all that stuff. Well, it turns out these inferencing workloads are also

in a sense, just another container. And so you can reuse all these tools and just run them as highly available production systems that way. So a lot of that stuff translated, but AI workloads

are also different in some important ways. And so I like to think of AI on Kubernetes, AI below Kubernetes, and AI in Kubernetes. So on the AI on Kubernetes side, that's what we just talked about, right? Like running containers with AI workloads using all the

observability, security tools and so forth that are already there. AI under Kubernetes, that's the, okay, how do I use the AI hardware efficiently in Kubernetes? And the community actually started working on this several years ago, before the big bang chat GPT moment. And so, support for GPUs as resources that can be scheduled.

And, you know, using the DRA, for example, it's a Kubernetes API that's been in there for quite some time. NVIDIA started working on a GPU operator several years ago. this will, there's been good AI hardware support in Kubernetes for quite some time. Now, there, as these workloads become more prevalent, I think there are also tons of opportunities for optimizing how things work. So.

Tobi Knaup (21:30.51)
the AI workloads, they are different in some ways and in the way they need to be scheduled. For example, if I run a large training job, that's not just like a handful of containers, that's a very large job. And I need to, in a lot of cases, do what's called gang scheduling. So I need to deploy a very large number of containers and kind of deploy all of them or this job won't work, right? And so there's work happening in the community to...

improve the Kubernetes scheduler or build a different scheduler that is more tuned to these types of workloads. Great thing about Kubernetes is it has a flexible schedule architecture, so you can plug in a different scheduler and make that more efficient. Then I think the last thing that I see, we talked about it a little bit earlier too, my wish list of having this higher level agent that translates my business needs into YAML,

cool stuff out there like K8's GPT, that's a project in the community that does things like that, you know, removes the burden from Kubernetes operators and gives them a nicer, more human interface to the system. you know, it's on Kubernetes, it's in Kubernetes, it's below Kubernetes, it's everywhere.

The Cloudcast (22:45.721)
Fantastic, I love that. That's great summary and I think that's a great place to close us out for this episode as well. So Toby, where can everyone go to kind of learn more or get started or kind of follow what you're doing in the community?

Tobi Knaup (23:01.09)
Yeah, so, you know, we blog a lot on our Nutanix blog, actually, if you just go to Nutanix.com slash blog, you'll find us, you know, writing about things that we do. And you can find our team members in the CNCF community too. So, you know, we work on cluster API, for instance, you know, some of our folks are hanging out there.

The Cloudcast (23:25.571)
Well, Toby, thank you very much for your time this week and everyone out there. Thank you very much for listening. If you enjoy the show, please tell a friend. If you can leave a review wherever you get your podcasts, we would certainly appreciate that as well. And we're always looking for feedback. Show at the cloudcast.net. Thank you everyone for listening this week and we will talk to everyone next week.