EPISODE 1482

[INTRODUCTION]

[00:00:00] ANNOUNCER: This episode is hosted by Lee Atchison. Lee Atchison is a software architect, author and thought leader on cloud computing and application modernization. His most recent book, *Architecting for Scale*, is an essential resource for technical teams looking to maintain high availability and managed risk in their cloud environments. Lee is the host of his podcast, Modern Digital Business, an engaging and informative podcast produced for people looking to build and grow their digital business with the help of modern applications and processes developed for today's fast-moving business environment. Subscribe at mdb.fm and follow Lee at leeatchison.com.

[INTERVIEW]

[00:00:46] LA: As companies move more and more workloads to the cloud, cost management in the cloud is becoming critically important. Modern application development methodologies increasingly involves moving to service and microservice architectures enabled by the cloud, and Kubernetes is the backbone of this modern infrastructure trend. Managing costs of Kubernetes clusters is therefore becoming increasingly important.

Kubecost is a company focused on giving visibility into Kubernetes resources allowing you to reduce your overall Kubernetes-based infrastructure spend. Webb Brown is the founder of Kubecost and joins us today.

Webb, welcome to the show.

[00:01:28] WB: So great to be here, Lee. Thank you so much for having me. Really excited for the chat.

[00:01:33] LA: Great. And this is not your first time on the show, is it?

[00:01:36] WB: It is not. I'm proud to be a return guest. And a lot has happened since I first joined. It was soon after we started the Kubecost project that I first joined Jeff. So, excited to share many updates since then.

[00:01:49] LA: Cool, cool. And I certainly suggest that everyone who's interested in what we're talking about today, go back to early 2021 and listen to that episode of Software Engineering Daily with Kubecost. It's a great episode with great information. And I'm looking forward to finding out how things have changed between then and now, and what you're focused on and all that sort of thing.

Let's just get started with some basics. You're a SaaS-based monitoring platform. Is that a correct high-level analysis? You're a SaaS-based platform?

[00:02:20] WB: Actually, our product itself gets deployed in our users' clusters or BPCs. It is by default fully deployed. We actually do have a SaaS solution that is in limited availability that we actually haven't announced broadly. But we're going to announce that soon. But yeah, our core product, Kubecost community as well as enterprise is based on our open source and is deployed fully in our users' clusters.

[00:02:49] LA: It's deployed on-site or within the user's environment, whether it's cloud-based or on-site. How easy is it for someone to add your product to their existing Kubernetes clusters?

[00:03:02] WB: It generally takes just a couple minutes. It is, today, two commands for most users. It is simply a Helm add package and then Helm install. And that deploys kind of all of the dependencies that Kubecost needs. If you've run like a Grafana, for example, or maybe even Prometheus, I would say think of it similar, in that it just spins up like core Kubecost pod, as well as time series database and an optional Grafana if you want that. So yeah, generally, minutes if not less for most clusters.

[00:03:36] LA: Now, you mentioned Prometheus. It actually runs on top of Prometheus. Is that correct?

[00:03:41] WB: We support basically anything that speaks PromQL. Prometheus, Cortex, Thanos and other time series databases. That is – Like, we ship with a Prometheus by default. It can be optionally disabled. And then Kubecost itself can be deployed in, say, like agent mode where you aggregate data across many clusters. And there, we would just be kind of shipping with a very lightweight time series database. Like, all of the metrics and data that we generate around costs, and efficiency, and allocation, all that is written to a Prometheus data store. So then, you can do really cool things, like, use alert manager, or do real time rules are monitoring in the Prometheus UI, that's sort of stuff.

[00:04:28] LA: Cool, cool. Okay. It is very heavily tied into the Prometheus environment anyway. But it's not Prometheus-based you can use any of their Prometheus compatible systems that work together. And they all work together with Kubecost.

[00:04:44] WB: Yeah, that's exactly right. That interface ends up being kind of PromQL. And then we expose kind of like our metrics from a /metrics endpoint that is just pulled from any kind of Prometheus-based ingestion.

[00:05:00] LA: Makes sense. Yeah. Well, let's talk beyond monitoring for a moment. Let's talk about optimization. I know you focus a lot on cost optimization. And optimization is not just measurement, but it's also finding ways to reduce and to manage the costs of a Kubernetes environment. How do you provide insights into usage optimization specifically?

[00:05:24] WB: Yeah, we do many things today. You touched on it first, is deploying our application and getting that map of usage across your environment and resource consumption. And again, we want to start by giving that like transparency and awareness. Doing that based on kind of your enterprise discounts and your use of spot, versus RIs, versus Savings Plans. First is starting to get that like accurate and real-time allocation of cost.

And then we actually typically start with teams or recommend they start with just one number, which is our notion of cost efficiency, which is effectively saying, in your Kubernetes cluster, for all of the things the resources that are allocation-based that you provision, say, a node, or a disk, or attached, or PV, etc., what percentage of every dollar that you're spending are you actually utilizing, right? Like, doing that on a cost-weighted basis. That is oftentimes a great

indicator or a great starting point for like where you can optimize and the actual impact of that optimization.

From there, we would look starting at the bottom of the stack, like the actual infrastructure itself. Thinking about like right-sizing or rebalancing infrastructure. We have some teams that like dig in at the OS level and think about like prioritization of different requests, etc. And then a lot of time is spent like at the orchestration level itself, right? Thinking about auto scaling, and replication, and pod disruption budgets, all these things that are at the like scheduler and Kubernetes API level. And then from there, we'll dig in even to like containers and the applications themselves. So a common one being like right-sizing or even scaling requests and limits over time, etc. It's really looking at all aspects of the stack to think about these like really fine-tuned optimizations where most teams can typically see savings of like 50% plus depending on kind of the state of their infrastructure and also what they're willing to like dig into.

[00:07:34] LA: In order to do that, you need metrics not only on the number of each resources, and things that are being used, and how they're being used and what the cost of each of those resources are, but also, how they utilize. How well they utilize. You just grab that from other Prometheus metrics, correct? You're requiring performance monitoring to be enabled in order to get those metrics so you know usage utilization. They can then do those optimization percentages.

[00:07:59] WB: Yeah. We'd be doing this join across your billing data, data from the Kubernetes API, data from, say, like cAdvisor for CPU utilization, etc., the kernel itself for like network monitoring. Yeah, we would join all that together. And our view is fundamentally that you need to think both, again, kind of the like allocation or provisioning side of the equation for cost, as well as the utilization.

We've actually just released a big new spec on this. Just part of the brand new OpenCost launch, which really details exactly that and shares exactly what you just said, which is like the individual cAdvisor metrics that are typically scraped from a Prometheus.

[00:08:44] LA: I actually want to get into OpenCost. I've got a little bit further down the list. If we can hold on to that just for a little bit. But I'm definitely very interested in what OpenCost is doing

and how it all works. But let's keep on the optimizations for a moment here. And I know you're a young company. And so, answers have not yet probably are phrases that you use a lot. I get that. But what about using other performance optimization technologies other than the base Prometheus? Using things like commercial products like Datadog and New Relic? Do you see yourself as working with companies like that to provide this capability on top of what they provide? And do you do any of that today? Or is these things you're thinking of in the future? Or how do those commercial products fit into this whole universe?

[00:09:28] WB: Yeah, absolutely. We are working with some of those already. And I would expect a growing number over time. And it's a really interesting time for that question or discussion in my view. And I would say it's in a large part due to OTel, or Open Telemetry, which is the fact that when you support that like data standard or format, really, most observability stacks can ingest these metrics, right? And by speaking common formats that Prometheus ingest, a lot of observability solutions are able to just ingest these metrics out of the box.

So, yeah, it's a really exciting time because of that. And I think, increasingly, as a ecosystem, we're starting to see that. Yeah, we're aligning on a lot of the same interfaces and standards. And that I think will only get easier.

[00:10:21] LA: Yeah. I'm really excited to see some of the things that Open Telemetry can be used for and how it's going to change really the dynamics of the whole monitoring space in general. It's going to be very interesting several years, I think.

[00:10:33] WB: Agree. It feels like it will be big. I will just say that, over the past couple years, have spoken with hundreds, if not like a thousand plus teams at this point. I just hear it you know as a topic of conversation top of mind more and more. And again, I think just like the portability and all the really interesting benefits we get from it I expect will be big.

[00:10:56] LA: Cool. Cool. I know when I worked at Amazon, I was involved in Amazon. They're relatively early in their career in the early days of AWS and retail before then. When we were doing some of the early retail days, managing infrastructure costs was a huge issue for us, because resources were allocated and owned by individual teams. And individual teams

controlled their own destiny with determining how many servers they needed for their services, etc.

And as you can imagine, we have 100 people all allocating their own resource utilization. There's not a lot of optimization that goes on. And that actually became huge problems for us, especially as we started approaching holiday times, Christmas time, which was, for Amazon retail, we would have double, or triple or quadruple the traffic we would any other time of year.

And so, costs of managing resources was always a huge issue because of the way we manage costs at an individual team by team basis. How does a product like Kubecost help companies like that manage at a team by team basis?

[00:12:08] WB: Yeah. I think it's a great example and just a very relevant question to lots of companies in this ecosystem. And in my view, at least oftentimes in Kubernetes land, is in a lot of ways we're talking about balancing these two very important concepts, which is innovation or ability to ship and move quickly with managing costs, right? Because giving your engineering teams the ability to deploy whatever it is, jobs, deployments, replica sets, etc., on your infrastructure and do that without a ton of like hurdles is a great thing for shipping products. But yes, when that is decentralized across all of your organization, there's no awareness. Tragedy of the commons can ensue.

We think about it in three ways. First is just awareness and transparency, right? What we see is – As former engineers and the experience we had working at Google on similar problems is, as an engineer, you're busy. It's really hard to like – It's not right in front of you to like even know what you're spending, right? Like you spin up application and you request five cores and you aren't focused on like the number of replicas as they're configured in your auto scaling group. And like you wake up and you are told that you spend many multiples of what you expected to spend later sometimes.

First, it's just like bringing that data to engineers we found can be transformative, right? And again, it's much harder, generally harder in Kubernetes, because everybody's sharing resources oftentimes, right? You have multi-tenant clusters. So it's less like you go to a core cloud group

and say, "Give me X VMs," and they send you a bill and you kind of show that charge back. That's part one.

And then second is just having some measurement of guardrails in place, right? Whether it's budgets, alerts, policies that are, say, utilization or efficiency base. There's a lot of things that can just kind of set up these loose guardrails without actually slowing innovation.

And then lastly, I would just say like some measurement of automation can be really helpful here. I'll just give you an example. We have some Kubecost and some really interesting integrations with products like Spinnaker, where as part of your CI/CD pipeline, you can actually look at utilization history of applications that have been running. And when you go into a new release, you can have automation kind of right-size that or take action on your behalf. And that's just one example, I would say. Like a growing amount of automation here that can really help teams at scale.

[00:14:48] LA: You mentioned the P word, policies, and I think that's important especially for a lot of these larger companies, like the amazons, if you will, and how they manage their cost at scale. And I'm talking about organization scale as much as I am product scale.

I imagine that's an area where you can talk about policy management now, but you probably have some ideas for how to add more capabilities there and expand that either within Kubecost or with integrations to other products. Do you want to talk at all about some of the things you see where how policy management might become important to you as time goes on?

[00:15:29] WB: Yeah, absolutely. And I think it's worth pointing out that, in Kubernetes, a lot of teams are kind of just getting to this stage. In the sense that most teams, when they start using Kubecost, they generally didn't have a lot of visibility in the cost is most common, right? Because they may be monitoring costs at like the EC2 level. When it comes like accurately breaking that down by Kubernetes tenant, or team or application, oftentimes, it's a lot of a black box, right?

Part one is simply getting that visibility in place. Part two is, "Okay, let's start looking at typical optimizations available." And then part three is typically, "All right, then let's put those guard rails in place."

And so, what we see is actually pretty interesting because Kubecost tends to work with – Work closely with a lot of like medium and large-sized enterprises. And a lot of them already oftentimes have kind of policy management or some type of financial policies in place, because many of them had like showback and chargeback and other programs in place from their BM days.

Right now, I would say we're doing a lot of like integrating with other systems. I would just say, I do see an opportunity for Kubecost itself to make that a little simpler. The interesting thing there is that, yeah, a lot of kind of custom actions are taken based on those policies. Sometimes it can be alert this owner. Other times it can be kind of prevent new deployments from happening. Depending on the context of the group or the scenario, you want to be generally really thoughtful. Because, again, I think you can start tapping into throwing that innovation and cost management balance out of whack if you go too far one direction.

[00:17:16] LA: Yeah, there'll be demons there if you're not careful. And how to make some of this work. Now, you also mentioned – And when I did a little bit of research, I noticed that you mentioned that you also help manage related infrastructure costs but aren't directly Kubernetes related. And I think the example on your website that you talk about is S3. You store containers. I'm assuming the use cases. You store containers in S3, and they're managing that S3. And you start Docker images in S3. Either managed in S3, right? There's part of the Kubernetes ecosystem, but not directly part of Kubernetes. Can you talk a little bit about what you do for those sorts of resources and how you incorporate that into your whole strategy?

[00:18:00] WB: Absolutely. This would be a big change. Since I was last on the show, this would be all new. We kind of started with Kubernetes because we viewed it as a black box for a lot of teams and kind of saw a real problem there. But what we started to learn working with a lot of large enterprises is that, if you're using Kubernetes at scale, you're probably using many other, at least multiple other like external cloud services, right? Whether that's a storage bucket, or external database, or something else.

And what we've seen is that teams inevitably want this single view of like all their infrastructure spend. While we see Kubernetes at the center of a lot of our users' infrastructure, it's typically not 100% of their spend. Kubecost itself can now integrate with your cloud account on top of just being deployed in a cluster. And then it will pull on just all your billing data and let you see this unified view of everything side by side.

You can do interesting things like allocate whether it's a storage bucket or something else to a single deployment or to a single name space where tenants in that name space share that resource. That's a really powerful feature. And then, we also have some really other cool functionality that looks at, basically, your network traffic across different cider blocks so that you can actually say, "This particular pod, or staple set, or job is actually driving my cost with data transfer from this S3 bucket."

And again, for a lot of teams, untangling that is really hard because it's like, I don't know, just something from this cluster or something from another region is talking to it. So, this gives you that visibility out of the box.

[00:19:47] LA: Yeah, that sounds incredibly powerful. You can spot runaway processes even, or poorly-designed processes and all those sorts of things a lot easier with that sort of capability. That sounds really cool. That's kind of related to my next question, but it's a little bit different take. And we always talk about one of the advantages of modern digital applications, the Kubernetes-based cloud, cloud native applications is their dynamic, right? You can change the size dynamically. You can allocate resources only when they're needed and freed when they're no longer needed. And a lot of those sorts of things that help you control costs in a dynamic fashion.

Since your cloud cost is central to what you do, I'd be interested to get your perspective. Are people really building dynamic applications like that where they are trying to control costs and trying to control infrastructure by bringing infrastructure up and down only when it's needed and doing things in that sort of a dynamic way? Or is that more theory and they're still statically allocating their resources for the most part?

[00:20:55] WB: Yeah, it's really interesting. I would say that even since we've launched the company, which is just over three years ago, we've seen a shift. And that is teams getting way more comfortable with dynamic scaling. And that could be, again, kind of at the application level, or the replication level or at the infrastructure layer itself.

Yeah, I think this is, as teams get further along in the adoption curve with Kubernetes, you can oftentimes just very clearly see the benefit. Again, that's kind of going back to that cost efficiency number. When you're in single digit cost efficiency because you're like basically scaled for – Yeah, like provisioned for peak load, right? It doesn't take a ton of analysis with a tool like Kubecost to say like, "Here's the potential benefit from scaling." I think it's a function of, as we just grow scale, the win, like the dollar gains from like being really efficient there become more and more appealing.

Now, that being said, we work with a lot of teams closely on how they dynamically provision, right? Because, obviously, going too far on the cost dimension can lead to reliability or performance risk or concerns. Generally, we still see teams being very thoughtful and cautious when they move down that path. But absolutely, we see teams realizing the benefits from dynamic provisioning and see them just getting bigger and bigger, it feels like, every day at this point.

[00:22:33] LA: Imagine, over the last few years, if you've been able to keep track of this, from an industry standpoint, the percent utilization of resources has probably grown steadily over the last several years as dynamic resources have taken control and a lot more dynamic systems. And so, a lot less wasted resources for peak capacity, etc.

And so, I imagine, if you could keep track of that, and maybe you are. And if you are, I'd love to know about that. As an industry, I imagine that number slowly creeping up over-time.

[00:23:08] LA: It's a fascinating point. I think there's actually two ways to look at it. And obviously, our data set would be heavily skewed towards like Kubernetes users and microservices-based architectures. The two ways I think would be interesting to look at that is, one, on a cohort basis. And then two, on an aggregate basis.

And one thing I can say with like very high confidence is that if you look at it on a cohort basis over the last like two years, I absolutely see like major, major improvements. But in parallel to that, Kubernetes has seen just insane growth, right? And so, there's been a lot of users that are new to Kubernetes. And what we've seen is that first period with Kubernetes transitions is typically let me get my arms around Kubernetes itself. And then I'll worry about efficiency once I'm kind of stabilized and/ or reach a certain scale. On a cohort basis, I definitely think it's improving. On an aggregate basis, it's probably improving there as well. But I think –

[00:24:14] LA: Well, it may not be as obvious based on the skew of the new users.

[00:24:18] WB: Yeah. Yeah, that's my intuition. I would say that like because we are a deployed product, I have a lot of teams that will share their data with me with our team to get input. But because the way that Kubecost is architecturally architected, we very intentionally don't require ask you to send anything. But that being said, anecdotally, we see a lot of teams that are around the like 20% cost efficiency mark when they start. That can, I would say, very commonly be kind of like increased by 3X. A lot of teams that can get to 60% utilization pretty quickly.

And then obviously, when you drill deeper, that varies heavily by use case and predictability of demand, etc. But I will say that I'm surprised by the like clustering, if you will, like in the 60s in terms of cost efficiency.

[00:25:12] LA: Got it. Got it. I want to go back to a quote that you used in the last podcast. I'm kind of throwing this at you. There's no preparation. You probably don't even remember saying this. But let's see if we can wrap our mind around what we were talking about.

In the last show you stated that there is a lot of complexity in the day-two part of the Kubernetes journey. And I think what you were talking about is that managing applications at scale is still a challenge for most companies. This goes back to the – There's a lot of newcomers that come in and say, "I just want to build my application with Kubernetes. I'll worry about making it more efficient and performant later. Just get me in the door now and I'll do the rest later." Were you referring to those that move to the more sophisticated customer is something that's still in its infancy, if you will? Or what were you thinking of with that?

[00:26:08] WB: Yeah, yeah. I apologize if I don't have all the context of the exact quote. But like it's really interesting to think about that versus more than a year ago. And I think we've seen a noticeable shift here as well. For sure, at enterprise complexity, I think there's a lot to think about when deploying Kubernetes. And my view is, today, kind of part of the trade-off of like having all of the flexibility and the power and the control that's granted to you by Kubernetes.

I think there's a couple things that are happening. One is you've seen this massive rise or shift towards manage Kubernetes offerings, right? EKS, we see has just like grown so much in adoption. And that versus standing up your own network stack and managing Kubernetes yourself has allowed a lot of teams to kind of focus on higher up the stack activities, if you will.

And then secondly, I think you're just seeing a lot of the ecosystem transition to this discussion, right? Which is, "Hey, can we do better here?" And again, I think it's really natural evolution with where the market is, if you will, right? As more users are managing production applications and Kubernetes at really large scale, it makes sense for the providers and observability players and all of these other companies be more and more focused on that.

I think you've seen major, major improvements. But at the end of the day, I think it's fair to say that like Kubernetes is this incredibly a powerful tool, and there still can be a fair amount of complexity exposed. And I still see opportunity from a cost dimension as well as others for us to like help organizations running containers and Kubernetes at scale.

[00:28:12] LA: Cool. Cool. Now, you mentioned this early on, but I want to come back to it, because I wanted to give you plenty of time to talk about this. And that is I want to talk about OpenCost.

Now, you recently announced an open source project for monitoring Kubernetes spend called OpenCost. First of all, why don't you tell our listeners what exactly is OpenCost? And what does it do for them?

[00:28:36] WB: Yeah, absolutely. OpenCost is this new open source solution for monitoring Kubernetes cost. It allows you to monitor costs across basically any Kubernetes dimension in

real-time. And so, then you can then use that as an input for other interesting things like scaling and learning, etc.

There's two parts of OpenCost project today, which was just announced last week. There's a brand-new spec that was built with a bunch of amazing ecosystem partners. So, partners that are cloud providers. So, AWS and Google and then other players in this ecosystem, like New Relic, and SUSE, D2iQ, as well as kind of like end users who are major users of Kubernetes like Adobe. That group came together and developed this really detailed spec for how you can do cost monitoring in Kubernetes environments. And again, it's touching on all the topics that we touched on earlier, like taking into consideration usage-based cost, as well as kind of like allocation or capacity-based cost.

And then there's a second part of OpenCost, which is an actual Go lang implementation, a software implementation of that spec. That actually was built as the original Kubecost core application engine. It is everything that's used in our product. And really, the kind of like heart of how we do kind of this real-time cost monitoring.

[00:30:17] LA: So, is OpenCost more than just Kubernetes costing? Or is it focused on Kubernetes?

[00:30:23] WB: It is focused on Kubernetes today. The implementation supports through really four different environments. So, like, all the three major cloud providers; AWS, Azure, and Google as well as on-prem environments, where you can bring your own custom pricing sheet. If you look at kind of the charter or scope of that, it is in Kubernetes-based environments. There's already discussion about potentially extending that beyond. But again, where we see lots of ambiguity is in like really dynamic Kubernetes environments and really working with a lot of end users as well as partners over the last couple years has really highlighted that.

[00:31:02] LA: Mean. Mean. Someone who wants to think about managing the Kubernetes costs, coming into the ecosystem today, and looking around and they see Kubecost on one end and OpenCost on the other end, how would it differentiate between them? What would you tell them to move towards one versus the other? How are you positioning them?

[00:31:24] WB: Yeah, great question. And OpenCost is brand new. Literally, 0.1, which is released last week. This will likely be evolving quickly. And again, I think you can think about it as the Kubecost cost project implements the OpenCost spec and ingests the OpenCost implementation to build all of its functionality.

The Kubecost project is designed to be kind of enterprise-ready, if you will. Give a lot of the things like RBAC, and advanced reporting and things that like larger organizations need. Whereas if you're building an integration, or building a tool, or just want it on your like own cluster or your smaller team, that's where we think OpenCost is a great fit.

But I'm really excited about – And hope we see lots of just cool innovation built on top of that. Because again, I think we're just at the very beginning where us as engineers are increasingly getting costs into our kind of observability metric set. And again, I just think we're already seeing really cool examples of things that you can do with that information once it's available. You can kind of touch on that like Spinnaker example of like I could see a world where all CI/CD pipelines have this notion of cost and allow you to make either kind of manual or automated decisions on that information in real-time.

[00:32:48] LA: That's cool. Yeah, I can see lots of use cases there. That would be quite interesting. We'll have to see what actually is developed in the universe and who all is involved in this. Outside of the major cloud providers, who's signed on to OpenCost so far? I think you mentioned like Datadog and New Relic. But who else has been involved?

[00:33:09] WB: Yeah. We've had worked closely with New Relic, and D2iQ, and Armory, and the group at Minecurv, SUSE and a couple others. And all of these were really valuable founding contributors to the spec and really kind of help think through and architect this design. That entire founding group has, again, open sourced everything we've done. And we've also started the process. We've applied to contribute this project to the CNCF. And our hope is that we can find a great neutral governance home and just get further input from the broader ecosystem as time falls.

[00:33:50] LA: Yeah, you actually anticipated my next question. You've submitted it as a sandbox project specifically. What do you expect that that will bring you? Assuming you get that,

what would that give to the project besides the stamp of approval of CNCF? What benefits do you see of getting listed as a sandbox project?

[00:34:11] WB: Yeah. I think it naturally brings kind of better and better alignment with these like really important CNCF projects today, right? We talked about Prometheus as an example. And obviously, like Kubernetes project itself. There's a lot of really interesting benefits there.

I think two is just saying that this is something that we truly want to be neutral. And the like ip and everything owned and managed by CNCF subsidiary of Linux Foundation. And hopefully, we see just more and more kind of contributions and collaboration across the broader ecosystem. Because we just think this is such an important foundational layer that we have a language of cost in the land of Kubernetes and cloud native. Because we feel, today, that when teams do kind of do like a DIY or implement cost monitoring on their own, a lot of them think about it differently in terms of how you're allocating shared resources. And like you said, are you using requests, or limits or usage? There's all these like nuanced questions that we just fundamentally believe that if we can agree on an ecosystem, then we can have consistent and reliable measurements of cost across all the major environments where Kubernetes is running.

[00:35:37] LA: Cool. Cool. Yeah, I'm anxious to see what happens there. It should be quite interesting. I'm really intrigued by the company Kubecost in general. I think you've got some great ideas you've been doing a lot in the last year. And I know you've just closed your last funding round in February, I think? Or not that long ago. Anyway, whenever it was. What's your roadmap? You don't want to make everything public, I know. But tell me a little bit. What are you working on now? What should people be excited about following Kubecost?

[00:36:08] WB: Great question. There's a lot we're excited about. I would say a couple like critical areas that we've kind of already touched on a little bit. One is just kind of the broader effort around OpenCost. We would very much expect to see us making a big investment just trying to help build a community around dynamic management of cost. Like, real-time monitoring of cost, etc. And our hope is there that you'll see many, many cool integrations built from our team as well as kind of our team enabling or supporting others.

We talked a little bit about how you know hosted or SaaS Kubecost is on the way. Stay tuned there. Lots more exciting updates to bring there. And then from there, I think it's really just kind of going deeper and just getting more and more intelligent with a lot of our optimizations that exist today and just getting better and better about predicting the state of your workloads and your infrastructure, as well as being more and more just contextually aware, right?

We now have tons of really interesting metadata about both your infrastructure and your applications when Kubecost is deployed in that environment. My view is we can really start to help teams kind of take out some of that complexity that is part of kind of managing cost as well as reliability and performance in these environments.

[00:37:41] LA: I think complexity management is a huge untapped valuable area of work in the cloud in general, but certainly in Kubernetes environments. I think it's going to be intriguing to see how things go in that area. That's great. If someone's interested in getting involved in open cost, either as an individual who wants to follow and see what's going on, or someone who's interested in helping out, or being part, or getting their company involved with what's going on there, what should they do?

[00:38:11] WB: Yeah, there's a lot, given that the project is brand-new. I'll just call out a couple areas. We just released this brand-new spec that we hope will help kind of define a way of doing cross-monitoring accurately and fairly in these environments. If you have interest there and want to review, or contribute or edit anything around kind of documentation and definitions, very much open and welcome to contributions on the open source project itself. There's a contributing guide that just – Like, if you go to githubkubecost.org and then open cost project.

And then, we also just have a community, whether it's joining our Slack community or joining our working group. If you just want to be involved in the conversation and kind of help set priorities or give feedback, we'd love to have you there if there's any interest. Again, I view it as like we're just getting started and there will be many more ways to contribute going forward. And again, one thing I'm really excited about is just all the cool stuff that we hope will be built on top of this new language of cost. Those will definitely be valuable, valuable contribution for anybody that has interest there.

[00:39:27] LA: And I'd be remiss if I didn't mention that OpenCost has a website at opencost.io. Is that correct? Is that a good place to get information?

[00:39:35] WB: Yes, that absolutely is. And that can take you to the GitHub repos, other places. That website was just launched last week. And there's a public repo for that. If you have you know changes you want to see or feedback, we very much welcome it.

[00:39:49] LA: And Kubecost itself, if someone wants to learn more about Kubecost, what should they do?

[00:39:55] WB: I would say just go check out kubecost.com. Feel free, again, to join our Slack community if you have questions and just want to talk about product or just optimization in general. Wow we'd love to have you there. Or you can also reach out to me at webb@kubecost.com and happy to help any way that I can personally.

[00:40:15] LA: That sounds great. Is there anything else that you want to say that I haven't asked you?

[00:40:21] WB: No. I think we've touched on lots of great stuff. Yeah, just super grateful for you spending this time with me. And really, really appreciate it, Lee.

[00:40:29] LA: No problem. Thank you. And this was really great, Webb. I really appreciate your time talking to me today. And my guest today was Kubecost CEO Webb Brown. Webb, thank you for being part and a guest on Software Engineering daily.

[00:40:43] WB: Thank you again, Lee. My pleasure.

[END]