EPISODE 842

[INTRODUCTION]

[00:00:00] JM: Google's cloud business was long regarded as a place where startups could build a business, but not established enterprises. For serious workloads, enterprises chose Amazon almost unanimously. This phenomenon of Amazon as the default was described by a phrase that harkened back to the days of IBM's dominance. Nobody ever got fired for choosing AWS.

But with the rise of Kubernetes, Google has established reputation as a reliable provider of container orchestration. As enterprises looked to roll out Kubernetes workloads, Google is convincing many of them to work with Google Kubernetes Engine, or GKE.

Kubernetes is one part of the changes which fall into the description of digital transformation. Many enterprises are looking to build out their digital transformation strategy. They're looking for strategies around multi-cloud and on-prem and policy management and consulting partnerships. There's a ton of stuff that's changing in the purview of the digital transformation. So how do you do a digital transformation?

Anthos is a platform where enterprises manage the resources and configuration of their cloud deployments as well as they can partner with service integrators and independent software vendors. Anthos is a platform from Google.

Aparna Sinha works on Anthos and Google Kubernetes Engine and other projects at Google. In today's show we discuss the process of digital transformation as well as the tactics for how a digital transformation is actually implemented at a place like a bank or an oil refinery. Aparna talked about GKE on-prem and the kinds of tooling needed by on-prem application deployments.

We have a new version of the Software Daily app for iOS. You can use our APP to access all 1,000 episodes of Software Engineering Daily in one place. You can sort by technologies, like you can find all the episodes about streaming data, or cryptocurrencies, or Kubernetes, or if

you're learning to program, you can connect with other listeners through the comment section. You can access our transcripts and our related links. Everything in the app is free. Although you can become an ad-free listener and support the show for $10 a month of $100 a year by going to softwaredaily.com/subscribe.

We're also booking sponsorships. If you're interested in sponsoring Software Engineering Daily, you can go to softwareengineeringdaily.com/sponsor. We're hiring two interns for software engineering and business development. If you're interested in either position, send me an email with your resume to jeff@softwareengineeringdaily.com. Put internship in the subject line.

With that, let's get on to today's show.

[SPONSOR MESSAGE]

[00:03:13] JM: When a rider calls a car using a ridesharing service, there are hundreds of backend services involved in fulfilling that request. Distributed tracing allows the developers at the ridesharing company to see how requests travel through all the stages of the network. From the frontend layer, to the application middleware, to the backend core data services, distributed tracing can be used to understand how long a complex request is taking at each of these stages so the developers can debug their complex application and improve performance issues.

LightStep is a company built around distributed tracing and modern observability. LightStep answers questions and diagnosis anomalies in mobile applications, monoliths and microservices. At lightstep.com/sedaily, you can get started with LightStep tracing and get a free t-shirt. This comfortable, well-fitting t-shirt says, "Distributed tracing is fun," which is a quote that you may find yourself saying once you are improving the latency of your multi-service requests.

LightStep allows you to analyze every transaction that your users engage in. You can measure performance where it matters and you can find the root cause of your problems. LightStep was founded by Ben Sigleman, who is a previous guest on Software Engineering Daily. In that show he talked about his early development of distributed tracing at Google. I recommend going back and giving that episode a listen if you haven't heard it. If you want to try distributed tracing for free, you can use LightStep and get a free t-shirt. Go to lightstep.com/sedaily.

Companies such as Lyft, Twilio and GitHub all use LightStep to observe their systems and improve their product quality.

Thanks to LightStep for being a sponsor of Software Engineering Daily, and you can support the show by going to lightstep.com/sedaily.

[INTERVIEW]

[00:05:24] JM: Aparna Sinha, welcome back to Software Engineering Daily.

[00:05:27] AS: Great to be here. Thanks for having me.

[00:05:29] JM: We've been doing a lot of shows on digital transformation, and this is a trendy term, but it's also a reality. There are lots of older companies that are restructuring themselves as software companies, and these are companies like banks and oil companies, manufacturing companies, and they have a very different problem than startups. They have legacy servers to manage. They have legacy software to manage. These are actual physical servers running old software, and they have thousands of engineers.

Let's say you're the CTO of one of these companies, whether it's an insurance company, or a bank, or a soy bean manufacturer. You're in charge of the digital transformation. What is your strategy?

[00:06:19] AS: I think this is perhaps the most exciting thing that is happening in enterprise software and enterprise IT today, is the opportunity that is in front of people like that, in front of CIOs and CTOs at large companies. I've had the privilege of talking to many such large company's CTOs and CIOs, because they are increasingly looking at Google Cloud as a cloud that can help them modernize and take advantage of the systems and the data that they have today in their data center.

I think the best way to understand it is to look at a few examples. Large companies, like you said, once that have been around for hundreds of years in fact. I had two customers speaking

with me at our next conference. One is HSBC, which is more than 100 years old, and the other one is Loblaw, which is a Canadian grocer, which is also more than 100 years old.

Both of these companies, I think the CTOs at these organizations, they're actually kind of I would say fresh talent that those companies have hired to take that company into the next generation. For example, with Loblaw, they're building a loyalty program that is rewards-based and is a car-based program. They're running on GKE. They're using this cutting edge technology because it allows them to do things in a digital way and much more quickly and to establish a relationship with their customers, but also by building on modern technology, it allows them to attract engineering talent that they wouldn't have access to otherwise, because engineers today, they want to work on the latest, and Loblaw wants to get access to the latest engineers.

By doing this, by using modern technology, they not only help their business achieve results that weren't possible before, but also set kind of a new standard for their whole industry, and this is what HSBC is looking to do as well. HSBC is looking to transform the relationship between a bank and you, the consumer, to make it a much more personal, much more real-time kind of relationship.

So even though these companies have thousands of legacy servers and data centers, they realize and they recognize that in their industries, they need to set the standard for what is technologically possible, because that's the way for them to retain and engage their existing users as well as win new users. That's where technology is. It is entirely possible to do those things with technology today.

I'm particularly excited about the opportunity that these large, older enterprises have, because they have so much existing data. They have so many existing customer relationships that they're actually more powerful than, say, a startup would be. Even though a startup may have an edge on the technical side, when you bring in talent, a talented CTO into a large company, the things that they can do I think are more powerful than even a startup can do.

[00:09:20] JM: If you're one of these people who is making technology decisions at a large company going through a digital transformation, you have this palette of options to choose from.

You've got multiple hyper-skill cloud providers. You've got your own infrastructure and you can deploy your own infrastructure in a way that feels like a cloud provider. Like you have a cloud provider on-prem. You've got providers like Confluent, or MongoDB, or just these companies that have a ton of experience running a particular software. How do you choose from this palette of providers that you could go and shop around with?

[00:10:08] AS: Yeah. I don't think people choose providers so much. I think when you're a CTO at a large bank or a large retailer or any of these enterprise companies, you look at your opportunity. You say, "What do I have and what is happening in my industry and what do I want to enable?"

So, for example, I was talking to a really old, a long-time, super well-known automotive insurance company. I won't name them, but everybody uses them, and they brought in a new person. One of the things that you want to do with a new person is like have an impact really quickly. So what he did is he said, "Well, we have so much data about our users and we're not doing anything with it. It's just sort o sitting around. There's also all of these information that you can get from maps and from a real-time information about traffic patterns. Why don't I use that?"

He kind of put those two things together, and in six weeks, came up with a better way of deploying their field force. So this is sending help to people that are stranded on the road. Came up with a better way of doing that so that they could reduce the time that it takes to get there to 5 minutes, or 10 minutes from half an hour.

As a user, as a car insurance user, you can immediately feel, "Wow! This company is like awesome. They actually know what I need." How do they do that? They did that on Google Cloud using our machine learning capabilities on GKE with maps data. So they know where the accidents are, what their patterns are, where is more likely that one of their users is going to be stuck with an automotive incident, and they just place their field force in proximity of those events. So they're able to improve their customer satisfaction, and it's something that's palpable, and he did it in six weeks and presented it to his board, and his board was floored, like, "This hundreds of year-old company had this data forever, but hasn't been using it."

I think when you join or when you start to lead a company like that, you have to take stock of, "Well, what's happening in my company and what can I do that would really move the needle either in terms of revenue or in terms of cost reduction or in terms of user satisfaction?" Usually, if you can make the company improve its user satisfaction or get closer to its users.

Target, for example, how can we stock the stores with clothing and with items that are more appropriate for the local clientele there, the people that are coming into the store? How can we make sure that the right sizes are in stock? The right styles are in stock for the type of population that we have in area? Instead of just the generic kind of genes going to every store? That's something that you can do with machine learning. That's something that you can do with your existing point of sale data if you analyze it and using the demographics of that area and then optimizing your supply chain, and it leads to a whole host of benefits, both topline benefits, because your users are walking in and they're immediately noticing, "Oh! The thing that I want, this Target knows that that's what I want." As well as in terms of like if you're shipping that good line through your mobile app. Well, now it's in stock, and so it's much faster. It also reduces the amount of returns that you have. Your store inventory is not going to waste. So you're saving cost as well. So you have to look at your business and think about what are the things that I can do, and then choose a provider or choose a tool or a database or software. I think it's all in the service of the business goal.

[00:13:54] JM: I would agree with that. However, many decisions are made within a company, whether they are wise or not, to do some kind of refactoring or migration or breaking up the monolith into microservices. It seems like sometimes these decisions are made because the company like can't think of anything else to do, or just things that we need to refactor this thing or we need to write a bunch of unit tests for this thing. We need to modernize our old applications. We need to modernize our old workflows.

It seems like there are some cases where the company should actually just leave its infrastructure where it is for the most part and it should do what you're saying. Look at your exhaust data, build greenfield applications, focus entirely on building cool, new things as long as you can keep your old legacy servers running. Because in many cases, you have limited resources. You can't do both. You can't go for the cool, new greenfield application while also modernizing your old application, while also migrating your monolith to whatever new monolith

running software or microservices-based approach you want to do. So how do you choose between those two opportunities? The painful refactoring and modernization, versus the greenfield data exhaust opportunities?

[00:15:27] AS: Yeah. That's not what I'm saying. I'm not saying that you should only do greenfield or new development. Actually, the development that I'm talking about, these examples that I gave, they are on existing systems in these hundreds of years old companies both at Loblaw as well as with HSBC as well as the auto insurance that I was talking about. These are with existing systems, existing data, building on top of them and making more out of the existing systems that you have by building new capabilities, and in some cases, transforming existing capabilities, definitely transforming existing capabilities. I don't think you want to wait until something is broken.

Actually, I would say my experience has been that too many companies, in fact, wait until the monolith gives out and they wait until the customer is complaining that the monolith does not scale or the monolith not scaling leads to a customer issue. Then they think about refactoring the monolith, which is actually quite painful, because if you wait that long, then you sort of have to put that whole project into sort of this freeze mode while you're refactoring it and you can't add any new features.

So it is important as an IT leader to have foresight and think about you're going to grow 2X, or 3X, or however much you're going to grow. Will your systems keep pace? You have to have IT systems that are going to be useful and applicable and scalable for the next five years. You have to do that investment upfront.

So, I don't think administrators are taking on modernization projects just for the heck of it, just for modernization. That's not what I see. What I see is that their developers are often asking for greater speed, greater agility, and if they cannot get it on-premise, they're moving to the cloud and they're skipping over the systems that are on-premise and it's a little bit unfortunate, because the systems that are on-premise are often running the majority of the transactions for the company. That's where all of the – And those are actually fairly reliable systems and they have a lot of the data.

So, there are ways to modernize those systems, and this is where a lot of our investment in Anthos is. A lot of our investment with GKE on-prem, which is part of Anthos, is bringing a modernization platform on-premise, taking what you have and modernizing it in pieces. Not all at once, but at the pace that makes sense for you.

So, for example, if you have legacy systems, at least converting those systems into services so that you can access those systems as services from both the public cloud as well as from other tools that you may be running on-prem.

[00:17:59] JM: Can you explain that in more detail? What do you mean by – When you say convert to services, what would be an example of [inaudible 00:18:04] getting converted to a service?

[00:18:06] AS: Making sure that there are APIs that are accessible from any kind of system, and then that you're monitoring those capabilities in a common framework. So, for example, with Anthos, if you're writing microservices, you're monitoring them and setting policy on them and securing them in a certain way. If you can also do that with your virtual machine-based apps with your VM-based apps, which maybe legacy applications, and maybe also with your old school maybe mainframe application.

[00:18:34] JM: Right. So like a COBOL app. Like you got an old COBOL app. You've got it running on-prem. You want to "modernize it". What exactly are you doing? Are you putting like a shim in front of it that makes it more accessible or like do you try to actually break up the COBOL application? What more specifically are you doing to turn what might have formerly been a monolithic application into a "service"?

[00:19:03] AS: Yeah. So it depends what the application is and whether it is currently constrained in terms of scalability. For example, some of the things that you can do, if it's a virtual machine-based app, you can still use something like Istio and roll that application into a service mesh so that you can monitor the traffic that is going to that application and set policies on that traffic. That's kind of step zero.

Step one is if it's an application that can be containerized as a monolith without breaking up the monolith, containerizing it. We have introduced this technology called Anthos Migrate, and actually you can containerize legacy databases like Sybase or even an Oracle database, and containerize it as a monolith. That's not a permanent step. That's an interim step. For things that are less mission-critical than an Oracle database. Certainly, I think very advisable to –

[00:20:01] JM: Sorry to interrupt, but why wouldn't that be a permanent step? Why not just leave your monolith in a big container? Who cares?

[00:20:07] AS: Yeah. Generally, there are several benefits of containerization. One of the benefits of containerization is that you will get bin packing. So you'll greater utilization because you can have multiple containers on the same machine. That is something that you could do if you containerize a monolith. It will cost you less, because you'll be able to bin pack it.

The other benefit is portability, and that is also something that you get with containerizing a monolith. You don't have to break it up. You can now get a certain level of isolation and independence from the underlying operating system and you can move it to a differ operating system, potentially move it to the cloud. So those benefits you can get.

However, the scaling benefit, that you are able to shut down a container and move your traffic actually to another container and still have that state available. That benefit of being able to – It's both a scaling benefit as well as a reliability benefit. So when you have a microservices or a containerized system, you'll spin up many containers and your service is distributed across all of those containers, or what we call pods in Kubernetes, and we kill pods all the time and we start new ones.

So, for example, if a particular node were to become unavailable, we could just retire that node and move the pods to a different node, and this is a reliability benefit. So, generally, users find that when they move to Kubernetes, their application becomes more highly available, because it's more resilient to the underlying infrastructure platform.

This is a benefit that you would get with a monolith that you containerize, because that application may not be able to work as well in that kind of environment where you're killing the

pod and then restarting a new pod, right? So the way that we containerize with Another Migrate, it's actually migrating into a stateful set.

So it is somewhat resilient, but it all depends on the application, and it is much better if you actually break the monolith and move to a microservices architecture. When you move to a microservices architecture, you're inherently rewriting your application such that it can run on distributed system. So it isn't assuming that your VM is always up. You don't have to reboot the same VM. You can move the application. Not just move it, but it stays up. There's no downtime involved in that movement. So that is the benefit, that reliability benefit of Kubernetes is something that you get mainly by rewriting, and there are other benefits. You can get into the dev ops tool chain, right? Be able to release several times a day. Be able to do CICD in an agile fashion, those kinds, and be able to publish the image to a registry and then deploy in other places. Those things, in order to that effectively, you have to break the monolith.

[SPONSOR MESSAGE]

[00:22:54] JM: ExpressVPN is a popular virtual private network. ExpressVPN is useful for getting a private, secure, anonymous connection for your internet browsing. It encrypts your data and it hides your public IP address. You've got easy to use apps that run seamlessly in the background on your computer or your phone or your table, and turning on the ExprsesVPN protection only takes a single click.

If you use ExpressVPN, you can safely surf on public Wi-Fi without being snooped on or having your personal data or your Bitcoin account information stolen. For less than $7 a month, you can get ExpressVPN protection.

ExpressVPN is the number one VPN service rated by TechRadar, and it comes with a 30-day money back guarantee. You can also support Software Engineering Daily if you check out expressvpn.com/sedaily. You would get three months free. Everybody needs at VPN at some point in their life. So if you want to get your ExpressVPN subscription for free for three months while also trying out ExpressVPN and supporting Software Engineering Daily, you can kill all those birds with one stone by going to expressvpn.com/sedaily.

Thanks to ExpressVPN for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:24:33] JM: Okay. Really. But it seems like even if you just ported it – I mean, the monolithic issue of state management does seem like an issue, because if I have a big monolith that I put in the container. That monolith is probably serving long-lived transactions, and if I try to have a containerized monolithic system that's handling long-lived banking sessions and I port it to infrastructure that expects things to fail all the time, then my long-lived user sessions are going to have sudden failures that I can't handle.

If I put it in a stateful set, maybe there's more reliability. I don't exactly know the dynamics of a stateful set well enough to know what the reliability would be if that's like what I'm porting it to versus its previous life as sitting in a virtual machine. I don't know what consistency guarantees or whatever would change on. But it does seem like in any case I would – if I'm containerizing it, I would have access to continuous delivery. I would have access to a container registry technology. I would have access to the interface of a service mesh, like Istio. But you're saying that if I don't break up that containerized monolith, I am not going to have access to some of those technologies. Why not?

[00:26:01] AS: You have access to those technologies. You definitely have access to those technologies, and the reliability is not any less than it would be on a VM if you're moving into a stateful set, which is what Anthos Migrate does. Because Kubernetes does have basic storage management. So Kubernetes will take care of, as your pod moves, reconnecting the storage, mounting the right storage and so forth and maintaining the state of your application.

However, if you use a non-monolithic, like a distributed database or a distributed stateful application, then you get the additional benefits of the higher reliability that such a system has when running on Kubernetes. So it's not less reliable by moving the monolith into GKE, but it can more reliable by breaking the monolith into something that is actually higher resilient to failures and is kind of a clustered database.

Also, the tool chain. Sure, you have the tool chain with the image registry and CICD tooling that if your images are monoliths rather than being broken into pieces, then you can rev only the monolith, right? Versus if your application is broken into microservices, you can independently rev those pieces. You can rev, say, the authentication service. You can replace the authentication service with a better authentication service. You can rev just the user profiles piece. You can see the relationships between the different pieces and optimize them.

[00:27:34] JM: You've given a good outline for the context that these digital transforming companies are going through and some of the issues that they're dealing with. I think really well exemplified by the story of a monolith that you have to figure out how to deal with. These companies have not just one monolith. There's like probably hundreds of monoliths at a bank that's 100 years old and they need to figure out standardized procedures for getting these monoliths into a state where they're more usable. They need to figure out their strategies for working with cloud providers. They need to figure out long-term strategies for dealing with service integrators. Because in certain occasions, they're going to need to bring on employees from Accenture, or Deloitte, or some other consultancy.

I think this is the perspective that Anthos was created in. Can you describe to me the vision for Anthos and what Anthos is?

[00:28:40] AS: So Anthos is a hybrid cloud platform that is consistent that runs in your environment, in your data center, that runs in our cloud and Google Cloud and is capable of running in other clouds. Providing you a consistent environment in which to deploy your applications and treat them and manage them as services. So that's in a nutshell what Anthos is, and it provides a number of capabilities for enabling that paradigm.

But, fundamentally, that paradigm of running your applications as services in a consistent fashion that can be ported from cloud to cloud, that enables a level of efficiency and a level of modern development that I think is transformative to most enterprises, because it allows them to have the freedom of choice in terms of where they run, which users they're close to, and their applications are close to and what infrastructure they run on. Making sure that it is best for their regulatory and compliance requirements. That it is best for where their data is based. It's best for where their users are based. It's best for optimizing their cost. It's best for optimizing their

choice of types of services that are available. Whether those are cloud services or those are services that are on-prem.

So those are the benefits of Anthos. I think it is a game changer as far as modern application delivery and modern application development in the environment that your business requires. That's the benefit of Anthos. Then, specifically, what enables you to do is, obviously, it is based on GKE. GKE in the cloud as well as GKE on-prem. GKE is the longest running and most stable and most advanced Kubernetes offering that is available today. It's been around since November of 2014, since the early, early days when Kubernetes was first created and open sourced, and it's built by the same team that wrote Kubernetes at Google and maintained by that team. It has a lot of advanced capability for security, for reliability, for ease of use, so that you can write and host and develop your applications reliably, and that is the piece that we've brought on-prem for your own data centers.

So that's one piece of fundamental piece is the consistent platform. The second piece is service management. So the service management piece is through Istio capability and then hosted capability in the cloud. What that does is allows you to view all of your microservices as well as non-microservices as essentially layer 7 services. You can monitor traffic, control traffic, set quotas and then set security policy on those services.

Then lastly, Anthos has a marketplace from which you can obtain applications that will run in this environment, that will run on GKE, GKE on-prem. It basically will run on Kubernetes, anywhere, and those are applications that are published as containers in the marketplace and deploy into your clusters and are supported either through Google or through third-parties that are publishing in the marketplace.

So that's the kind of basics. Then there are a number of other additional capabilities that are multi-clustered capabilities, and that's the piece that I think is extremely unique and exciting about Anthos, is it's not just one cluster or a set of clusters that you're managing individually, but the ability to manage those clusters together from a single set of policies and a single set of logging and monitoring and update capabilities.

So you might have clusters on-prem. You might have clusters in the cloud, and you can upgrade them and view them and deploy to them from a consistent single pane of glass, and you can set policies on them, RBAC policies, quota policies, Istio policies from a single source of truth. So it just kind of simplifies management.

If you think about how this is used in real-world. So, we talked about some of the legacy companies, older companies. They have data centers on-prem and they want to be able to manage across on-prem and in the cloud from a single source of truth, both from a policy perspective as well as a deployment perspective.

You also companies that have a lot of locations at the edge and they have clusters at the edge. So a lot of retailers have clusters that they need to deploy inside warehouses or inside stores, or banks have branch locations. Sometimes these are hundreds of clusters. Being able to manage them from a single source of truth is a game changer. It isn't something that is provided anywhere else. So Anthos brings that capability to those kinds of environments.

[00:33:33] JM: If you were to say that to a grizzled CTO who's been managing a bank for 20 years, they might say, "Okay. That's a great story. I heard this from OpenStack. I heard this from Cloud Foundry. I heard this from all the other management layers that have been sold to me over the years, and they work. They do a great job for a while, but eventually they age out. I need to build some different layer of abstraction. It never seems like my dreams of having this single pane of glass get fulfilled." Why is Anthos different?

[00:34:16] AS: Yeah. I think that's where the distinction is that is this is a system that you essentially take on and you manage it yourself? Because if you do, if it's a system that you take and you manage in your data center in isolation by yourself, which has happened to a lot of these modernization system, then it sort of falls into this practice of it's upgraded only when it's convenient and each sort of installation or cluster is managed separately, or set of clusters is managed separately. Every data center is managed separately, and things fall out of sync and they become obsolete in some cases.

The difference with cloud is that there is a connection to the cloud and there is a centralized management by Google in this case, which is permanent. It lives forever and is constantly

monitoring your systems for security vulnerabilities. It's constantly pushing patches to you. It's constantly upgrading and providing the latest capabilities.

[00:35:21] JM: This is [inaudible 00:35:21] GKE on-prem. So there's like a relationship.

[00:35:23] AS: That's right. It's a service.

[00:35:25] JM: Oh, it's a service. So you deploy it on-prem and it's talking to the cloud.

[00:35:29] AS: That's right. It's a service that's not – Then that's the fundamental difference with Anthos, is that it's a managed service that is in your environment much like a cloud service. So that's the difference with a cloud service. Cloud services, when you're using services in the cloud, they are always changing. They are always upgraded. They're always the latest and they are being managed. So they don't become obsolete. They don't go out of sync with your requirements, and that's the difference, and I think that's a fundamental difference in how software is being delivered on –

[00:36:03] JM: That is different, because I think these previous iterations, the previous iterations were always, "I deploy this thing. Maybe I have some consultants that come through every two months and like help me upgrade it or something." But the idea of having a service that is on-prem, like GKE on-prem, it's running my workloads on my servers, but it has a relationship with the cloud so that it updates Kubernetes and whatever else I'm using. Okay.

[00:36:32] AS: That's why people are excited about GKE on-prem, because it's GKE. GKE is managed. GKE is upgraded. GKE is always the latest. GKE is always getting new features. It is SRE supported and so forth. That's why people are excited about this. It's not just software. It is software as a service.

[00:36:51] JM: Is GKE on-prem open source?

[00:36:53] AS: GKE on-prem is like GKE. It is taking open source and it's a service that's based on open source. It is not open source in the sense that the code that runs GKE, that runs Kubernetes, is not open source, but Kubernetes itself is open source.

[00:37:09] JM: Is Anthos open source?

[00:37:11] AS: Anthos is open. Anthos is a collection of components that are open source with a number of manageability capabilities added in.

[00:37:19] JM: Okay. So the management layer itself, that's not open source. Okay. So what if I'm one of these enterprises that has been digitally transforming for 10 or 15 years and I've already adapted AWS and I've already got a bunch of AWS resources. I've already ported some of my applications to VMs running on AWS.

So part of my infrastructure is AWS. Part of it is my on-prem infrastructure. Now I want access to Google resources, or is the Anthos story that I need Anthos managing my AWS stuff as well? What's the story there for going hybrid plus multi-cloud?

[00:38:05] AS: Yeah. A lot of our users and customers are coming from AWS. They have deployed something in AWS or are using AWS in some way, shape or form, and they are looking at having another cloud provider because they find certain services and GCP that are of interest to them. Whether those are machine learning services or often, actually, it is GKE that they are interested in, because it is a more advanced Kubernetes offering and they want to have access to that, or they are just interested in having some level of resilience or redundancy in terms of what they have in the cloud, or in some cases they're interested in migrating back and forth. Lastly, there are some banks actually that want to have the ability to be able to move their applications back and forth between clouds. It's kind of regulatory requirements.

Actually, we're Europe, and Europe, you sort of have to prove that you can run in more than one cloud. So your board or your –

[00:39:00] JM: You can prove that you can run everything in more than one cloud or just like parts of your application?

[00:39:05] AS: You have to prove that you can run some of your critical systems in more than one cloud, because it's a risk reduction. It's a risk reduction requirement from the regulators. So

– Because what if there's a failure or there's an outage? You have to prove the resiliency. Then, lastly, many of them need to have certain restrictions as far as which locations they keep their data or their infrastructure. In some cases, they are in geographies where is no cloud and they have to have data located there.

So, those are some of the reasons why people, customers are using Anthos both on-prem as well as are using Anthos so that they have that insurance that they can run it in a different cloud.

[SPONSOR MESSAGE]

[00:40:00] 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 CICD 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 CONTINUED]

[00:42:07] JM: Do you use Anthos to manage stuff in AWS or is it like if I'm the CTO of this company, I have multiple single panes of glass. One is my AWS single pane of glass and one is my Anthos Google pane of glass.

[00:42:25] AS: Yeah. So what we have right now, which is an alpha, it's an early access program, is to be able to connect your clusters, your Kubernetes clusters that maybe running in AWS or maybe running on-prem to our console. So as I mentioned, the multi-cluster management capability. It has two layers. You can do multi-cluster management at the cluster level, which is you can upgrade all of your clusters. So you have GKE on-prem in various stores, and in on-prem, and in GKE. You can manage all of those from the Google Cloud and be able to upgrade all of those.

But those clusters are – Currently, we only offer GKE on-prem on-prem and in Google's cloud. We don't offer another cloud. But what you can do –

[00:43:13] JM: You could do that, I guess? That would be cool, right? Like on the AWS marketplace, you could offer GKE on-prem, right?

[00:43:20] AS: We could think about that. What we do offer in alpha or early access today is if you have your own clusters. Let's say a CUPS cluster that you've deployed on AWS. You can connect that to that same console that I'm talking about, that centralized console, and you can deploy workloads to those clusters, and you can monitor those clusters.

[00:43:39] JM: [inaudible 00:43:39].

[00:43:41] AS: Through the multi-cluster management plane in Google Cloud.

[00:43:46] JM: I see, but it's able to interface with AWS because it's just using like RESTful APIs or like –

[00:43:57] AS: It's through the Kubernetes API.

[00:43:58] JM: Kubernetes API. Okay. Got it. Then do you know to what extent you – This is getting super into the weeds, but do you know to what extent you could potentially spin up AWS resources through the Google Cloud console? If I talk to my Google Cloud console and it talks to my AWS Kubernetes thing, do the APIs reach far enough to be able – You can't do that. You couldn't like spin up a DynamoDB –

[00:44:29] AS: We don't provide any such capability to spin up resources in a different cloud. No. Most customers build their own scripts or that they have their own provisioning systems that do that, and you could certainly exercise those systems remotely, but that's not something that is part of Anthos.

[00:44:43] JM: Would you want to do that? I mean, strategically, is that desirable?

[00:44:48] AS: There's no strategically. I think it all depends on what customers ask for. So far we've had customers ask for, "Hey, we really like GKE. We want to run it in our environment. We like to also have the benefit of being able to have a single pane of glass across all of our clusters. We want to do multi-cluster management," and that's what we're providing.

The policy management capability is also something that you can run in any Kubernetes cluster. A lot of the technology that we've built can be run on any Kubernetes cluster. So that's why it makes sense to provide it. Like I said, for all of the reasons that I mentioned, customers are interested in running both on-prem and in the cloud and, increasingly, in multiple clouds and being able to manage it in a way that they don't have to train their people twice or thrice to be able to deploy applications and manage applications across clouds. So that's the requirement that we're fulfilling.

[00:45:39] JM: Google has a strategy of being the open cloud. What does that mean to you?

[00:45:47] AS: So, Google has a very long history with open source. I think since our founding, we have been contributing to open source. We have been also open sourcing our core technologies consistently overtime and it's something that is in our DNA and it is something that we find rewarding to engage with the open source community and to provide software that is

transformative as open source so that the maximum number of users can benefit from that software and adapt it, and that we also get feedback about usage and are able to improve the software.

So when we say we're the open cloud, a lot of it has to do with that respect for contribution, that participation in the open ecosystem. I think, lately, it translates to bringing a number of open source software vendors and providers and enabling them to run on Google Cloud and with a viable business model. Providing them basically a platform on which they can reach a very large set of users. Potentially a platform on which they can reach users across clouds and certainly on-prem as well.

So Anthos, in future, can be a platform where software vendors that are writing open source software have access to a very large set of users. So I talked about the marketplace, the Anthos marketplace. There's a number of companies, both legacy companies, legacy ISVs, software vendors, as well as open source software vendors that are writing applications that can run on Kubernetes. I think it's a really good platform to run applications. So you think about companies like Redis Labs and InfluxDB and Data Stacks. There's a whole number of sort of modern applications that run really well on Kubernetes and they're publishing their containerized offerings in the Anthos marketplace. Then that allows users that are using Anthos, again, as a consistent environment, on-prem or in Google Cloud, or elsewhere, to deploy those applications.

So I talked about clusters, your CUPS clusters in AWS that you're connecting up to this control plane. You can use the marketplace to deploy applications into those CUPS clusters. So it gives the ISVs a much broader market. Then it gives users access to a marketplace of applications. So it's a win-win.

[00:48:15] JM: Oh, can you not do that on AWS? You can't deploy to your own Kubernetes clus – You can't like deploy an InfluxDB managed by the InfluxDB company, InfluxDB cluster, to your Kubernetes cluster on AWS?

[00:48:30] AS: I don't know if there's a Kubernetes marketplace that you can use. What I'm speaking about is the Anthos marketplace. In the Anthos marketplace, there's a level of certification and we're building a marketplace that's specifically for Kubernetes.

[00:48:43] JM: Okay. That's cool. So is the vision for Google's open cloud to open source everything, or is it to just selectively open source things that differentiate Google Cloud?

[00:48:58] AS: I mean, what to open source and when to open source is – Like it's a case by case decision. So there's no one answer to that. You don't open source everything. Obviously, I don't think it makes sense to open source things that maybe are plumbing that you're using for your own infrastructure. There's a lot of work that we do that is about our network or about the way that we run on compute offering or our storage offering. Majority of the work is in building the cloud. That's not really valuable as open source, neither to our users nor to the company. Maybe there's some use probably from an academic perspective. You can write papers on those things.

But I think open source is useful when you want to make sure that there's a technology that is kind of a better way to do things. When you think that the world is sort of ready to engage in that technology. So I think containers, for example, Google has been running on containers for 15 years and we've optimized the ability to run on Containers with Borg over that time and there's a lot of knowhow that has been developed and it wasn't actually developed in a way that it was to be used externally.

But at some point, we had the realization that actually the world is starting to embrace containers. They are starting to move beyond the VM abstraction and are ready for containerized applications because of the developer productivity boost, because of the portability capability that containers provide. So when Docker came on the scene, I think we realized that, "Yeah, the world is starting to use containers," but there isn't really a good paradigm for managing containers, for orchestrating containers, and we knew that that is actually critical for running containers in production and having it run at the scale that it can run at. That's why we then thought about taking the learnings from Borg and writing them into a system that is useful for the external world, and that's how Kubernetes was born, and that's the spirit with which it was created.

We decided that that is a capability that made sense and the time was right to bring it to kind of an open source standard, to set on open source standards, so that technology would be available to the maximum number of users. That's how we made that decision for Kubernetes.

[00:51:21] JM: What has changed between this KubeCon and the last KubeCon?

[00:51:25] AS: KubeCons are happening fairly frequently. So the last –

[00:51:27] JM: Yeah. You're in North America, right?

[00:51:29] AS: I've been at, I think, all of the KubeCons. Obviously, there's a lot of growth every time there's more users. I think in this KubeCon, there's a thousand users [inaudible 00:51:40]. I think since the last KubeCon is probably what doubled – I think has probably doubled in terms of the number of users. So there's going to be a lot of companies.

I think the biggest change in my mind is just the number of large enterprises that are using Kubernetes. The number of applications that are available on Kubernetes, including legacy applications that are being containerized. In my mind, the work that we're doing with Anthos Migrate to containerize existing applications is perhaps the biggest change, and it's still yet to play out completely. So it's just the beginning.

Then the maturation, I would say, of service mesh type technologies, as well as of the developer experience with Knative. Running both serverless as well as just containerized applications, stateless applications that basically you write the code and then it self-deploys into a Kubernetes cluster and self-scales based on the traffic without the developer or the operator having to do anything with regard to the settings underneath, with regard to exposing the services and so forth. You don't have to do any of that.

So the developer experience is getting much better and it is getting more standard. I think that's just starting. It will continue to happen over the next couple of KubeCons.

[00:52:55] JM: Okay, last question. At this point, I've been to KubeCon. I've been to Google Cloud Next, and I've been to Google I/O, and it feels like from KubeCon, to Google Cloud Next, to Google I/O, I'm moving from the past towards the future, and I can imagine a world in which these legacy companies look and operate at the high-level of abstraction that you see at Google I/O, or you see at Google I/O, you see like the kind of demos, like the assistant demos. I want that world. I want the world where I can talk to my banking application like an assistant. Do you get that same feeling of moving between the past the future when you're going to KubeCon versus going to Google I/O?

[00:53:59] AS: Yes. For me, it's a lot less about the conferences. But the movement from the past to the future is the reason that I'm here. It's the reason that I work in enterprise software, because I do think – This trend started a while ago, the consumerization of enterprise software. Because if you think about it, consumer software is so much more usable. It's so much more current. It's so much more fun. It's so much more relevant, and I think therefore more powerful in terms of its revenue potential for the businesses that are running it. It's bound to happen that enterprise technology will make use of those paradigms, and we are actively involved. I am personally actively involved in enabling that change, and that's what's exciting about the role that Google has. That's what's so exciting about cloud, is that the technologies that we are introducing are born out of that fundamental capability to run software in a much more personalized, in a much more intelligent, in a much more scalable fashion. That power is there. Whether you're a startup, it's available to you, whether you're an enterprise. Then it's just a question of who will use it?

I think if you're a large enterprise sitting on all of these data and sitting on all of these services and with all of these users, someone's going to pick it up in your industry and they're going to use it and they're going to have that competitive advantage. So can you afford to just sit by and wait? I don't think you can.

[00:55:28] JM: Aparna Sinha, thanks for coming back on.

[00:55:29] AS: Thank you.

[END OF INTERVIEW]

[00:55:34] JM: GoCD a continuous delivery tool created by ThoughtWorks. It's open source. It's free to use, and GoCD has all the features that you need for continuous delivery. You can model your deployment pipelines without installing any plugins. You can use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your cloud native project. With GoCD on Kubernetes, you define your build workflow. You let GoCD provision and scale your infrastructure on-the-fly, and GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn how you can get started.

GoCD was built with the learnings of the ThoughtWorks engineering team, and they have talked in such detail about building the product in previous episodes of Software Engineering Daily. ThoughtWorks was very early to the continuous delivery trend and they know about continues delivery as much as almost anybody in the industry.

It's great to always see continued progress on GoCD with new features like Kubernetes integrations so you know that you're investing in a continuous delivery tool that is built for the long-term. You can check it out yourself at gocc.org/sedaily.

[END]