EPISODE 635

[INTRODUCTION]

[0:00:00.3] JM: Enterprises want to update their technology faster. One way an enterprise can accelerate the adoption of new tools is to move more aggressively towards the cloud. By giving internal developers access to the cloud, it becomes easier to provision new servers allowing for rapid experimentation, test environments and scalability.

In previous shows, we have explored how large enterprises successfully learn to move their technology faster. Much of this process is rooted in being able to experiment quickly, which requires well-defined testing procedures and the ability to quickly provision and destroy infrastructure. Many enterprises have large on-premise infrastructure deployments. An enterprise's movement towards the cloud can be made complex by this existing set of servers.

In today's show, Aparna Sinha discusses how Kubernetes is useful for enterprises and how it can improve development speed, experimentation and observability. Aparna is the leader of the product team for Kubernetes and Container Engine at Google. Much of her job is centered around understanding what would be useful to enterprises who are choosing a cloud provider. The open source version of Kubernetes is useful on its own, but most enterprises choose a managed provider of Kubernetes such as Google Kubernetes Engine to help with support and onboarding.

Full disclosure; Google is a sponsor of Software Engineering Daily.

[SPONSOR MESSAGE]

[0:01:33.2] JM: Citus Data can scale your PostgreS database horizontally. For many of you, your PostgreS database is the heart of your application. You chose PostgreS because you trust it. After all, PostgreS is battle tested, trustworthy database software, but are you spending more and more time dealing with scalability issues? Citus distributes your data and your queries across multiple nodes. Are your queries getting slow? Citus can parallelize your SQL queries across multiple nodes dramatically speeding them up and giving you much lower latency.

Are you worried about hitting the limits of single node PostgreS and not being able to grow your app or having to spend your time on database infrastructure instead of creating new features for you application? Available as open source as a database as a service and as enterprise software, Citus makes it simple to shard PostgreS. Go to citusdata.com/sedaily to learn more about how Citus transforms PostgreS into a distributed database. That's citusdata.com/sedaily, citusdata.com/sedaily.

Get back the time that you're spending on database operations. Companies like Algolia, Prosperworks and Cisco are all using Citus so they no longer have to worry about scaling their database. Try it yourself at citusdata.com/sedaily. That's citusdata.com/sedaily. Thank you to Citus Data for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[0:03:18.5] JM: Aparna Sinha, you are the group product manager for Kubernetes at Google. Welcome to Software Engineering Daily.

[0:03:24.7] AS: Thank you. Glad to be here.

[0:03:25.6] JM: I want to talk to you about Kubernetes and its relation to enterprise. So there are large enterprises that want to update their technology more quickly. Many enterprises that we've had on the show that have legacy technology, they're in this situation where they can't iterate as quickly as they would like to. So this is a prototypical problem. I'm sure that you have encountered it in enterprises that you've talked to. What are the problems that these enterprises run into at a fundamental level when they're trying to move quickly?

[0:04:02.4] AS: Jeffrey, as you know, software is becoming more and more critical to the competitiveness of every enterprise in every industry. So the desire to move to new technology and to iterate on their existing software set is really rooted in business outcomes and the desire to be more competitive. But as a large enterprise, there can be many obstacles to moving quickly and they come in many forms. I guess I'll highlight three.

One is if your internal IT systems and IT processes are based in the traditional way of releasing software, then they typically require quite a few approvals. They may have some manual processes, where in order to provision new software or develop new software, you need to actually order hardware and provision virtual machines and install operating systems before your developers can even get started with rolling out applications. That usually leads to some delays.

The other sort of day two problem is keeping the software that you have up-to-date and current and therefore being able to utilize the latest innovations in your field, and that's pretty critical if you want to be competitive. But enterprise software, especially when you have kind of a lot of process, tends to get stale. Then it's difficult to upgrade it, and then that's another reason why enterprise companies fall behind in terms of what they have in their tech stack. Those are two reasons.

I think the third one really has to do with talent and capabilities. These days, developers are very much at a premium and it can be difficult to hire the best developers or to hire developers that are up to date with the latest technologies and, ultimately, that determines the competitiveness of your entire enterprise, and certainly of your IT capabilities. I think those are some of the basic challenges that companies run into.

Then on the infrastructure side, I would say that that the platform that you choose, the stack that you choose and the processes that you build around it are some of the fundamental decisions that determine your agility not just in the short term, but over the long term.

[0:06:23.2] JM: In order to be moving faster, a lot of the enterprises have started to move a lot of their infrastructure into the cloud, and I think, initially, people were just looking at the cloud as maybe this is going to save us costs or reduce the operational burden we have by having to in-house our infrastructure management. But over time, there have been all of these other benefits that people have seen come with the cloud.

For example, you have these really beautiful dashboards that are really well-designed that the cloud providers build for people. It makes it much easier to interface with stuff. You don't necessarily have to go through a command line. You can go through like a really nice web

interface, and that makes it more pleasurable to interact with, which lowers the barrier to engaging with this stuff and spinning it up. But that over time, even the command line tools for interfacing with cloud products have gotten better. I guess Kubernetes is not exactly a cloud product, but it's more like a product for managing your own internal infrastructure like a cloud or with the assumption that you always have access to infrastructure to spin up new resources on. But I think I'm getting ahead of myself.

So for these enterprises that want to move to the cloud in order to accelerate their operations, what are the challenges that they run into? What are the hurdles that they have to overcome in order to get that situation where I'm an average developer at a large enterprise, like an insurance company, or a bank, or some kind of factory and I'm just an average developer and I have access to cloud products?

[0:08:13.0] AS: Yeah. Well, you're absolutely right about the value of the cloud, and I think it has gone through certain phases. In the beginning, there was this perception that cloud is just about converting capex to opex. Going from fixed cost for infrastructure to something that's more on demand, and that's true. Cloud is that. But I think as users started to leverage the cloud, they discovered the other benefits, agility, being able to start very quickly. You dream up a project and immediately you can start coding it and in a little while you can start releasing it and testing it and making it available to the world. So that turned out to be potentially a bigger benefit, a topline benefit versus a bottom-line cost reduction.

I think particularly for smaller companies or newer projects that have an uncertain, an unknown demand, the scalability of the cloud. So you don't have to ever over provision, and yet you are not under provisioned. You can actually grow without overpaying, and yet without curtailing your growth. So I would say cost reduction, agility, scalability, these were some of the early benefits that folks realized.

Then I think, few years ago, there was another major shift in the adoption of the cloud. When I think the world realized that the cloud can actually be more secure than on-prem infrastructure. That really has to do with the fact that the cloud is a constantly updated, very current environment. So Google, of course, excels as a security. We saw this last year with some of the

security changes that Google made even before those exploits were announced, and it is on an ongoing basis.

So customers have started to realize that actually the cloud can be more secure, the cloud can be more reliable. Those are some of the, I would say, properties of the cloud and properties of your IT infrastructure, of your IT stack that you look for. Now, customers are moving much more rapidly to the cloud because they realized its costs, it's agility, it's scalability, it's security, its reliability. Boy! I can get a lot of that at par or better than what I could do on-premise.

I think what takes it over the top is the services that the cloud provides, the innovative services. So, for example, the machine learning capabilities, the data and analytics services that we have in Google Cloud. Things like a managed Kubernetes service as well and other types of services, log service, and then build on that logs analytics service.

So it's sort of like I almost think of it like your smartphone, your android smartphone, has all of these applications and you can actually kind of put them together and do things that you couldn't do if you just have the hardware. So I make that analogy and it helps me see the true benefits of the cloud beyond just the fact that it's more secure, more reliable hosted hardware. So that's what I would say to your first point about cloud is very popular and companies are moving to the cloud.

Your second point was really Kubernetes as a cloud and what are some of the challenges that companies face in moving to cloud. I think Kubernetes is absolutely an enabler of cloud. It's a very, very close companion. The thing about Kubernetes that makes it an enabler of the cloud is, it actually enables that high utilization of the underlying infrastructure. It enables that agility where you can get started quickly to build your application. You don't have to worry about starting up a new VM or starting up a new operating system. It does have great scalability and auto scaling capabilities.

So it gives you some of those aspects of the cloud even on-premises, even in an environment which isn't actually in elastic environment, right? So you can you can use Kubernetes to achieve that agility or similar agility, similar utilization and performance and reliability. So customers often tell me that with Kubernetes, my VMs could go down and it didn't matter. My application still

stayed up. So some of the things that they noticed when they used Kubernetes is the increased utilization, the increase reliability and, of course, the ability to release software more frequently.

So that's why Kubernetes is often referred to as an enabler of the cloud. But the other piece of it, which I think is even more important, is the fact that once you are running your applications on Kubernetes, it makes it very, very easy to move to a public cloud, or to move back to on-premise or to move between cloud providers, and that's the fundamental portability of Kubernetes and of containers. That allows you that freedom. I think that is unique. Truly unique amongst the different types of platforms that you could base your applications on.

[0:13:01.1] JM: So what I'm hearing you say is that there are issues with moving to the cloud because a company already has pre-existing infrastructure. I think there's some hesitation to buy into a solution that would lead them towards where it's something that they would feel like they would be locked into. But what's useful about Kubernetes is that it is this layer of optionality and that why there's the comparison to Linux, which is a very apt comparison, where I think enterprises are fairly comfortable. Most enterprises are fairly comfortable with Linux at this point or many enterprises. I don't know about most. But many are very comfortable with Linux as a personality layer that's not going to lock them into anything specifically that they don't want to be locked into.

So could you give a little more color on why that's appealing to enterprises? Like that layer of optionality? The layer of flexibility?

[0:14:02.7] AS: Yeah. So you talked about the issues with moving to the cloud. I think there are three main issues that companies encounter. Number one, do I have the capabilities that I need to get to the cloud? Do I have the staff and the training that I need to get to the cloud? I think that's number one. Second, are all of the constraints that you mentioned? I have on-premises hardware, or most large enterprises have on-premises hardware, but it's not just the hardware. It's actually the applications. Can the applications be moved to the cloud or are these applications that aren't going to be able to run in the cloud? Maybe there are some constraints. Maybe it's the way that they're architected, or the way that they're connected together, or maybe they need to be on-premises for other reasons.

Then I think number three, and most important, is will I get the anticipated benefits? I think a lot of people, sometimes, they get the training, they work around the constraints, but if they haven't done it before, they may not achieve the benefits that they set out to achieve. In some cases, they've had to turn back their deployment. So how do you overcome that?

So number one, the training, right? If you're going to move to the cloud and the cloud environment is going to look radically different from your on-premises environment, then you pretty much have to train a new skillset and possibly an entirely new workforce that's going to be familiar with all of the technologies that are in the cloud and familiar with how to write applications and run applications in the cloud. They need to talk to your internal systems and your internal security and networking and internal controls and then your internal developers. So you kind of end up fragmenting your resources both on the operation side as well as on the development side.

The more number of clouds that you choose, and I think most enterprises, I think we have some data that says 80% of enterprises are hybrid or multi-cloud, because they want to use the best capabilities in each cloud. So the more clouds you use, the more this fragmentation of training. So that's a big problem.

Wouldn't it be better if you only had to train on one platform, and that platform went with you to all of these different environments? So you didn't have to retrain. I mean, the efficiencies that you get from the operations and developer productivity can be very significant. Secondly, the constraints – The constraints are what they are, but if you can actually work on the constraints in a more gradual manner so that you can work on the constraints in your own environment, for example, so that you can integrate your more modern, your more cloud native applications with your legacy applications, say, in the framework of your on-premises security and networking setup before you move to the cloud. That can actually be a great way to take care of the constraints. Keep in the environment where your constraints are real and in production and rooted, and then having address those with your new architecture. Move that new architecture to the cloud. That can be another way of sort of handling the constraints.

Ultimately, we call that improve and move. This improve and move strategy actually helps you get the true benefits of the clouds. So where Kubernetes fits into that picture is containerizing

your applications, writing them on Kubernetes for the things that you are developing on-premises, doing that with Kubernetes on-premises. For the things that you are already developing in the cloud, doing that in a managed service or in Kubernetes managed by you in the cloud.

Then you have kind of a one-for-one footprint in both places, in on-prem and in cloud, and then it makes it very easy to move that application from on-prem to cloud. It also – You sort of figured out how to connect to whatever needs to remain on-prem, and that connection can then just be carried over when you move to the cloud. That way, you don't run the risk of sort of moving your application as is, and suddenly in the cloud it's actually consuming more compute than you expected and it's not scaling the way that you expect it and so you're not getting the benefits of the cloud.

If you actually put in the work beforehand, before migrating to the cloud to make sure that actually that application is cloud ready, then you overcome these three common challenges of training, of constraints and actually getting the benefit, and Kubernetes is a big enabler for that.

[SPONSOR MESSAGE]

[0:18:34.5] JM: This episode of Software Engineering Daily is sponsored by Datadog. Datadog integrates seamlessly with more than 200 technologies, including Kubernetes and Docker, so you can monitor your entire container cluster in one place. Datadog's new live container view provides insights into your containers health, resource consumption and deployment in real time.

Filter to a specific Docker image or drill down by Kubernetes service to get fine-grained visibility into your container infrastructure. Start monitoring your container workload today with a 14-day free trial and Datadog will send you a free t-shirt. go to softwareengineeringdaily.com/datadog try it out. That's softwareengineeringdaily.com/datadog to try it out and get a free t-shirt.

Thank you, Datadog.

[INTERVIEW CONTINUED]

[0:19:31.5] JM: A lot of these – When we talk enterprises. Enterprises have thousands of developers and they have lots of teams of varying skillsets throughout the organization. I'm wondering how you are seeing adoption of Kubernetes make its way through an organization? For example, three years ago, we did a bunch of shows around companies adopting Docker. The ramp up process would often be a small group within an enterprise would use Docker and they would do it for something not mission-critical. So something like a job board. The classic was Netflix moving their job board into Docker or moving it into the cloud. It's like the job board is not critical to Netflix running streaming videos.

So you often see this, it's a proving of a use case by testing it out on non-mission critical infrastructure, and then they say, "Wow! This was actually a great experience," and they gradually ramp up the difficulty of what they're doing. So that's one part of my question is, is sort of how they gradually ratchet up the difficulty level of what they're moving on to Kubernetes. But the other side of the question is, when you have these giant organizations and you have different teams that are not talking to each other in the organizations, you might get different people who are standing up Kubernetes clusters in different regions of the organization. Is that a problem? Do they need to eventually merge these different clusters together? Give me the lay of the land for how these really large enterprises are adopting Kubernetes and how it's making its way through organizations.

[0:21:11.6] AS: Yeah. So there's a pattern for adopting new technology, and Kubernetes is not that much different from the way that you might adopt any other new technology. Typically, there tend to be early adopters or people that are at the forefront that are interested, and many of these tend to be developers. So it is a very popular technology in the dev ops and developer community, and they'll go out and they'll try containers and then they'll try Kubernetes and they'll start maybe a pilot.

So if you're a smaller company, a startup, then this is just something that like makes complete sense and you start with it, and I've seen companies that don't have anything else. Their entire portfolio is based on Kubernetes. But when you talk about the medium to large enterprises, like you said, there are thousand sometimes, tens of thousands, right? 20, 40,000 developers. It tends to be a couple of teams.

Developers certainly are starting. It will start to use containers before anyone else in the organization. But what we also find is that there are innovation teams or platform teams. These tend to be kind of cluster admins or these are teams of admins that are charged with, “Hey, we’re really looking for the next generation of improvements and productivity. How are you improving our ability to launch? How are you improving our ability to achieve business results?” and those are typically the teams that they will go out and they will experiment with technology.

So over the last three years, we've met teams at almost every large enterprise. I can't think of any. I mean, any industry, like I've seen them in every bank, every insurance company, healthcare more often now, and retail, and gaming, and media. Across all these industries, every company has this type of team that starts to look at what's our next generation platform? How can we improve the productivity and utilization and get our business to the next level?

That's where they typically bring in Kubernetes. Then what they'll do is they'll set up a Kubernetes environment. They'll, of course, test it out themselves. They’ll start a few applications and then they’ll make it available to their constituents, which could be other portions of the IT organizations, or it could be just developers in a particular business unit or across business units.

Then those developers will start to use this, and typically it starts the way any other thing starts. I’ll use it for my web app, or I’ll use it for my frontend, or I'll use it for an e-commerce app, or maybe I'm writing something that needs a machine learning, Chatbot or something, or I'm doing something that’s stateless. That's typically where it will start. Then they discover that I actually do more than that on this platform. Maybe I should try and run some batch jobs. That'll actually be very efficient. I’ll only use what I need and it seems like a good platform. Then they’ll find actually, yeah, there's a lot of documentation. There's a lot of history. There's a lot of tutorials. There's a lot of, actually, examples. I can do that. Well, that works. Or if they're trying something like serverless or functions, this can be a great platform for that.

Then eventually that platform team will get enough demand that they start to formalize the platform. They, of course, make sure behind the scenes that they get the security and compliance and networking capabilities. If they’re building a hybrid cloud, they’ll build a pathway

to the public cloud so that this platform can be used in the public cloud and in the on-premise environment and then they'll start to support more and more types of applications and kind of create an internal documents or set of documents about how to use the platform. Then I've seen many companies that have built an entire PaaS for their internal teams that's based on Kubernetes and for them to use Kubernetes.

So I think the thing different about Kubernetes relative to other technologies that I've seen is the rapid adoption of it, and I think that is rooted in the fact that you see the benefits pretty quickly. Your developers see the benefits pretty quickly. As an operator, you see, "Well. Okay, this is actually really bringing benefit to my organization and I know how to manage it, and it's pretty elegant how I can extend the APIs. Elegant how I can do things in Kubernetes, how I can configure things, how I can set policies and how I can provide capabilities to my developers."

So I found that a lot of these platform teams have become fans of the software They've started contributing to the open source. They've become part of the community. So they're not just customers or users. They're also – They start to become contributors or at least participants in the community and they feel this sense of ownership. So the passion around it, I think it comes from the effectiveness of the software and also just the openness of the community and of the software. So that then is a virtuous cycle. It generates more usage. Then, of course, the end users find it useful. So that's how I think Kubernetes is different and that it's like any other new technology, the pattern that you use to adopted it. But then it spreads faster.

[0:26:10.7] JM: Sometimes enterprises can be hesitant to pick up a technology when there are security risks to picking up that technology. What were the challenges of security of Kubernetes that the platform has overcome in the last couple of years and how have you seen those security challenges impact the perspective of Kubernetes from those enterprises?

[0:26:35.5] AS: Yes, absolutely. Security is an extremely important criteria for most enterprises. Particularly, as I mentioned, there were some statistics I think last year from RedMonk that 54% of the top Fortune 100 are using Kubernetes 90s in some form and there's a high percentage of financial services that are using Kubernetes 90s. These are obviously regulated industries. These are industries that have a very high bar and compliance and also security. They have a lot of customer data that's very, very business-critical.

So I think three years ago, there was much concern about using containers as a technology for security conscious enterprises. There's been a lot of progress on the technical front in the realm of security and container security. Container security, infrastructure security, runtime security and just security constructs and Kubernetes overall have made tremendous progress in the last three years. I think that is evidenced by the fact that there are now so many of the financial institutions and so many of the regulated industries that are using Kubernetes in production in fact.

I wouldn't say that it is being used at a very large-scale, the entire companies using Kubernetes. There are some, but I think it's still on the way to mainstream and massive adoption in these regulated and highly security conscious industries. But it's certainly being used in production, which means that the base capabilities are starting to be there. Some of the base capabilities, and these were mostly all introduced more than a year ago. Role-based access control was stable in the 1.8 release and for contacts. We're now working on the 1.12 release. We just finished the 1.11 release last week. So this is a while ago now that role-based access control became stable, and it was available for a year or more before that. But that basically allows you to set granular controls on who can do what in your cluster.

Then network policy, which is L7 – Sorry, size and L4 construct for which applications, which pods can talk to each other and being able to set that at a policy level at an L4. It's a basic capability. Again, I think that enterprises need combining that with L7 policy controls which are now offered through Istio. Istio is an add on top of Kubernetes that provides not only service-to-service authentication and this type of L7 security policy, but a number of other developer facing functions, like load-balancing and [inaudible 0:26:35.3] as well f.

But back to security network policy, RBAC, those are some of the building blocks. But then in every release, we've been adding more capabilities upon security policy and secrets encryption, which was released in 1.7, and in the last release 1.10 was enhanced and I think is now beta or stable. Tt those are some of the things that enterprises expect, and I think that we're more, I think, 50, 70% there in terms of security capabilities of Kubernetes for the more security conscious enterprises.

Then on top of that, if you work with a provider, a managed provider, like Google Kubernetes engine, security has really built-in to that offering. Obviously, the various types of compliance, HIPAA compliance and other types of compliance or something that the cloud provides, the JCP provides on GKE as well.

But then we also provide a very locked down image, operating system image, which is patched and upgraded regularly for any CBEs that may come out. So all of that is managed for you as a user. Then, of course, these capabilities like RBAC and net policies, Istio, Pod Security Policy, encrypted secrets, these are some things that are a part of the Kubernetes package and of course available also on GKE.

As a whole, I think it has become much more reliable, much more secure, and that is evidenced by the fact that you have now large financial services and, of course, huge retailers with PCI and customer data that are comfortable, and they feel that Kubernetes is a security used in production.

[0:30:44.8] JM: So the typical model of using a container on Kubernetes is you're a sharing a kernel, an operating system kernel with all other containers, and there are some fundamental security concerns that come from this shared operating system model. So I'd love to hear you discuss those, but also the sandboxed to container design is something that can potentially remedy those concerns, and this sandbox container design allows you to escape from this shared infrastructure layer. So can you talk about the risks or the perceived risks at least of sharing infrastructure and how the sandboxed container idea potentially absolves you of some of those risks?

[0:31:34.4] AS: Yeah. So there's security and then there's sort of hard multi-tenancy and hard security boundaries. In the case where you have, say, Coke and Pepsi on the same cluster, you want to have really hard multi-tenancy boundaries and isolation between those two types of tenants. There are other cases where even within your organization, maybe it's not multitenant, but even within your organization. You want to make sure that there is never any risk of – I mean, you can never say never, but there's hypervisor level isolation between workloads that there can be more sort of vulnerable workloads that are less secure that you want to run in kind of a more secure environment.

For example, WordPress or some of the other workloads that are more prone to container breakout or to issues, security issues, you may want to run in a more locked down environment. So that's where things like sandbox containers and the ability to have hypervisor grade isolation between pods and containers comes in.

So that is a specification that we've been working on in Signode for several months, and at CubeCon [inaudible 0:32:49.2] we introduced gVisor, which is Google's implementation for secure pods or secure containers. That is something that's open source and it's still being productized, but is available in the open source to use for that. It can be, like I said, there are two use cases. One is just to provide better secure isolation between workloads. Then the other is hard multi-tenancy, and I think the hard multi-tenancy is the more severe or the higher end of requirements where you really want to make sure that there's a hard boundary between different tenants that could be working in the same cluster. It's an added layer of security for truly security conscious and isolation conscious use cases.

[0:33:34.8] JM: How does the multitenancy question of containers compare – Because you've worked in virtualization and distributed systems for pretty long time. How does a multi-tenancy model of containers compared to the multi-tenancy model of VM's on the different axes of noisy neighbor problems and observability and security? How does it compare to your time spent just working in VM contexts?

[0:34:02.1] AS: Yeah. I think the multi-tenancy model of containers is mind-blowing. It is extremely exciting. From a distributed systems point of view, you almost get the best of both worlds, right? With containers, you get that tight utilization, that high utilization, the bin packing that you want as a multitenant provider, right? Because as a multitenant provider, you have your infrastructure and you want to bring in as many tenants and use it as efficiently so you can pass on the savings to your tenants as the system will allow, and containers in particular, the way that Kubernetes schedules and manages containers is exceptional. That's what our internal Google infrastructure is based on. It uses containers under the hood to run things like search and Gmail and maps all on the same infrastructure at a scale that requires us to be extremely cost-efficient, right?

So that technology is what's also at the heart of what Kubernetes does. So I think putting that into the hands of SaaS providers or users that are providing multitenant applications is extremely powerful. I'm very, very excited about that. I think that it is a step above what BMs can do in terms of utilization and also elasticity and flexibility.

But then there's usually a trade-off for that versus security and isolation and how much security in isolation can I get if I'm actually doing this bin packing, and that's where the sandbox containers concept comes in. So on that spectrum, on that trade-off, the sandbox containers construct takes you towards much more isolation, much more similar to what you would get with a hypervisor. So I think it's extremely exciting because it gives you sort of the best of both worlds. You can run now much more efficiently.

So I think it's a big enabler for software as a service. It's a big enabler for any customer to sort of take their application and say, "Hey, I'm going to offer this as a service to others and others can run multiple tenants, can run on my application, and I'll be able to keep them isolated and secure. I'll be able to bill for them individually. I'll be able to recognize those tenants and provide services to the tenants individually, and yet do so at a very cost-effective scale for me as a provider." It's a great enabler for that I think in a way that that is a step above what BMs can be.

[SPONSOR MESSAGE]

[0:36:39.7] JM: Cloud computing can get expensive. If you're spending too much money on your cloud infrastructure, check out DoIt International. DoIt International helps startups optimize the cost of their workloads across Google Cloud and AWS so that the startups can spend more time building their new software and less time reducing their cost.

DoIt international helps clients optimize their costs, and if your cloud bill is over $10,000 per month, you can get a free cost optimization assessment by going to D-O-I-T-I-N-T-L.com/sedaily. That's a D-O-I-T-I-N-T-L.com/sedaily. This assessment will show you how you can save money on your cloud, and DoIt International is offering it to our listeners for free. They normally charge $5,000 for this assessment, but DoIt International is offering it free to listeners of the show with more than $10,000 in monthly spend. If you don't know whether or not you're

spending $10,000, if your company is that big, there's a good chance you're spending $10,000. So maybe go ask somebody else in the finance department.

DoIt International is a company that's made up of experts in cloud engineering and optimization. They can help you run your infrastructure more efficiently by helping you use commitments, spot instances, rightsizing and unique purchasing techniques. This to me sounds extremely domain specific. So it makes sense to me from that perspective to hire a team of people who can help you figure out how to implement these techniques.

DoIt International can help you write more efficient code. They can help you build more efficient infrastructure. They also have their own custom software that they've written, which is a complete cost optimization platform for Google cloud, and that's available at reoptimize.io as a free service if you want check out what DoIT International is capable of building.

DoIt International art experts in cloud cost optimization, and if you're spending more than $10,000, you can get a free assessment by going to D-O-I-T-I-N-T-L.com/sedaily and see how much money you can save on your cloud deployment.

[INTERVIEW CONTINUED]

[0:39:03.9] JM: That sandbox that the sandbox container is running in – So I'm assuming it has some kind of engineering, some sort of fundamental restriction that prevents the application from breaking out and accessing resources that are on the same the same hypervisor, I guess? Or I guess on the same kernel is what I should be saying, right? Do you know more about the implementation of the sandbox or what that actually means? What those hard security boundaries are?

[0:39:39.8] AS: Well, I mean, the security boundaries come from looking at syscalls and filtering syscalls. So the implementation of the sandbox container, gVisor in this case, is filtering syscalls, and that is fundamentally the mechanism that's used here. There are other filtering mechanisms, like SE Linux and AppArmor that are more static filtering mechanisms. So it's not that they don't exist. They do exist, and then the hypervisor sort of kind of a more heavyweight mechanism. But there are there other mechanisms like SE Linux in Linux itself that you can use,

but they're more static and you kind of have to know what you're looking for and set the policies beforehand, but you might not know what kind of vulnerabilities there are. So they're more risky and they're more mental. But with gVisor, it's something that's more automated. It's something that we've used internally, and there's a lot of engineering that's gone into it from Google over the years. So we've been using it for quite some time.

[0:40:37.7] JM: Yeah. I probably need to do a whole show on that topic.

[0:40:40.1] AS: Yes. I think that might be useful.

[0:40:43.2] JM: To what you said about the economies of scale – I did some early shows on Docker where I would ask people about how big are the economies of scale that people are getting when they go from just a VM-based infrastructure or bare-metal infrastructure to containers that, and nobody really had like hard numbers at that point, because I think it was a little bit early.

But at this point I've done a couple shows where people have really benchmarked their spend and how much they're saving, and there's particularly a case study that I heard about on the Women in Tech Show about, I think, WP Engine, where they saved something like 50% of their costs, which is just insane. Like saving 50% of your infrastructure costs by migrating your infrastructure layer. It's just – I mean, that's going to improve your margins a lot. It's seems like something that can really change the economics of just software businesses as a whole. But, anyway, I don't need to tell you that.

[0:41:52.3] AS: Yeah. It not at all surprising to me. That's not at all surprising. Of course, I've seen the WP Engine numbers and I've worked with them and so forth. But I've had customers – It totally depends on your architecture and how much work you put into the re-architecture if you're doing a re-architecture. How much benefit you get?

But I've had customers that have said they went from 4% utilization to 90% utilization. That's the one that's like shocking to me. Like, "Wow! That's incredible. How did that happen?" But, usually, in going from VMs to Kubernetes in the cloud, you are going to see something on the

order of 30% to 50%. You don't have to do something extraordinary to get that in terms of benefits. But I've seen more.

Yeah, it's typical. So that utilization and that efficiency, sometimes you take out whole load balancers. Obviously, the utilization of the machines, the virtual machines goes up. You're auto scaling, so you're not – And your auto skill isn't actually working. So you're not using capacity that you were before.

So it's the combination of the bin packing, plus the auto scaling, plus the architecture itself requires less. So that's where you get the 30 to 50%, and in some cases 90%, depending on what your application was. If you are using, say, preemptable VMs on Google Cloud and you're running batch processes that you could get 90% kind of benefit versus running them on VMs, on-prem or somewhere else where you don't have preemptable.

So there's a range. But the cool thing is that it's not just a utilization benefit, right? Customers tell me, "Yeah! I was really surprised, very pleasantly surprised and very happy with my cost-reduction, but I also saw that my application was more reliable. That the underlying VMs could go down and it wouldn't matter. My application would stay up," and that was really pleasant, right? You can imagine that that was something that is just a very, very positive experience for users.

So I've heard that a lot and people say, "Yeah, my reliability went up. I don't have to get up in the middle of the night and fix things, and I love that." Then, of course, sort of lastly, I guess, is the developer productivity. Now I'm releasing so many more times a day, which often that's the first thing I think of. But in terms of the experience of the user, they see the utilization. They see the reliability and they see the productivity benefit. Then, of course, they also like the fact that it's portable.

[0:44:19.0] JM: So as far as the faster release times, we have done a lot of shows about continues deployment. We have some more that are coming up. I wanted to ask you about that. But there's another burning question I had. So maybe we can get to continues deployment a little bit later, but the question around the data platform.

You and I can be very excited about machine learning, and should we use Spark or should we use Dataflow or how should we get our machine learning jobs deployed? That stuff is very exciting and very interesting. But at these enterprises, there's a ton of low hanging fruit in – Let's just get Hadoop clusters running. Let's just get batch map-produced jobs going across our data. I think the Hadoop providers that came out in the earlier 2000s, like the MapRs and the Clouderas of the world and, of course, the other cloud providers were very helpful in modernizing some enterprises and bringing map-produced and straightforward data platforming into these enterprises. But there are still like challenges, at least I get the sense. There are challenges at the data platform layer that can be improved by Kubernetes.

So when we're talking about the data platform, what are the ways that Kubernetes can help some of these enterprises like a big insurance company develop its data platform?

[0:45:52.7] AS: Well, there's a couple of ways. I would say what is sort of being done today is just sort of have your Kubernetes environment. It could be in GKE, it could be on-prem, and you may be integrating with an existing data warehouse or an existing Hadoop cluster, right? That's sort of, I think, the traditional way. You're doing some batch processing in Kubernetes and then you're doing some ETL, say, on-prem with your data warehouse or with your Hadoop cluster and that's happening at some frequency that makes sense for your business.

That's kind of not using Kubernetes as a data platform or only using it as a part of your data platform. The other is to run stateful services on Kubernetes. We have invested about two years to build up the capabilities. So, originally, I think when you think about containers, they weren't really thought of as and they weren't capable of running stateful applications. That's sort of more of a recent development I think over the last two years where we've invested quite a bit of time in running stateful applications on Kubernetes.

So the basics are there. The primitives are there. So you're not just running web services. You can run batch. You can run machine learning and you can run stateful services, like Redis, or Elasticsearch, or MongoDB or MySQL or a whole host of sort of database like or data store like services, ZooKeeper, Kafka, obviously, etcd. So you can run these things on Kubernetes 90s. Then when you run them on Kubernetes, they obviously have all of the benefits of the efficient

infrastructure, the auto scaling and the reliability piece. But they are architected differently than stateless applications.

So providing the primitives to run these types of stateful services on Kubernetes is what we worked on over the last year or two years providing the primitives, so the primitives in terms of stateful sets and in it containers and the jobs capability and the batch API. Those primitives are stable actually. We've matured them over the course of the last two years, and now what we're doing is we're encapsulating a lot of that capability into what are called operators or application operators.

An application operators sort of automate the deployment of stateful applications in your cluster and they extend the Kubernetes API to provide you an API endpoint that understands – That speaks the language of that application. With further development, we're actually building application lifecycle capabilities into the operator. So the operator will actually upgrade and do maintenance of your application.

These are – And we've picked certain applications to begin with. So like you mentioned, Spark, and I think also Tensorflow, and Airflow, and Redis. It's based on sort of what's a good fit for Kubernetes, plus what is their sort of immediate demand for. Hadoop isn't yet – It isn't a good fit for Kubernetes. So I think that hasn't – That is sort of at the level where we're integrating with it, but not really running it

Spark is and has been a very good fit. So we have native support for Spark, and Kubernetes is a native scheduler for Spark. Then cluster databases obviously are natural fit or cluster data stores are a natural fit for Kubernetes. But then also things like MySQL, CockroachDB and things that are – And MariaDB and kind of more traditional kind of databases as well. Then, of course, Redis and Elasticsearch. These are things that people tend to use very commonly with what they're building on Kubernetes.

I think, over time, we may find that Kubernetes expands to be a more universal platform. That's certainly – It's come a long way already. It may expand in other ways to be a more universal platform. There's still more engineering, I think, to be done.

[0:49:40.8] JM: How has your role changed as the Kubernetes project has scaled?

[0:49:45.7] AS: My role? I lead product for Kubernetes and it has been an extremely exciting ride, I would say. I've enjoyed very much bringing this project and this product up and being there in the early days and bringing it up to kind of a more mature project where there's getting to be widespread usage.

My role I would say and my team's role has shifted from guiding the initial technology, working very closely with engineering to bring some of the innovation to market and explaining the innovation and changing mindsets, evangelizing and kind of describing the benefits of containerization, the benefits of this platform, everything we talked about here with regard to utilization and agility and portability. These were things that weren't obvious. People were very rooted in VMs. Two years ago, these things weren't very obvious.

So we worked on a lot of demos and use cases and tutorials to teach users about this and also to learn about user environments. To see how this technology could fit it. But I think last year, a year and a half, that has shifted to, actually, users know and they come to us and they tell us, "Hey! I got high utilization. I'm being very productive." I'm like, "Okay. That's great!"

What we're working on is, "Okay. How do we understand the enterprise requirements? How do we understand all of the security capabilities and build those security capabilities and going back and working with our engineering team to really prioritize and really understand the enterprise environment in which Kubernetes is used?"

What are the other things that Kubernetes is useful? What are the network constraints? What are the storage constraints? What are the traditional sort of hardware and legacy software that it needs to interface with? What are the things that we need to build to make it simpler for a new class of developers or maybe data scientists that are going to be using Kubernetes? How do we make it simpler for them to use?

So both at towards the lower end of the stack as well as at the top end of the stack, we've been innovating as a product team to get that understanding of the customer and build that roadmap for the product for Kubernetes and especially for GKE to start to move into mainstream

production, regulated, highly secure enterprise environments often times in on-premise use cases or in use cases that span enterprise and cloud and making it easier to use for a wider class of developers.

So those are the ways in which my job has changed. I come from an enterprise software background, so I've naturally gravitated towards making this software work for large enterprises. Providing a higher SLA. Increasing the security and reliability. Getting it into on-premise environments and making sure that hybrid works well. But then we've also been innovating on the developer efficiency and the developer ease of use side of GKE and Kubernetes in general.

[0:52:46.5] JM: Now, to some degree, Kubernetes is – You could just let Kubernetes take the wheel from you and the drive the pace at which you're working, and the project has so much momentum and so many different people working on it that it's probably taking on a life of its own to some degree, but I know that at Google, there is a high value placed on objectives and key results and key performance indicators s that you have this framework of measurement that can keep you sane when things are getting really crazy. Also, if things feel like they're losing momentum, then these numerical principles can help you find what to focus on. So, in light of that, what are the OKRs and the KPIs for Kubernetes this year?

[0:53:37.4] AS: I don't think that especially in a space that's moving as quickly as cloud and cloud native technologies, I don't think you can let – I don't know, the pace of the project drive. Where you're going in the future, that's a recipe for disaster. I think that as a product owner and a product lead, you have to have a vision for the future you have to be driving towards where you think the technology can go.

I think we're very fortunate and Google Cloud and that we have a 15 plus year history with containers and with container orchestration. So we've used it and we've seen a lot of it and I'm very fortunate to work with an engineering team that originally worked on board. So that's the kind of environment that's just you can't replicate and you can't dream it up. It just sort of happened and we're very fortunate to have that.

So that makes it a little bit easier to know, at least from a technical side, where things are going to go. But in terms of objectives and key results, I think it's very much about gaining traction in

new areas, and it has been that way since the beginning, even in the first years when I started working on Kubernetes. It was about, “Well, what is the segment and what is the use case that we think is the next use case that Kubernetes should grow and evolve to fill?” and it's still the case. We’re always looking at the next.

So, at this point, as I mentioned, enterprise and then developer experience, making it – As you become a mass adopted technology or a technology that's adopted by the masses, you have to have a way for people to learn it and use it that's intuitive where they're not having to gain a level of expertise. They can actually just come and use the platform and do the thing that they came there to do without having to learn a lot of the nuances. So that's where a lot of our focus has been now that we have kind of an underlying technology that's flexible that can be used for multiple types of applications. How do we make it easy for developers to use it? So building CICD capabilities, providing a service abstraction layer where there's a catalog of services that developers can choose from and easily integrate into their applications and then automating all of the rest of the underlying infrastructure and underlying capabilities of Kubernetes. So automating the provisioning, automating the scaling, automating the scaling down and then providing all of the auxiliary services, like monitoring, and logging, and auditing and compliance, providing all of that as a package so the developer is productive and can focus on what they need and they can easily get to what they need.

Those are the type of things that are on the OKRs, that are on the next phase. Then, of course, expanding the footprint into enterprise and into hybrid and making sure that not just one cluster, but that you can manage multiple clusters and you can manage them across environments and you can run different types of applications and you can run them in a secure way with multi-tenancy. So some of the things that we talked about earlier in this talk, those are new directions, and I'm very optimistic, because we have a team that's extremely passionate. We have a community that is extremely passionate. A number of partners that we work with that are that are world-class. So it's extremely important to me as a as a product lead to set the right vision and to have a good compass for this for this team to work towards.

[0:57:04.3] JM: Aparna Sinha, thank you for coming on Software Engineering Daily. It’s been really great talking to you.

[0:57:08.4] AS: Thank you.

[END INTERVIEW]

[0:57:11.5] JM: You listen to this podcast to raise your skills. You're getting exposure to new technologies and becoming a better engineer because of it. Your job should reward you for being a constant learner, and Hired helps you find your dream job. Hired makes finding a new job easy. On Hired, companies request interviews from software engineers with upfront offers of salary and equity so that you don't waste your time with a company that is not going to value your time. Hired makes finding a job efficient and they work with more than 6,000 companies, from startups to large public companies.

Go to hired.com/sedaily and get $600 free if you find a job through Hired. Normally, you get $300 for finding a job through Hired, but if you use our link, hired.com/sedaily, you get $600 plus you're supporting SE Daily. To get that $600 signing bonus upon finding a job, go to hired.com/sedaily.

Hired saves you time and it helps you find the job of your dreams. It's completely free, and also if you're not looking for a job but you know someone who is, you can refer them to Hired and get a $1,337 bonus. You can go to hired.com/sedaily and click "Refer a Friend".

Thanks to hired for sponsoring Software Engineering Daily.

[END]