EPISODE 1045

[INTRODUCTION]

[00:00:00] JM: When a developer spins up a virtual machine on AWS, that virtual machine could be purchased using one of several types of cost structures. These cost structures include on-demand instances, spot instances and reserved instances. On-demand instances are often the most expensive, because the developer gets reliable VM infrastructure without committing to any long term pricing.

Spot instances are cheap, spare compute capacity with lower reliability that's available across AWS infrastructure. Reserved instances allow a developer to purchase longer term VM contracts for a lower price. Reserved instances can provide significant savings, but reserved instances can be difficult to calculate how much infrastructure to purchase ahead of time.

Aran Khanna is the founder of Reserved.ai, a company that builds cost management tools for AWS, including tools for managing reserved instances. Aran joins the show to talk about the landscape of cost management and what he's building with Reserved.ai.

[SPONSOR MESSAGE]

[00:01:14] JM: Over the last few months, I've started hearing about Retool. Every business needs internal tools, but if we're being honest, I don't know of many engineers who really enjoy building internal tools. It can be hard to get engineering resources to build back-office applications and it's definitely hard to get engineers excited about maintaining those back-office applications. Companies like a Doordash, and Brex, and Amazon use Retool to build custom internal tools faster.

The idea is that internal tools mostly look the same. They're made out of tables, and dropdowns, and buttons, and text inputs. Retool gives you a drag-and-drop interface so engineers can build these internal UIs in hours, not days, and they can spend more time building features that customers will see. Retool connects to any database and API. For example, if you are pulling data from Postgres, you just write a SQL query. You drag a table on to the canvas.

If you want to try out Retool, you can go to retool.com/sedaily. That's R-E-T-O-O-L.com/sedaily, and you can even host Retool on-premise if you want to keep it ultra-secure. I've heard a lot of good things about Retool from engineers who I respect. So check it out at retool.com/sedaily.

[INTERVIEW]

[00:02:51] JM: Aran Khanna, welcome back to Software Engineering Daily.

[00:02:54] AK: Hey, thank you so much for having me again, Jeff.

[00:02:57] JM: You've been on the show a few times. Once you were talking about a side projected related to security and privacy, and one time when you were at AWS. When did you start working on a company?

[00:03:09] AK: Yeah. This company actually came out of a previous company I was at, but it's really colored from my experience at AWS working with customers and launching new products there, such as the SageMaker product as well as talking to a lot of these large enterprise customers who are constantly saying, "Hey, this is awesome. We love all the machine learning tooling that you're bringing to us, but can you apply some of that to my build, because it's 2 terabytes large now and absolutely inscrutable? I don't know what's going on. How do you even forecast on it? Let alone, optimize against it?"

Heard that a lot, and then definitely I was very well-aware of the problems and particularly being a data scientist in the role I left AWS to go into. Working at large, say, fortune 5 companies and seeing the pain from the customer side that's really on managing cloud cost. Essentially, I got to the point where no one was doing anything about it and we felt like we had to take some bias for action and build a solution here.

[00:04:07] JM: Well, you say nobody is doing anything about it. There are a ton of cost optimization companies. Did you have an idea for doing something particularly different?

[00:04:18] AK: Yeah, definitely. The thing that we would constantly hear from customers, and my response was always, “Hey, go look an AWS partner. Go look at one of the many, many partners out there who focus on this.” What I hear time and time again when those customers came back to us is, “Hey, they’re great. They got us 20% of the way there, but I still have to have a team full-time spending four hours a week doing the billing administration and the accounting.” Even then, the recommendations and suggestions that I get out of these third-party tools need to be implemented and go through sort of a verification process on our side.

There’s still a lot of overhead, and the problem is that while they might save money, they spend more time. Our goal was really to come at it from an automation-first approach and really get to the point where we could take that ops or finance person who’s spending four hours a week doing a lot of the billing administration, the RI management, the purchasing and take that down four hours a quarter with ample use of automation and sort of machine learning and optimization on the backend to make sure that that could go smoothly and the risk for the customer was greatly reduced.

[00:05:27] JM: Okay. RI, that stands for reserved instance?

[00:05:32] AK: Correct. Folks who are probably more familiar with large appointments running on AWS, they would probably run into this. But there’s always a lot of confusion in the industry around what this term actually is. Is it a machine that you’re reserving in someone else’s data center or what is this sort of object? I think a lot of people are unfamiliar with the fact that the cloud gives you all of these flexibility and you’re renting essentially these virtual machines that are very fungible.

When people think of an RI, they think of it as a specific instance in their cloud infrastructure. That’s not actually it. In fact, what an RI is, is it’s actually a contract. It’s a contract that says, “Hey, as long as you’re holding this contract in your account, you can use anything that matches it,” and that will essentially be covered, have zero cost to you in lieu of that RI that prepaid contract covering that instance.

It's sort of an interesting and counterintuitive concept and it also creates a lot of pain especially around attribution of who owns this object? Because it can float between almost any instance running across your cloud infrastructure.

[00:06:44] JM: Are there particular kinds of applications that people use reserved instances for?

[00:06:49] AK: Yes. Usually, the general advice is if something is a long-running application that needs to be up 24/7, that's a really good candidate for a reserved instance. However, as obviously cloud has evolved, more and more contracts and billing constructs have been introduced. Now there're convertible reserved instances, and savings plan. The idea is they want to give you more and more contracts and more and more flexibility, more options to essentially purchase the underlying compute.

The net of that is that now you really think about your consumption a little bit differently. You don't think about it in terms of a machine being up for a long time. You think about it in terms of an aggregate amount of usage even if you're turning on and off machines, spiking up and down. There's maybe some base load level that you want to cover even if it's not a consistent thing across any individual applications. It might be consistent among all your applications.

There's additional complexity now with more contracts being introduced that make both the attribution side and the planning side and optimization side a little bit more tricky as there's more opportunities to get savings from different types of workloads.

[00:08:05] JM: Just to make sure I understand a reserved instance correctly. If I just stand up an EC2 instance on my own, do I know if that is a reserved instance or a spot instance or what kind of instance it is?

[00:08:21] AK: Yeah. Usually, when you just click the button and get a machine, that starts with the on-demand instance purchasing model, which is the default type. It's the classic click a button, get a machine, and it's with you as long as you want it, until you turn it off. That actually tends to be the most expensive way to purchase these sorts of compute resources. What you'd actually need to do is you need to go in and specify when you do that initial provisioning, "Hey, I want this to be a spot instance."

However, with reserved instances, it's a little bit different, because what you would do is you would consume the machines the same way that you would with on-demand instances, but separately, you would have to go and purchase these contracts now that then apply to these running instances. It's a slightly different model for both spot and for these reservations, reserved instances and savings plans.

[00:09:12] JM: Where is the complexity in managing reserved instances? If I say I've got – Let's say my company is doing great. I've got some really long-lived applications and I want to give these applications reserved instances to run. If I just want to buy these reserved instances to stand up for a long period of time, what's the complexity in managing them?

[00:09:36] AK: Yeah. There's sort of two steps in process that get quite complex. Once on the purchasing side, and then once you've actually purchased these instances on the management side. Let's start with the purchasing.

With the actual initial basket of contracts that you want to put together, typically you'd have to go to every single separate service in AWS, and I think in the native GUI right now, they don't give you a lot of flexibility. You either pay 50% upfront, 100% upfront for all one or three years. However, what you're able to do is actually mix and match these things if you're able to be smart about this, and most folks will basically go pull surprising sheets into Excel spreadsheets and try and figure out, "Hey, for any given resource, there's 40 different contracts I can use to cover this, and I have hundreds or maybe thousands of resources in my account."

As that sort of scales up, really figuring out the best basket of contracts is to get you the best discount under your set of constraints as a business becomes increasingly more complex. In fact, if you want to solve it optimally, say I have $500 to spend upfront and I want to cover all of these infrastructure. Solving that problem is actually going to be hard.

There's a lot of complexity just in that purchasing step if you want to get the best discount for yourself, and the gap between sort of a naïve purchase and a really optimized purpose where you're blending the one-year and three-year contracts blend with the different purchasing modes, etc., is upwards of 50% incremental savings sometimes. It can be quite significant, and

especially for large organizations, this becomes a problem that is quickly outside of the scale that one data analyst can easily manage.

[00:11:20] JM: As you mentioned, there are a few different kinds of reserved instances. There are convertible reserved instances and there are standard reserved instances. Define these two types of reserved instances.

[00:11:35] AK: Just on the reserved instance, and there's also savings plans, which are another reservation model that's sort of similar as well that we can talk about. On the reserved instance side, the standard reserved instances are essentially the least flexible offering of a contract here that AWS will give customers. What they essentially stipulate is that you must use that same exact type of machine in that same region with the same operating system for the length in the contract. They're not able to flexible between different sorts of regions, machines, operating systems or tendencies. You're really locked-in with a standard reserved instance.

Now what's interesting about the standard EC2 reserved instances specifically is that they have a secondary market place where you can actually sell the axiom of that capacity to other users. What we do for our customers is we guarantee for some subset of their contracts, they can trade it back to us. We actually de-risk a lot of that purchasing when you're locking in with a standard reserved instance by giving you some insurance that we could take it off the books if you're changing around your infrastructure.

Now the convertible reserved instances have a slightly lower savings rate, but they are a construct that lets you, essentially with the call of an API, change the type of machine that they're applied to, but there're some very complex rules around those conversions. I think it's sort of a reason why the savings plans, particularly the compute savings plans were introduced just in the last couple months.

[00:13:08] JM: For the convertible reserved instances, when would I want to exchange a convertible reserved instance? I bought a convertible reserved instance to run my application. When would I exchange that?

[00:13:21] AK: Yeah. The typical logic is that you buy a convertible reserved instance, say, for a T2 machine, and then you turn that T2 machine that it was previously covering off, and now that contract, that convertible reserved instance is going completely unused. Every hour of the day, you're paying money for that reserved instance, but there's no machine that it's providing a discount for. What you'd want to do in that case is then find another machine in your infrastructure that's not covered and actually call the AWS API to make that conversion explicitly so that this contract now can cover the new machine that is running.

That's actually a very complicated process because of the conversion rules that AWS puts around these contracts. You can only convert to something that's of equal or greater value for the same time period or potentially a little bit more. It creates this complexity just around doing these conversions not mentioning the fact that basically every hour of the day that you're not converting, you are wasting some amount of money. There's a little bit of management overhead just making that conversion just in-time.

[00:14:31] JM: For the standard reserved instances, you can modify a standard reserved instance. When would you want to modify a standard reserved instance?

[00:14:42] AK: Yeah. I think this is actually something that applies only to a subset of standard reserved instances. For example, the database standard served instances for RDS or Red Shift or Elastic Hash don't, I believe, allow this. However, for EC2, I think there are certain flexible standard reserved instances, the Linux operating system family of instances comes to mind, where you can actually take an instance and actually apply it to different sized machines, or reserved instance, I should say, and apply it to different sized machines. A T2 large could equal some number of T2 mediums based on the conversion rate give to you in the rules set. Again, this is another bit of complexity that you then have to manage as you go through and select the basket of things you want to purchase.

[00:15:28] JM: Is your company completely focused around reserved instance optimization?

[00:15:36] AK: We're actually more of a lifecycle tool, because that step of just purchasing the reservations and managing them is really the last step in what we do as a broader journey, which starts with actually analyzing your infrastructure. Splitting it up into different, what we call

segments, and actually doing the analysis of, “Hey, what of my teams, my applications, etc., is covered? Which bits of this have opportunity for waste production? For rightsizing?” Really, once you go through that process, figure out what different bits of your infrastructure look like from a consumption basis and from a waste basis. Then you can really look at, “Okay, what now makes sense for me in terms of a reserved instance optimization strategy?”

There’s definitely work beforehand that we help with. The fact is that we have definitely tooling within our platform to make that some degree self-service, but because we started as this sort of data science shop first as well as serving enterprise customers, we really invested heavily in building a robust data platform that allowed us to create a lot of customized analysis for that first bit, because everyone infrastructure and consumption patterns do look a little bit different. Nailing that really then helps you nail the saving strategy that you implement down the line.

[SPONSOR MESSAGE]

[00:17:09] JM: If you are selling enterprise software, you want to be able to deliver that software to every kind of customer. Some enterprises are hosted on-prem. Some enterprises are on AWS. There might be a different cloud provider they use entirely, and you want to be able to deliver to all of these kinds of enterprises.

Gravity is a product for delivering software to any of these kinds of potential environments or data centers that your customers might want to run applications in. You can think of Gravity as something that you use to copy+paste entire production environments across clouds and data centers. It puts a bubble of consistency around your applications so that you can write it once and deploy it anywhere. Gravity is open source so you can look into the code and understand how it works.

Gravity is trusted by leading companies, including MuleSoft, Splunk and Anaconda. You can go to gravitational.com/sedaily to try Gravity Enterprise free for 60 days. That’s gravitational.com/sedaily to find out how applications can run the way that your customers expect in their preferred data center. That’s gravitational.com/sedaily.

Thanks to the team behind Gravity, the company Gravitational, for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:18:36] JM: The market, I just want to understand why are there so many cost optimization companies? Is there enough space in the market for all these different cost optimization companies?

[00:18:50] AK: I think it's part of the joint model that cloud vendors have had with their customers. People will say cost optimization is a part of what we do, but there's a broad range of services I think that are adjacent that are often confused with cost optimization, things like governance and putting in place policies around usage. Even security tools often will claim to have some amount of cost optimization.

I think in terms of the market, messaging is difficult because everyone wants to say they do everything, and there are definitely tools that are just focused on cost. I think there's a lot of space just within cost and it's moving so quickly and it's definitely an unsolved problem. It's something where people can adapt all the solutions and not really see a perfect coverage or perfect utilization pattern just because the environments are dynamic and the vendors are very dynamic. The ways in which you can actually get discounts are constantly changing. It means that the market is one that's sort of constantly turning over and innovating.

[00:19:55] JM: Do enterprises buy multiple cost optimization products or they just go with one particular vendor?

[00:20:04] AK: It's a still new market, but I think the bend I've seen is more towards the latter, where folks will adapt different platforms or different specific use cases. For really that high-level single pane of glass between finance and IT, we found a really strong foothold in the enterprise. For more strategic things that are actually touching infrastructure and doing cost optimization at the level of the actual machines that you're running, there are other vendors that have made a great impact there. I think it's really sort of a function of where in stack you operate and what you're providing, because especially the skills in these large enterprises with hundreds or

maybe thousands of developers on cloud, there is enough space and enough people feeling the pain at different points in the stack where a number of these solutions together could be relevant.

[00:20:58] JM: As a company that is building cost optimization software, how good are the unit economics? When you look at the business that you're trying to build, how good of a business is it look?

[00:21:13] AK: From our perspective, we're really trying to charge for the value that we provide. The unit economics are really purely based off of the time and money savings that we're able to provide to our customers, and just based on early results with the customers that we have been working with for the last year or so, we can pretty confidently say that even at large, large-scale, we make enough of a dent in terms of the value we provide with additional savings, de-risking a lot of the reservation purchase lifecycle as well as time savings, that we're pretty confident that we can provide a reasonably high-margin service that provides a ton of valued customers.

[00:21:58] JM: When you sit down with somebody that runs an enterprise and you start to tell them like here's how you can save a lot of money. What are the typical problems that they're encountering? What are you diagnosing when you're talking through their infrastructure and their waste?

[00:22:21] AK: Yeah. Obviously it changes customer to customer, but in general, there're a few buckets, and the first bucket is obviously waste reduction. We look at a ton of folks that just have stuff that's left on, unattached EVS volumes, idle EC2 instances, idle databases, etc. It tends to be low-hanging fruit, but definitely a place where for most organizations we see some amount of movement there.

Beyond that, obviously we look at the reservation strategy and any improvements that we can make to what they're doing on that side to either increase coverage, make sure that things like reservation renewals don't get missed, because the contracts expire open and then your cost spike up the next day. Additionally, sort of exchanging, automatically exchanging or even selling back unused reserve capacity. That's really important. Lots of passage just within that reservation strategy and reservation management.

Finally looking at provisioning options, like spots or even looking at rightsizing for machines. Are you even selecting the right sort of underlying instance type for your workload? There's a number of different data streams that we plug into to deliver this analysis, but it is customized per customer and the impact of each one of those optimizations vary just based on where the customers at and their cloud lifecycle and optimization journey.

[00:23:49] JM: Tell me more about that. If I integrate with the software that you've built, what are you plugging into? I've got a big complex AWS deployment. How are you observing and taking in the data that is necessary to understand where I could save money?

[00:24:09] AK: Yeah. What's beautiful about AWS and the other vendors in the market who are catching up very quickly is the fact they offer such robust underlying API. In fact, essentially, everything you do with AWS is through that API. What we use is a read-only identity and access management role to essentially give our production account external, but very limited access into your AWS APIs. We'll be calling essentially inventory and billing APIs looking at the metadata with our data platform and using that to make a suggestion.

Really, with the infrastructure that the identity and access management system has put in place, it's basically fully a trustless thing where you just have to put the credential into your account, trust that Amazon gives us sort of limited access and we just hit those APIs on your behalf. It's pretty smooth and it actually doesn't impact or touch any of your underlying production infrastructure. You don't have to install an agent or do anything that would compromise in terms of security your underlying deployment. It's all just metadata access and official AWS APIs.

[00:25:20] JM: And are those APIs that you're accessing for actually buying and selling instances or are you only giving the users analysis into what is going on in their infrastructure?

[00:25:36] AK: We actually offer a free read-only installation that is just doing the read access for the analysis. If it makes sense for the customer, then we actually upgrade them to the right version, which is a slightly modified credential that does actually have the access to automate the reservation management, purchase reservations and resell them as well.

[00:26:03] JM: What actually happens if – So if you start to buy and sell the instances for the users, what actually happens there? What kinds of improvements are you making about how people buy the reserved instances?

[00:26:18] AK: Yeah. One simple one is just, because it's software, our timing is better. We don't leave any gaps in terms of coverage, in terms of renewal, time, etc. Just by having a little bit of automation there, there's a lot of dead time where stuff is uncovered. Just because it's somewhere no someone's stack and no one's been looking at it, that we're able to just provide immediate savings improvements on.

Beyond that, if we actually look at the actual planning and the execution bit, we're able to put in place automation policies around reservation purchases and reservation exchanges. You can actually set a high-level set of constraints and we do the optimization and run the algorithm 24/7 in the background. So you don't actually have to go in and worry about doing the alerting and say when something is underutilized, reselling it, or when something is underutilized, exchanging it. We can actually have automation on top of that. Just making sure that happens in the background and you just get an update email about it. Really, on those two axes, were able to make a pretty big dent for the customer in terms of incremental savings even if they had someone who's doing this full time four hours a day beforehand.

[00:27:32] JM: So you're buying back AWS reserved instances like when the user has been using their reserved instance and then they don't need it as much as they've actually provisioned for? Is that how it works? Can you just tell me more about like when are you actually buying their reserved instances back?

[00:27:55] AK: Yeah. What we call it is an RI lease, and what that means is essentially when a customer purchases one of these special reservations through our platform according to the terms of the AWS marketplace, they have to hold it for 30 days. But once that term is up, we'll actually give them a guarantee that they can click a button and send it back to us and essentially take that commitment off their books at any time. It's very simple transparent process that just happen from the dashboard, and really automation around that is the ability now to then basically have an automatic email sent when something is underutilized and automatically send things back to us. You're not wasting any time with the commitment that's sitting in your books.

On the other side, when you have a short-term spike, we're actually able to take some of this capacity and give it to you for a short term to cover that, which is useful if you don't want to go into a large commitment, but you do want to kind bring down that operating cost, say, for the month of that deployment.

[00:28:55] JM: Got it. Do you – If you're buying back AWS reserved instances that people buy, then are you ending up with this pool of reserved instances that you own that just customers of Reserved AI have provisioned and then they've sold to you?

[00:29:18] AK: That's correct to a degree. Yes.

[00:29:20] JM: Then what you do with those instances that you have lying around? That pool that you have lying around?

[00:29:27] AK: Yeah. We have a number of things. Primarily, we, like I was saying before, will help release those to other customers and really try and drive savings between all of the folks in our network, because I think the usage patterns are quite unrelated. When someone on one side of our customer base spikes, it often is not correlated with other folks who were able to balance out this pool between our customers in the backend, essentially.

[00:29:57] JM: Interesting. Basically, what you've done is you've built a way to incentivize your user base to buy longer AWS reserved instance leases and then you get to capitalize on the pooled gains across the entire customer base.

[00:30:20] AK: Well, I think more from the customer's respective, the idea is that it actually came out with a lot of the work we're doing at these fortune 5 companies working on data science. One month, we'd be using tons of CPU-heavy instances trying to process down to terabytes and terabytes of time series data, and the next month we'd be running GPU instances. Actually, trying to train models on these things. Obviously, in isolation, any of those usage patterns might not have made sense for reservation. But if I were to join it with, say, other labs that were sort of doing the opposite of us, but we're bursting on different sorts of instances quite frequently, it might make sense to trade. You start to see at scale this does make sense

especially among certain subsets of customer's environment such as their development environments.

[00:31:10] JM: It's pretty cool. The marketplace where this computer is actually being bought and sold, this is an AWS marketplace? So they like give you APIs for buying and selling?

[00:31:23] AK: Correct. This is an official API, and you can Google it. It's the EC2 reserved instance marketplace. Yeah, it's been an active API for number of years. The problem is that it's always been hard to find a counterparty in there, and there's not been a lot of volume just because the number of folks who've really been using it is quite low. But as a function of having the visibility and the trust that we do have with our customers, we're very aligned with all of our customers. Our goal is 100% to save them time and money. We're able to actually create a little bit of a dark pool, just leverages API on the backend to provide the value of actually moving sort of these commitments between our customers and off their books.

[00:32:09] JM: It's kind of a niche two-sided marketplace, but it's one of these two-sided marketplaces where it it's kind of niche in a good way in the sense that anybody that would potentially have bursty capacity or situations where they're going to provision a heavy amount of infrastructure across reserved instances that they may not need in the near future, they could potentially want to sell those instances back to you. But then of course you have to set the price at which they would be selling it back to you. So how do you do –

[00:32:54] AK: It's actually – Yeah. It's actually not a price per se. Obviously, we'll leverage some impartial upfront and fully upfront paid instances, which actually do have some prepaid value and do have a price, and we obviously use the market price for that to be fair to our customers. For instances that are just a commitment to pay these no upfront standard instances, we're actually just trading the liability. There is not really a price. It's just the commitment to pay, if that –

[00:33:27] JM: If I am using Reserved AI, I make a commitment to pay and then I can –

[00:33:35] AK: Yes. That's exactly what a no upfront RI actually means. It means that I'm committing to pay some amount every month for this contract, but you're not actually prepaying for it.

[00:33:46] JM: Got it. What happens when they sell it back to you?

[00:33:50] AK: Yeah. That contract, which is a commitment to pay, is something that essentially gets transferred out of your account so that next month you will not be billed for that commitment because it's no longer sitting in your account.

[00:34:05] JM: What's your strategy for building up the marketplace? Like the volume of reserved instances that you would have under your control?

[00:34:14] AK: It's really a function of our customers and where we can drive savings for them. We're squarely looking at the places that we can create these offers and create these opportunities that provide the most mutual benefit to our customer base and being obviously very data-driven. I come from data science background. My cofounder is actually from D.E. Shaw where he a futures trader, which is strangely relevant to this world. There's definitely a lot of analysis put into it, but the net of it is how can we maximize our customers' savings? How can we provide them the most value?

[00:34:47] JM: What are the biggest sources of AWS waste?

[00:34:50] AK: Yeah. I think the biggest one that we see is just idle databases and instances. It's sort of this time-honored thing that every time people talk about cost optimization, they talk about that as the obvious thing and everyone nods their heads and they're like, "Yes. Of course, I'll never do that." But obviously that's not the case. You see places everywhere where a machine should be at least switched off, if not completely removed. It's ironically one of the big things that we see in terms of just simple cost reduction.

I think beyond that, in terms of just raw waste, there's a lot of enterprises, especially ones that used to be working with CRI's that let these contract go underutilized, and it just creates a very unfortunate situation where often they'll even purchase more reserve capacity even though they

have a lot of this reserved capacity sitting around doing nothing for them. It's another big source of waste that we try and help customers avoid.

[00:35:48] JM: Tell me about like an example customer. I just like to walk through a case study to better understand what is actually happening.

[00:35:59] AK: Yeah, definitely. Well, we have a number case studies linked on our website, which is nice, but I think we can maybe talk through what we worked with Valtix on, which the cloud security startup based out of Santa Clara. Essentially, they had a number of developers who had full autonomy the spin up machines in the cloud. All their core infrastructure is running in the cloud. In terms of just tracking down what was being used by who as well as then putting in place a reservation strategy, they just didn't have the bandwidth to deal with it. They had too many other customer requests that were high-priority.

We came in, and because we are truly customer-obsessed, we actually try and really handle the customer through the whole journey and kind of take ownership of the cost optimization part of their job. Our representative would come in, basically do that initial analysis with the customer sitting on the call. What they had was actually a number of pre-existing convertible reserved instances that were being underutilized. That was sort of the first thing that we saw there. What we did was we basically work with the customer to put in place an automation policy to make sure that those were constantly utilized at the highest level.

Then beyond that, we saw that a number of the things they were previously covering with convertible reserved instances were instances that were actually up for a really long time and were very unique workloads that probably weren't going to change how they are being hosted at least for the next year. We actually moved those two more of a standard reservation model just by sort of looking through our analysis tools and understanding from the history what's likely to be up.

All of that are represented, basically put in place on behalf of the customer. The experience to them after the initial meeting was just us sending emails asking for their approval to basically implement these cost-saving measures on their behalf, and that actually just constantly kept going as their infrastructure grew. We would put together new purchases as we saw savings

opportunities and send it over, and they would approve it. Really trying to take the heavy lifting off of their shoulders while delivering those same results of basically having your own in-house cost optimization tooling.

[SPONSOR MESSAGE]

[00:38:22] JM: When I'm building a new product, G2i is the company that I call on to help me find a developer who can build the first version of my product. G2i is a hiring platform run by engineers that matches you with React, React Native, GraphQL and mobile engineers who you can trust. Whether you are a new company building your first product, like me, or an established company that wants additional engineering help, G2i has the talent that you need to accomplish your goals.

Go to softwareengineeringdaily.com/g2i to learn more about what G2i has to offer. We've also done several shows with the people who run G2i, Gabe Greenberg, and the rest of his team. These are engineers who know about the React ecosystem, about the mobile ecosystem, about GraphQL, React Native. They know their stuff and they run a great organization.

In my personal experience, G2i has linked me up with experienced engineers that can fit my budget, and the G2i staff are friendly and easy to work with. They know how product development works. They can help you find the perfect engineer for your stack, and you can go to softwareengineeringdaily.com/g2i to learn more about G2i.

Thank you to G2i for being a great supporter of Software Engineering Daily both as listeners and also as people who have contributed code that have helped me out in my projects. So if you want to get some additional help for your engineering projects, go to softwareengineeringdaily.com/g2i.

[INTERVIEW CONTINUED]

[00:40:11] JM: Now, is there a feeling of being a consultancy in that situation where you have to manually send them in suggestions, or how much of it is a programmatic process where you're

just monitoring their infrastructure and you can actually programmatically give them recommendations for what to change?

[00:40:32] AK: Well, so I will say that most of it actually programmatic. What we do want is a human face on the other end. While the tool is largely self-service, most of the alerting is programmatic. We do want to have an initial conversation. We do want to sort of be on the call with you when you're going through the product and making that first purchase, because it could be a big commitment and it could be big and scary. While, yes, most of it self-service, most of it automated, we definitely want to have a human on our side in the loop as well to make sure the customer is having a great experience and they have someone to lean on in case you have questions or concerns. I would say that it's definitely less of the consultancy having done that sort of before. It's a little lighter weight than that, but we do want to make sure that we provide the support that customers need and want.

[00:41:22] JM: Does your software only work for raw EC2 instances, or can you help with the managed services? If I have a managed service that's running an EC2 instance under the hood, can you help with cost optimization for that?

[00:41:39] AK: Yeah. Just the way that the reservation system works is if you are consuming EC3 under the hood, you’re being billed for it and you can basically cover that infrastructure cost with the reservation, be it in a managed service ECS or in SageMaker even. The other thing is that recently with savings plans, they are now able to cover Lambda and Fargate. You can use a reservation model in those sorts of serverless services now as well.

I think the scope is totally expanding of where you can use this pricing model as well as the fact that the other database services and a number of storage services, particularly on Azure as well, allow you to consume with this model.

[00:42:26] JM: If you have these people who are just using the version of Reserved AI where they just have a dashboard and it's reading their information. You're seeing where their waste is. What is like the most common next step? Just to make sure I understand how the system works correctly. What happens? Like what should they do if they have some kinds of waste? What will be the recommendation that Reserved AI will give?

[00:43:01] AK: It's very dependent on the type of waste, right? There's a number of different ways that you can be idling resources that you're being charged for that you definitely should not be. So things like elastic IPs, load balancers, even EBS volumes that can be sitting idle or unattached. Those often tend to be big source of waste. Beyond bad, instances that are just being underutilized for hours at a time that should be turned off. There's lot of provisioning and turn off thing sort of suggestions that come out of the initial analysis based on what's really being wasted. Then the next step is always sort of to put together purchase that makes the most sense and drives the highest savings rate or purchase of reservations.

[00:43:46] JM: Does any of that data hard to acquire? The data around what is being wasted?

[00:43:54] AK: I think the difficulty comes in the fact that it's a number of different APIs and a number of different services within the cloud environment that you have then go and gather this data from. It's definitely accessible. It is difficult and that you will need to do a lot of legwork. Even the services like Trusted Advisor that are supposed to bring this all together into one pane of glass really only expose a subset of the data. It ends up being sort of a hunt through a number of different APIs to get just the baseline data in place to then start making these higher-level intelligent recommendations.

[00:44:33] JM: What's been the hardest engineering problem so far in building Reserved AI?

[00:44:39] AK: I think, for us, probably the most difficult thing was solving the purchasing optimization, because like I was mentioning earlier, especially for really large customers, it's technically an NP-hard problem. For any single machine, there's now 40+, maybe 100+ in savings plans now different ways to potentially cover them. Trying to figure out under certain constraints of time length commitment and upfront capital commitment, how to take all of these different potential shuttles of contracts that could apply to this infrastructure and pick out the best one is this incredibly difficult dynamic programming problem.

Just from an engineering standpoint of actually scaling the stuff out from large customers, trying to solve this sort of thing in parallel, it was very complicated, but incredibly rewarding and interesting especially coming from a data science background where a lot of these sort of work

could then be made more interesting by saying, “Hey, these are all based off time series forecasts now.” Everything you're doing is stochastic. I think as hard computer science and math problem, it's definitely been one of the more tricky ones that we’ve had to deal with.

[00:45:53] JM: Can you talk more about how you actually solved that?

[00:45:56] AK: Yeah. We did a couple different things sort of at a high-level. We did optimization on the algorithm side. We used, like I was saying before, this specific dynamic programming algorithm, and we also managed to paralyze a lot of the analysis and pre-work, and we used Lambda functions actually running on Kubernetes to scale that out. We actually had to put in place some new infrastructure to allow us to start doing that at scale on our Kubernetes cluster. What was sort of on the instructor side and on the algorithm side how we, in a nutshell, dealt with the problem.

[00:46:31] JM: The dynamic programming problem you're talking about is essentially I've got a reserved instance and its perhaps too big for what I am doing, and you can replace it with several smaller reserved instances?

[00:46:49] AK: It’s a little bit different. The way that we frame it is that you have, say, these four machines you’re running and you have different types of contracts where you know one contractor where one contract to cover the first two machines. One contract for the different savings rate but could only cover one machine, and different contracts might have different discounts based on how much upfront capital you put into them or what the term length commitment is.

Based on constraints at a high-level a finance team would give, say, I'm willing to spend $100,000 upfront and I'm willing to commit $50,000 for one year and $30,000 out for three years. Basically taking that as high-level constraints and finding the optimal mix of contracts under the hood that gets you the highest savings rate is highly nontrivial. That’s sort of the problem there.

[00:47:41] JM: Can you explain these discount rates more? This is the first I've heard about these, or savings rates I think you call them?

[00:47:50] AK: Yeah, discount or savings rate, either way. What that reservation is doing is essentially offering you a discount on that underlying resource, right? What the savings plan or RI is doing is giving you a lower price for that resource that you're consuming. Based on either you can pay all upfront, no upfront, or partial upfront, you can also pay for one year or commit over three years.

Based on whichever one of those you choose, it's going to have a different price. It's not a standard price across all services, across all instances, across all regions. In fact, it's a different price or a different discount rate I should say for all of those options across all of these different sort of resources. Because of that, it's really complicated to actually figure out how you get the highest overall discount rate from combining contracts with all of these individually distinct discount rates.

[00:48:51] JM: This financialization of compute resources, you said this is a pretty nascent market. It's only a few years old on AWS and there really just hasn't been much liquidity. Why is that?

[00:49:08] AK: I think to some degree, it's part of the fact that this has been pretty nascent and also fairly niche in terms of sort of who's participating. The large enterprises tend to be the folks who really at scale are consuming reserved instances. I think that even due to the fact that it's not a lot of folks even within those organizations know that the marketplace exists, it just means that there hasn't been really enough interest to run a volume of folks who would use this tool to make it sort of the robust thing that I'm sure they're envisioning when they put it out there. Really, what we're trying to do is to, at least on a small part of that vision, deliver and deliver on behalf of our customers.

[00:49:56] JM: What's been the process for finding early customers?

[00:50:02] AK: Yeah. Last year, we really started with – As you can see probably from our case studies and our customer list, sort of early mid-stage startups, particularly software as a service startups, where because – Especially in the last year, there's been a big emphasis on profitability going and finding folks where their gross margin was a big function of their cloud

cost was really helpful to us in finding early champions between both finance and IT when where those teams are really aligned and this problem was top of mind.

Really, caught our teeth and build the best product we could for that set of the market. Then as we sort of took our heads up, we found, obviously, really these large enterprises were ones where the time spent and the money spent was so enormous that any way that we can move the needle would just be a much, much larger impact.

Really, in the last six months, we've started building more for those enterprise clients as well. Actually, one of the recent folks who joined our team, our founding preview of engineering, came from some of those large enterprises. He was actually previously at Splunk. We're really trying to build a product that obviously still continues to serve the needs of these startups and smaller companies, but can also scale to meeting some of the demands of these larger businesses that are now sort of coming in and asking us to provide more customized services for them.

[00:51:27] JM: At some point in the future, do you think that there will be dedicated cost optimization engineers? Do you think this is a complex enough problem that will have people that are employed at companies entirely dedicated to using cost optimization tools?

[00:51:43] AK: Well, what's was quite funny is there already are. We talked to a number of companies, in fact, it's part of the impetus of us starting this company. We talked to a number of companies that you would think of as some of the most forward-thinking cloud native organizations where there are vendors like AWS and bend over backwards to please them. They have entire data science teams and engineering teams just dedicated to internal cost optimization and cost optimization tools. Large public companies in Seattle that I've talked to definitely have teams of 5+ people working on this. The folks in the Bay Area that you can think of as large, late stage, cloud native startups that are all-in on AWS. They have full teams doing this.

If businesses of that scale, of that resourcing and that relationship with their cloud vendor still need that resourcing to just handle and manage this problem, I think there is definitely a space for third-party tools to at least take some of the undifferentiated heavy lifting off the plates of

these folks and really let them get back to what makes them and their business tick and not basically wrangling the same old AWS cost and billing APIs.

[00:52:59] JM: Is your cofounder your brother?

[00:53:03] AK: He is indeed.

[00:53:06] JM: What's it like building a business with a family member?

[00:53:09] AK: I think something they say when you're finding a cofounder is there has to be a lot of trust, and I don't think there is anyone who I've built more trust with than someone I have basically known their whole life and pretty much vice versa. It's interesting, because we've always been into similar things. He's been a programmer since before I have. He was building games in high school before I even knew sort of what terminal was. But I think because of the fact that we've been in similar fields, we have very similar interests and there's a lot of trust, it's been absolutely wonderful experience.

I think that I couldn't have asked for more in a cofounder especially because we have such almost unspoken rapport where the communication overhead doesn't even have to be there because we almost know what each other are thinking.

[00:53:57] JM: All right. Well, last question. What would you be working on if you are not working on Reserved AI?

[00:54:03] AK: Yeah, I think this would probably harken back to the last term I gave, and I would definitely working on privacy tooling. If not, something within the confines of a business. Potentially something in the nonprofit space or on privacy, because I think since my interest sparked in 2015, it's only become more and more pertinent. I think people are really starting to wake up more and more every day to the importance of privacy in our increasing digital lives. Definitely a conversation that I'm still very interested in, and if I was not doing this, I would love to be more involved in.

[00:54:41] JM: Actually, I have one more question. Given how – Reserved AI seems like a company that is very much influenced by your experience having come from AWS. I mean, the fact that I had never heard of this reserved instance marketplace and it's kind of a subtle nascent market that clearly has a lot of potential though. It just makes me reminded of the fact that I'm sure there are a bazillion nascent opportunities just within the AWS ecosystem. Did you have any other ideas just based on your depth of knowledge from seeing AWS on the inside?

[00:55:24] AK: I think from my perspective, the machine learning tooling where I really cut my teeth and particularly machine learning in IoT is a place where I think there's a massive, massive wave coming in terms of new applications, new functionality and a lot of new infrastructure that's going to be needed to support that either co-mingled with AWS or built directly on top of it. I definitely have a number of ideas and things that I would love to worked on or see worked on in that space particularly around model management at the edge and particularly robotics use cases.

There's actually a portfolio company that shares an investment that our lead investor is investing as well. I guess a co-portfolio company that is called Covariant AI, and I think the stuff they're doing is incredibly interesting. The infrastructure to support that sort of breakthrough at scale is going to be something that I think will really change the way that our world works.

[00:56:26] JM: Aran, thanks for coming on the show once again.

[00:56:28] AK: Thank you very much, Jeff, and have a great rest of your day. Stay safe out there.

[00:56:32] JM: Likewise.

[END OF INTERVIEW]

[00:56:42] JM: You probably do not enjoy searching for a job. Engineers don't like sacrificing their time to do phone screens, and we don't like doing whiteboard problems and working on tedious take home projects. Everyone knows the software hiring process is not perfect. But what's the alternative? Triplebyte is the alternative.

Triplebyte is a platform for finding a great software job faster. Triplebyte works with 400+ tech companies, including Dropbox, Adobe, Coursera and Cruise Automation. Triplebyte improves the hiring process by saving you time and fast-tracking you to final interviews. At triplebyte.com/sedaily, you can start your process by taking a quiz, and after the quiz you get interviewed by Triplebyte if you pass that quiz. If you pass that interview, you make it straight to multiple onsite interviews. If you take a job, you get an additional $1,000 signing bonus from Triplebyte because you use the link triplebyte.com/sedaily.

That $1,000 is nice, but you might be making much more since those multiple onsite interviews would put you in a great position to potentially get multiple offers, and then you could figure out what your salary actually should be. Triplebyte does not look at candidate's backgrounds, like resumes and where they've worked and where they went to school. Triplebyte only cares about whether someone can code. So I'm a huge fan of that aspect of their model. This means that they work with lots of people from nontraditional and unusual backgrounds.

To get started, just go to triplebyte.com/sedaily and take a quiz to get started. There's very little risk and you might find yourself in a great position getting multiple onsite interviews from just one quiz and a Triplebyte interview. Go to triplebyte.com/sedaily to try it out.

Thank you to Triplebyte.

[END]

