Jillian (00:00):

Hello everyone. Welcome back to the AWS podcast. I'm your host for today, Jillian Ford, and I know a lot of you have been wanting to hear some customer stories about how they're doing some really cool things in AWS and we've got a really good one. We've got MobileEye and Daniel from the account team at Mobile to really share their journey of going into Kubernetes and specifically EKS. So let's do some intros. Daniel, do you want to kick things off?

Daniel (00:27):

Yes. Hi Jillian. Hi, now everyone. My name is [00:00:30] Daniel Goldenstein-Bar. I'm an AWS solutions architect working with mobile in the last two and a half years as part of the AWS account team, I invest in Israel and I'm working with the enterprise segment in AWS

Jillian (00:44):

And Benzi our star from MobileEye.

Bentzi (00:48):

I'm Ben. My full name is actually Ben Tzion Lifshitz, but let's just keep with Bentzi. I think it would be easier for everyone. I'm working at mobilize rem, which is the component that [00:01:00] creates the map that is the base for the autonomous vehicle to drive on. So we are creating technological map like AV map for the autonomous vehicle.

Jillian (01:14):

So Cool. Oh my gosh. And then knowing that Kubernetes is a part of that, so I'm sure a lot of people are like, alright, let's go through the actual journey of being able to get to Kubernetes and specifically EKS. So can you walk us through that

Bentzi (01:28):

Previously? First of all, [00:01:30] the way we are creating the map is by running a lot. I wouldn't say other phrase, but I think it's not proper for a podcast, but we're running massive loads of Spark applications that are creating the map. Okay. So hundreds of thousands of CPUs running and creating the map and we need a system to orchestrate all of that. And we previously had another system that ran pretty good, but [00:02:00] it didn't have the abilities of Kubernetes, it didn't have the dynamic configuration style of Kubernetes. We couldn't add components to the systems very easily. We couldn't view everything. We couldn't add third parties components to the system and we were very stuck in what we have to upgrade. The system was a headache and we wanted to go to a better place where we can actually build a massive scale platform on top of [00:02:30] Kubernetes. And because we're a SAVs, we wanted to try EKS for that.

Jillian (02:37):

Wow. It's so cool. And I'm just curious because I know a lot of people that I talked to who want to make the journey to Kubernetes, they've got mixed experience, some do have experience with Kubernetes, some don't. In your team, what was the experience with Kubernetes like before going to EKS?

Bentzi (02:55):

So we haven't had any experience in Kubernetes. We had other teams [00:03:00] in mobile, not our team that had experience and actually before we started the journey, we asked them about it and how they managed to do with it. And we saw that they are also running a massive scale in Kubernetes. So we got the approval that is possible to run a massive scale on Kubernetes, but everybody told us since the beginning, it's a very complicated system. It's good if you want to run a very complicated system on top of it, but if you just want to run something very simple, [00:03:30] do not use Kubernetes. But for complicated systems, for massive scale systems, this is a very good decision to go on. The guest and Danielle can

Daniel (03:43):

Yes. Yeah. And let provide a high level overview. Mobile are doing basically is a very large spark operation building those AV or HD maps that Bensi has mentioned. The process is starting with ingestion of a lot of [00:04:00] harvesting of a lot of data that is coming from the vehicle itself, driving all over the road across the large continent of the world, mapping semantic information of the road itself. It can be a traffic light, it can be the exact lane and more and more information of the road itself. All this data is harvested and ingested in AWS and once a modular are collecting enough data, they are starting to build a high definition mapping [00:04:30] of the entire continent. It may be, and as Ben mentioned, it requires a vast amount of VCPU as you can. Also, we can refer to you for the joint case study that was recently published where we discussed about a 400 K-V-C-P-U in parallel running in AWS for several weeks to build those maps. So this is the product, and this is based on the Spark technology [00:05:00] OS spar managed completely by the mobile and RAM team to build those maps. Mobile have used a third party container orchestration to manage this workload and as Ben mentioned, they wanted to expand the tools and component that they use and were looking for an ecosystem that has an increased availability and adoption worldwide. And they chose Kubernetes and EKS [00:05:30] as its platform to be used to be and they have basically migrated this entire workload from the third party container orchestration to A-W-S-E-K-S.

Jillian (05:42):

Wow. It's absolutely fascinating and it's clear that there were so many different challenges that you really had to weigh as you were thinking about what's the right approach. Now let's get into really the integration Amazon EKS into the infrastructure. Can you tell us that story of what that experience was like?

Bentzi (06:00):

[00:06:00] For us, one of the major advantages were the fact that EKS, when we are using EKS, we don't have to create the Kubernetes clusters ourself. We're taking, everything is prepared for us without messing with Kubernetes at all, which is a very hard thing to do to manage ES clusters yourself. So this is a given first of all and then all of the base components you need to run something [00:06:30] like you need DNS, you need network, you need storage. All of those things are provided easily by EKS blueprints, which are ways to integrate various components into a system very easily. So our DevOps team, this is not my team, I'm a programmer, but our DevOps team prepared the environment for us and it was actually very easy comparing to the previous third party orchestrator we used because everything is ready. [00:07:00] You have Terraform ready for you and you just need to implement it in your environment. So the beginning was very, very easy and to get to a system that the basic, the hell world Spark application works was very simple, but it got very complicated after that. I think let's give it to later.

Jillian (07:21):

Yeah. Another part that I'm also curious about that I'm sure on people's minds is when you're adopting really a new technology, there's a lot of different [00:07:30] changes that have to take place. Part of it is being the cultural aspects of a lot of different teams now adopting a new tool, really a new way of maybe even as just a new culture of way of actually being able to develop applications. So were there any changes that within the whole software engineering team had to make? And I'd love to hear what your lessons learned from that change of having to actually as an organization use Kubernetes.

Bentzi (07:56):

We are developing a system that our other developers are using. So [00:08:00] what we try to do is to make everything very easy to the other developers and to replace everything under their feet without they actually know that this is happening. But for this to happen, we had to learn ourselves to use Kubernetes because as you said, it's a very different animal. It's a very different way approach to how to manage stuff. So most of the developers, their tools kept working. We have replaced everything in the code [00:08:30] so everything seemed the same for them even though it ran in a completely different way. But for us, for my team, we had to learn how to work with Kubernetes, which is very different. You are doing everything is the base components are different. You have pods instead of dockers, you have the instances are not something you just connect to and see dockers on it.

(08:54):

You are connecting to pods and you see everything in different granularity. There are namespace, [00:09:00] there are different clusters, different, the Kubernetes context, all those things we have to learn. It was, at the beginning it was very confusing because there are so many terms everybody telling you there is a service account and service and pod and namespace. But after a week or so, all the new terms are actually making sense and they are actually, I think for me it's much easier to use it because in Kubernetes I think one of the major advantages [00:09:30] that everything is very visible. You can see each component on its own and you can see exactly what it's doing and how it's doing it. The shift was a bit confusing but it was very good for me.

Daniel (09:42):

And then I will elaborate more together with the AWS account team, we were working with Ben's team and other team members as well. Also to work on education about A WSE Ks about knowledge sharing best practices, how to [00:10:00] create the A WSE Ks ecosystem in the environment of the customer. We have also, we're using other projects of AWS like the data on EKS and I invite you to listen to a previous Amazon podcast that was recorded recently, talking specifically on Amazon data on EKS. And the data on EKS project offers open source resources for building modern data platforms on Amazon E Ks. This is a project [00:10:30] that were heavily used by RAM team and Bente can also discuss it further how it tested them in their journey. And we didn't mention a very important use case that all the vast amount of instances and workloads running attri are like more than 90% of it is based on EC2 spots.

(10:49):

So this is another challenge that we will need to tackle and to consider in the journey. And working very closely with the AWS account team and the data on EKS [00:11:00] project, we succeeded to achieve this migration. As Bensi mentioned, we have tried to provide the mobile end customer using the Spark application, the exact environment or even better environment with EKS versus the other container orchestration that were before. So this was the main challenge to provide that the customer of the platform will don't feel the difference between those system and will achieve the same results even [00:11:30] better and with much greater performance and speed versus the other system.

Jillian (11:35):

So amazing. Betsy, can you give our listeners just a sense of how long did this entire process take from being able to scope it out to actually being able to start using the Amazon EKS in production?

Bentzi (11:52):

The initial took I think four months to get to an environment, which is actually very fast because [00:12:00] in four months we created a system that can run a hundred of thousands of vcps at the same time. It is hard, this was only the initial process because after that we had to start first of all a very massive testing on the platform, just four months to prepare anything. Then massive testing and we have many rounds of testing identifying something that we need to do, want to do differently and then actually change it and test it again. So we took [00:12:30] us also a while, this took us several months also, while I can explain on some challenges we had along the way, many of those were actually related to our own platform. And then the migration. The migration is very hard and long courses, it actually still occurs because it's good enough that you have a system that works. But to migrate all the users, it's a really big headache. And internally, even though we created a way to [00:13:00] replace everything underneath the users, without them actually knowing it, still some experience are different and people want to do the change in their own terms. So it's actually another few months to do that as well. So we started February last year and we are still migrating. So four months for developing the system. The initial then several months, I don't [00:13:30] remember exactly how much to testing it in scale in massive scale.

Jillian (13:36):

Wow. Absolutely amazing. So now what performance improvements have you started to really see from this architecture?

Bentzi (13:43):

I think the most impressive improvement is that you can first of all run several environments with different configuration at once. Let's say you want to run different versions of Python on different spark applications. This is something that was not [00:14:00] even possible in the previous orchestrated. We have it now. We do have it and we actually use it. Let's say you want to run different spark versions. Let's say you want to run different configuration. This kind of job makes more memory. This kind of job needs more CPU, all of those things we had in the previous orchestration system, we were very static. We had to use the same configuration for all of the Spark applications and now we're running dynamically. [00:14:30] We have different things that's running all the same. So this is the most important improvement. Also, the implementation of third parties became very easy. For example, we tested I think three different monitoring and logging solutions and to test them was actually just deploying some help chart and starting to run.

(14:51):

While in the previous solution we had to do very complicated deployments and to make something get to all of the nodes everywhere [00:15:00] could take us a month. Also, another thing is things like environment variable. Some environment variable can control the way you are working. For example, just something that happened to us this week, some timeouts to S3 are variables in the new platform. We're doing it instantly in the moment we're changing it. Everything from this point on changes and in the old platform it took us maybe a week or two weeks to deploy it. So the [00:15:30] improvement we saw are everywhere. Performance wise it's very complicated because some parameters are very

different and we're still not able to say exact numbers on our improvement. So I guess we have also a technical blog that we'll write about it and we'll probably have a better numbers still. Then.

Jillian (15:49):

Daniel, I'd love to hear you were helping Bensi and the rest of their team on the architecture. How are you looking at Amazon EKS and specifically cost effectiveness [00:16:00] in this architecture?

Daniel (16:02):

So in regards to cost effectiveness is the ability of using the EKS construct like AWS carpenter to better utilize the infrastructure of AWS and for the specific use case of running a very large spark operation. What we were using is that on the driver instances, on the workload, we used eyes in order to maintain those instances and [00:16:30] for better price performance. But for all the worker nodes, we were using a e C2 spot and we were utilizing the e C2 spot, working together with the A WS accounting to identify the better location and to better utilization of spot instances. We're working towards the best practices. For example, how to use EC twos diversification with different types of instances, utilize the availability zones of AWS [00:17:00] and even utilizing the A WSS PS API with the S PS API, you can monitor and query the A WS regions for the availability of spot instances and the probability that you will have to get the SPOT request to fulfill the spot request for your desire. So we have utilized all the best practices that we have to offer to our customer to better utilize the [00:17:30] EC2 spot in order to reduce cost and to make the workload as best and cost efficient as we can provide to our customers.

Jillian (17:42):

Always love spot instances. That's a really good call out. So last question is for both of you. So now that you've gone on this journey and I know it's still continuing of using Amazon EKS, I'd love for each of you to give some advice to someone who's listening who is at the very [00:18:00] early stages, they want to be able to start using Kubernetes. Maybe they also don't have that experience as well, just based on your experience, what advice you'd have for them. So Daniel, why don't you start

Daniel (18:11):

First is to, we have a lot of AWS documentation and resources. You can start with our formal AWS documentation about A WSE Ks and best practices and getting started with EKS. After that, you can explore the project of data on EKS. It provides a lot [00:18:30] of information and open source tools and resources, how to get started with running Spark workload, specifically spark workloads or other types of open sources that are part of the data on EKS project with your project. So if you have a project that is part of data on EKS that already supported, I heavily suggest you get started. And we have our cloud formation templates or CDK templates to get started with Amazon EKS and data [00:19:00] on EKS project. And if you are working and you have the AWS account team, get in contact with the account team with your AWS account team to better collaborate with accounting. They can provide you the resources knowledge and connect you with the correct personas inside AWS to provide you the best practices for your specific workloads.

Jillian (19:24):

Benty, what about you?

Bentzi (19:25):

So for me as a developer, I would just say don't be afraid from all of the scary Kubernetes [00:19:30] terms. Don't be afraid. They're very simple once you get to know him, it's okay. Don't be afraid. Use open source tools, use carpenter. Use Spark Operator. If you are using Spark, you can use one of the many solution for logs you have. Use them. Don't be afraid. They're very nice and easy to use. And when you're getting to big scale, so first of all, understand that this is a challenge and ask help from the A Ws team because for us, once we reach a very large scale, they help us [00:20:00] a lot. And it was really a game changing help. So yeah, I think that's my message.

Jillian (20:08):

I love that advice and especially the one about yak, Kubernetes. I mean, when I go through the documentation, when I talk to customers, it really does feel like a different language with all the different terms and how you have to think about things. So I appreciate you saying that. It's not as scary as it sounds. Daniel, thank you so much for being here on the AWS podcast. I know both of you have really inspired someone else who's listening who wants to be able to make that [00:20:30] journey into Amazon EKS. So thank you so much for being here today.

Bentzi (20:34):

Okay, thank

Daniel (20:35):

You Jillian. Thank you.