

**EPISODE 799**

[INTRODUCTION]

**[00:00:00] JM:** Log management requires the processing and indexing of high volumes of semi-structured data. A log management service takes log data and puts it in a cloud-hosted application so that application operators can access the logs to troubleshoot issues.

A large tech company will produce terabytes of logs. Those logs are produced on the host where the service is running, and then a logging agent on that host will transfer the logs to the log management service in the cloud. Once the logs are in the cloud, they are parsed and indexed and stored in a way that's easy to query.

In 2014, Renaud Boutet, cofounded Logmatic, a log management service that eventually become a leading provider. Logmatic was acquired by Datadog, and Renaud now works as a vice president at Datadog.

In today's episode, Renaud joins the show to talk about the architecture of a log management service, as well as the product development. We talk about storage tiers, scalability requirements, failover strategies and logging for serverless functions.

Full disclosure, Datadog is a sponsor of Software Engineering Daily.

[SPONSOR MESSAGE]

**[00:01:24] JM:** Hired simplifies the job search for engineers with a data-driven, personalized matching process. Head to [hired.com/sedaily](https://hired.com/sedaily) and create a profile today. By creating one profile, you'll be matched with over 10,000 companies looking for engineers like you. Hired uses intelligent matching technology. Data science with years of experience matching engineers with job, and Hired also has a human in the loop. Hired gives personalized career coaching to match you with opportunities based on your skills, industry, interests and desired salary.

Create a profile today at [hired.com/sedaily](https://hired.com/sedaily). Find a job that you truly love that is personalized to your background and your preferences, and if you aren't an engineer, Hired also helps designers, engineering managers, product managers and other tech workers find their dream jobs. Just go to [hired.com/sedaily](https://hired.com/sedaily) and check it out.

[INTERVIEW]

**[00:02:40] JM:** Renaud Boutet, you are a VP at Datadog. Welcome to Software Engineering Daily.

**[00:02:45] RB:** Thank you for having me.

**[00:02:46] JM:** You founded a logging company called Logmatic back in 2014. It was eventually acquired by Datadog, but I want to talk a little bit about building a logging company, because there are many successful logging companies, and the product category of logging is kind of interesting to me, because you see newer logging companies come out all the time. They still get started and they manage to be successful.

What are the different ways that logging products differentiate from one another?

**[00:03:19] RB:** That's a very good question. So to explain Logmatic, I think I have to explain like where we are coming from, me and my partners. We started this company called Focusmatic first out of the company we were all coming from, like Quartet Financial System, and we were building business intelligence solution. So particularly, pivot tables.

If you think about pivot tables, it's all about like aggregating tons of structured data, which is different from logs, right? And to like make it easy for users like that are non-technical to dive into all these dimensions and mergers and everything. So we did that for 7 years, and then we started Focusmatic.

Focusmatic was actually a social analytics engine. So basically we are collecting tweets, Facebook posts and like all the social – I should say logs, like from anywhere. We decided then two years later to migrate to pivot to actually Logmatic, because we decided that it was more a

part of our DNA to build products for engineers. Actually, the log management market for us seems like much more promising.

So how did we really differentiate ourselves? I think if you see where we were coming from, we were building simple solutions that were like aggregating lots of data, like simple to use solution. I think all of these log management tools before, there were actually coming from – It was like administrators from servers that were actually getting these logs and then they were doing queries on terminals and etc.

So like most of the industry were coming from there, and we decided to do something like much more easier to use and with much more analytical queries into the user interface. So that was a good differentiator for us. I would like to say that Datadog was not the source of inspiration for us. Combine that with like the easy to use integration thing. Back in 2014, there was not a lot of SaaS log management solutions in Europe to compete against. So we did pretty well actually.

**[00:05:36] JM:** Why does it matter that you're in Europe. Isn't logging kind of an international business?

**[00:05:42] RB:** Yeah, yeah, but I think like if you look at all these SaaS solution for log management, they are mostly Americans, and I think they were focusing on the market, on the local market, and they were not in Europe yet. So by focusing in Europe, we didn't have like a lot of competition at that time. Even if we had it, I think like what we had, like the ease of use that were proposing and something that is really the motto of Datadog, that that's why Logmatic and Datadog were so compatible. This simple, but not simplistic motto is like answer to very complex use cases, like on the simple manner. But if your users needs like to go further, you're going to be able to do it with the user interface. So this plus that, like log competition at that time and this simple but not simplistic made us pretty successful in Europe.

**[00:06:42] JM:** Tell me about building an going to market in a somewhat competitive space, like logging, and what was the process for iterating on the product and getting feedback from people. One of the reasons I ask is because there's a lot of people that listen to this that are building developer-facing products. So they're selling into the market for developers, which is

kind of an emerging market to be selling to. So I'd love any wisdom that you've learned for building and selling products to developers.

**[00:07:15] RB:** So when we were working for a financial market in our previous company, we're actually selling a product for developers. I've been doing that to all my carriers. So this is not new. The only difference is that we were doing that on premise, while now we wanted to do it in SaaS.

So now to answer your question about like how did we enter into the log management market? I'd say like we were actually doing tweet analysis, Facebook personalities, and we were actually selling to marketers and people doing communication to understand like the impact of some tweets on the brand, on some brands. Actually, we pushed to go into the log management market, and we were very young entrepreneurs.

So I think the first product we did was actually a failure, because we didn't do the market study. We didn't do like everything we should have done, and we decided that if we had to enter into the log management market, we will have done it like – Now we wanted to do it right. So it was in 2014, summer, we decide to hire an intern that is going to help us with that, and I'm going to spend also be full time on that. We do a market study. We actually reach out to a lot of companies, mostly in France, because we were located in Paris. We show this product that was analyzing tweets, and we were all developers in that company. So we knew how to talk to developers.

So we sent logs into this product that was actually previously analyzing tweets and Facebook posts. At the time, Elasticsearch and Kibana was starting to get some momentum, but people wanted to have managed services, and we had some qualities as well that was differentiating us from Kibana. So out of the market study – So I think we did like 80, almost – We reached out to 80 companies. We started like 15 projects right after actually.

One of them was a company called Canal+. Canal+ is actually pretty big in French-speaking country, and in France in general is the – I call it the French HBO actually. Like the French HBO, and we are a young startup, they wanted to actually run their logs through Logmatic, which will

be Logmatic in the future. That was really the starting point of this adventure actually. A market study on developers showing the product, and then we pivoted to log management.

**[00:09:54] JM:** You were eventually acquire by Datadog. Why did you decide to join Datadog and what was the acquisition process like?

**[00:10:03] RB:** So 2014, we start, and with 10, 15 projects, and almost then become customers. Then a few months later we successful raised money and everything starts going fast. We hire sales people and then we focused entirely on log management. For us, log management was very natural, because we were developers and we're using logs internally. So we were doing Dogfooding as well, and we were actually pretty good at this. So great growth, etc. Everything goes super fine. So we grow – Even at the end, when Datadog acquired us, we grow more than 10% month-to-month. We were actually one of the hard startup in Paris. So why did we sell into Datadog, right?

So Datadog approached us multiple times, and Datadog was one of our biggest customer as well. Datadog tried most of the log management solutions out there and they decided that we were the best. Because Datadog was also probably funded by French people they were – And Datadog has an office in Paris as well, building an acquisition in Paris was making a ton of sense for them.

So they come to us and they make a first offer, and we didn't want to take it. I mean, everything was going fine, etc., and we wanted to just continue. This is really when we started to raise money in the time. So we were like raising for Series A this time, because were going fine. Hundreds of customers. Datadog just come again and say, "You should really join us."

I say Datadog was, for us, a source of inspiration. I was the product manager there. I've been a developer for years, but somebody has to talk to customers. So I became a product manager without even knowing I was becoming to be a product manager. I was taking inspiration from Datadog, and Datadog was really a reference in the monitoring market. In the right way of providing easy to deliver value to developers. With the devops culture going on as well, developers operating their own applications, Datadog was really like the solution we were looking at.

Especially on the feature that we were developing at that time, integrations, so we were honored that Datadog wanted to talk with us. We talked a lot and we decided that actually we were almost the same. We were smaller, obviously, but we were the same. Since we joined them, it's going very, very well.

[SPONSOR MESSAGE]

**[00:12:57] JM:** As a software engineer, chances are you've crossed paths with MongoDB at some point, whether you're building an app for millions of users, or just figuring out a side business. As the most popular, non-relational database, MongoDB is intuitive and incredibly easy for development teams to use.

Now with MongoDB Atlas, you can take advantage of MongoDB's flexible document data model as a fully automated cloud service. MongoDB Atlas handles all of the costly database operations and administration tasks that you'd rather not spend time on, like security, and high availability, and data recovery, and monitoring, and elastic scaling.

Try MongoDB Atlas today for free by going to [mongodb.com/se](https://mongodb.com/se) to learn more. Go to [mongodb.com/se](https://mongodb.com/se) and you can learn more about MongoDB Atlas as well as support Software Engineering Daily by checking out the new MongoDB Atlas serverless solution for MongoDB. That's [mongodb.com/se](https://mongodb.com/se). Thank you to MongoDB for being a sponsor.

[INTERVIEW CONTINUED]

**[00:14:19] JM:** So let's get into some engineering discussions. Now, log management is interesting, because at a high-level, you have a log message and it's a blob of semi-structured text, and it gets created by our machines and there might be thousands of these messages that are created every second across an application. So it's high volume. A developer needs to be able to use these logs to resolve issues. So you need to be able to index them intelligently. So there are all these tradeoffs, because you have this gigantic volume of data. You may need it very quickly and then you may need to save old logged messages in order to triage issues as they're coming up.

Describe the problem of log management as you see it and some of the technical tradeoffs that are made in building a log management architecture.

**[00:15:14] RB:** Okay. So when we're doing Datadog, we had to kind of – Of course, we recovered lots of our backend source code and etc., because we resolved lots of the problems, and there are tons of problems like handling the large volumes you are mentioning here. The fact that everything is very heterogeneous.

So we recovered lots of our source code, but actually Datadog wanted a seamless experiences. So we all agreed that it was the best [inaudible 00:15:45] to just trash everything and restart from scratch. Actually, people came to me and said, "Oh! That must be very boring to do the same thing twice." Actually, that's not boring at all, because there aren't a lot of teams out there that has the chance to reduce something more perfectly and like be able to just remove all the little failures that are accumulating on the product and that we call legacy, that we could not remove because we got so many customers that it's hard to change, that here we could really – We restarted from scratch and we could have just do something more perfect.

There is also something else. When you do log management, there are problems that everyone have that are not addressed by the market, but that everyone wants to solve. So I'm going to take one that I particularly like, which is the large volumes you're referring to. So one of the problem in the log management market is people have tons of logs. They have too much logs. There are actually three criterias attached to logs that we don't like really mention; large volumes, large variations as well. It can spike at any moment, like if you deploy something wrong, or if you are attacked. It also varies just normally like with your business. When people sleep, you have less logs.

But the third criteria is probably like, at least for me, it was very important as well and it was not addressed by the market. The value for the company actually also vary overtime, because you vary overtime, you vary over situation, you vary over users. An example of that is that if you have [inaudible 00:17:37] logs that are very like in terms of like tons of volumes, people tend to discount them and say, "I'm not going to retain them, because they're not variable to maintain my operations."

The reality is that if you have an incident now, now you need the maximum amount of data. So now I need to get access to them. The value of these [inaudible 00:17:57] also varies. With Datadog, we also try to address that big time, because as one of the founder of the Datadog told me one day. He said, “We don’t do log management today. So you can do anything you want, and we’re going to see what it looks like. As long as it’s good for the user, let’s try.” So we came up with that idea, logging without limits, and I can describe it more for you, but I want to be sure I answered the question.

**[00:18:32] JM:** Yeah, sure. I guess explain in more detail this logging without limits architecture you’re talking about.

**[00:18:39] RB:** So logging without limits is based on this fact, like all the log management solutions out there – And Logmatic was one of them, are actually charging for all the logs that you send to the system. So one of the problem it creates is that – Especially as we like most of the solution charges by gigabytes per day, if you have a high spike, you’re going to – The solution is going to come to you, or the vendor is going to come to you and say, “Hey, actually, we saw that for like a whole week, you were actually consuming 200 gigabytes a day while your plan is actually 150 gigabytes a day.” That’s actually like ridiculous, because maybe this extra capacity was actually not used by the user. It was not even useful.

So we really knowing that problem, and the tradeoffs, the users were doing then, is that they were saying like usually the manager come to you and say, “Can you stop logging this please, because it’s too much logs and the log management solution is starting to easy to maintain if it’s open source, or expensive if it’s vendor provided.” But that’s pretty bad, right? You don’t want your log management solution to actually work against you. You want your log management solution to provide solutions to your problem and removing logs, forcing people to remove logs that might be useful at incidental moments, for instance, or for any other use case. Actually, it can create a lot of problems.

So we realized that what was costly, and I think it’s based on the very simple idea, right? What was costly is actually indexing and retaining the data to run all the queries that are necessary doing a troubleshooting session. However, something that was not that costly was to ingest the



logs, to process the logs, and we can even provide [inaudible 00:20:38] on all these logs. So which means that you can keep some kind of visibility, like the real-time visibility on the logs that we're ingesting.

Actually, if we let the user define dynamically what he needs based on the situation, then we will bring a lot of flexibility on the control of the volume and the control of the actual value of these logs coming. So we came up with this idea to just instead of showing one price tag, which is the traditional gigabytes per day, we day, "Actually, what you're going to index is going to be charged separately from what you ingest." Then you can decouple the ingestion and the indexing. Something I should mention now as well is that it's not because you don't index that you lose the log, because we can also push the log back to your archives.

Generally, now, we have companies that are sending, let's say – That can send like one terabyte worth of log. They don't have the budget to index the one terabyte per day. So they are going to put the budget on the table for 100 gigabytes, for instance, because this is what they need to maintain their daily operation. If they need a real-time access they can [inaudible 00:21:57], and if they need to do an audit on logs that were retained, all the company operation history is still archived. The one terabyte actually are pushed back into the archives, and then you can use these archives for further analysis, or because you made a poor decision to exclude them from your indexes.

Does that make sense?

**[00:22:18] JM:** It does. I want to take a step back, because I think some people who are listening may not be working closely with logging, and so they may not know the path that a log message takes. So I want to talk a little bit more abstractly and so we can talk through some of the tradeoffs of the different steps.

So the way that I think about the path that a log message takes is first of all you've got an agent installed on your server, and the agent is taking the log messages that are produced by that server and forwarding them to a log management provider. So the log management provider does things like ingesting the log, figuring out what to index, figuring out what to store. Maybe doing some kind of stream processing, maybe building machine learning models around that

high volume of log data that's coming in. Maybe doing some compression on the logs. Maybe doing some summarization of the logs in case you don't actually want to store everything to be accessible immediately.

Then you also have decisions to make on the agent side of things, this agent on your server. You may decide you don't actually want to send all of your log messages to the cloud logging provider that you're using. You may decide, "I only want to sample logs every 5 minutes, or I only want to sample the logs that have exceptions in them," for example.

So I'm just trying to present the framework for all the different tradeoffs that can be made and I think the logging without limits idea that you presented is mostly to address the problem that it's hard to figure out what are the same defaults for log management, because if you set up your log management solution and you're just logging everything and then the logging provider is naively storing and indexing everything, you can end up paying way too much money, storing way too much data that you don't need. So you're not really like trading off things intelligently.

**[00:24:24] RB:** Yeah, exactly. People today – Or I mean not the companies working with Datadog obviously, but historically at least. As you correctly say, like they do tradeoffs at the agent level or at the application level. So you have two choices. You can say to your developer, "Remove these logs from the source code. So the log won't be generated." It can be a user had the problem at that level into my application for instance. So you remove this one because it's too noisy. The other one is about like setting a filter at the collecting agent level where you say, "All the logs that look like this, I don't want to send it to the log management solution."

The big problem – Let's focus on that second solution. The first one for me is the worst, because if you remove it from the application, it means that if you now need it, you need to really plug in the application, or you just don't do. The second one is probably less intrusive, because you have more control on it. However, usually the developer that needs to get access to these logs don't have access to the agent, because the server – It doesn't have access to the server. So it cannot just redeploy the filtering rules. Which means that he ask somebody to change the filtering rules so he can get access to his logs, which means it's a very painful process.

Here, we ingest everything. Everything is processed, which means parse and everything, and I can cover that topic later, because parsing and processing logs is also a very, very big topic. It can decide – Like the manager I was referring to before can say, “You actually have the right to have 50 million logs per day,” for instance, “and if you go overate, we’re going to just rate limit you.” That’s a choice they can make, or they can decide to go over, “but please do everything you can to just stay under that volume.”

This is the developer who actually knows what is the inherent value of these logs. So this is one that is capable of doing the real-time tradeoffs. So inside the user interface, they can just change that dynamically and control their volumes, which means that we put the return on investment in the hands of the end users who know what returns these logs are going to provide to the team.

Does that make sense?

**[00:26:58] JM:** Yeah. Let’s talk a little bit about architecting this kind of solution in the context of the modern, I guess, storage hierarchy available on cloud service providers. Datadog itself, you’re built on the cloud and you have a choice of different storage tiers. You’ve got S3. You’ve got databases. You’ve got disk storage. Describe the different storage tiers available to you, and from your point of view, when you’re looking at storing just tremendous volumes of data, where do you store different data availability levels and how do you evaluate the cost tradeoffs?

**[00:27:49] RB:** It’s a very good question, the hot topic for us. So basically, I talk about two kind of surges today; indexing, indexes, and what we call archives. Some people like call it backups. Indexes is not a way to store data actually. It’s just a way to retain the data that is really useful to retain for 7 days, 15 days, etc. This is the expensive storage.

So why it’s expensive, it’s because – So it’s working on disks and most of the time on SSD disks, and also it’s designed to be able to answer very fast to pretty much any kind of queries, and this is what the users want. Because when you troubleshoot, you need to go fast and you don’t really know in which dimension – If the IP address index column – I may call it like this, it’s going to be more important than the user one, or the URL one, or the specific query parameters. This URL is going to be more important.

So we need basically like logs. I'm not going to say something obvious here, like I can come with like tons of different attributes, because these logs are coming from tons of different technologies, applications, etc. We need to index everything, and this is why it's costly. It's because it has to go fast. It's very heterogeneous, and that's it. So retain them, and this is the part that is expensive.

However, some users are confused by the fact, as I said, this is where you store your logs. This is not where you store your logs. You store your logs into what we call archives, and archives generally – You can compress the logs and you're going to put that into, for instance, S3 storages and more likely after six months into [inaudible 00:29:43]. That's even like a simple vision of the tradeoffs we can do. We can do much more than this. However, today, in Datadog, that's the two tiers, if I can call it that way, that we are supporting.

So that being said, as all these logs are going into cold storages – So usually the compression rates on logs is going to be 95%. So it's going to be 120 of the original size of the log. It's very good to compress these logs, and because you don't index them into your archives, it's going to be like pretty cheap to keep them, and it's fundamental to keep these logs, because you never know when you need them. Sometimes you can have technical audits.

I had a problem during that period of time and I never had time to just figure out what is happening, and I'm frightened by it. Let's take some time to analyze this log. Security audits is one of the more obvious, like I discovered that I have a potential breach today. I want to know if it was exploited or not, and business audits as well. Because if you think about logs, they represent the history of the company operations for most companies out there.

All these companies running applications and everything, they are serving their customers through applications and services and these infrastructure they are hosting, right? So these logs actually represent the history of the company. So it's fundamental to keep it.

Now, we are working on providing solutions to rehydrate on-demand from these cold storage, and this is something like – We actually have a dedicated team of engineers and people

working on these matters, because this is only starting. Cold storages today are too slow to be used for debugging use cases. In the future, it might be different.

So right now we're going to rehydrate, we're going to offer possibilities to rehydrate from these archives on demand directly from the user interface. In the future, we might do much more and provide multiple tiers, some that are slower, but that can still offer some functionalities, etc., but it's so important for us because we think it's the future, and because the amount of logs keeps growing, that we need to provide cheaper solutions for the volume that we are facing.

**[00:32:06] JM:** Definitely. It is quite interesting to see the – I mean, I just remember in college, in some computer architecture course, when I first introduced to the memory hierarchy and they're like, "Okay, here's disk and here's the memory and the L1 cache and the L2 cache. If you go to this caching layer, it's a small latency penalty. If you go to this cache hierarchy, it's a slightly greater latency penalty. Then if you go to disk, it's a much greater penalty." Then it's like if it's on a certain page, it's a great difficulty, greater latency. It's just interesting to see these themes in computer science that you get introduced to and they continually reemerge, and now they're reemerging in the form of like different cloud storage offerings.

**[00:32:57] RB:** You're totally right. All these layers you are referring to, obviously we use them intensively when we do all these queries and everything, all these caching layers I use all the time. They are used sometimes by intent, because the developers are actually specially building caches and that kind of things. Sometimes because the architecture of the computer is doing like magical stuff all by itself.

Here, the tiers I'm referring to is the disks, and a cheaper disk. It's even more than this, because we are even thinking right now should we propose an ADFS Hadoop tier where you will be able to do queries. It's not going to be [inaudible 00:33:39] into the hot storage, but maybe like real-time queries on Hadoop, and that will be cheaper and so on and so forth.

**[00:33:47] JM:** I'm sure there are some customers that would want that.

**[00:33:50] RB:** Yeah, yeah, and right now we do a stream, because we have to – If you fight like 10 things, we try to focus on things that are like more important, but this is definitely in the table,

like having multiple kind of endpoints to export logs to somewhere or some places or maybe hosting it directly on Datadog. Yeah, change to pricing of storage accordingly explaining the tradeoffs the user is expecting.

[SPONSOR MESSAGE]

**[00:34:26] JM:** Today's sponsor is Datadog, a cloud monitoring service that unifies logs, distributed traces and metrics in one place. Datadog's approach of logging without limits enables you to cost effectively collect, process and archive all your logs. Explore and query a live tail of all your processed enriched logs and use the log patterns view to group logs for seamless investigation and troubleshooting.

Try Datadog logging for yourself with a free 14-day trial by going to [softwareengineeringdaily.com/datadog](https://softwareengineeringdaily.com/datadog). Datadog will give you a free t-shirt at [softwareengineeringdaily.com/datadog](https://softwareengineeringdaily.com/datadog), and t-shirts can be quite comfortable.

Thanks to Datadog for being a sponsor of Software Engineering Daily, and go to [softwareengineeringdaily.com/datadog](https://softwareengineeringdaily.com/datadog).

[INTERVIEW CONTINUED]

**[00:35:27] JM:** I'd like to know more about what the internal infrastructure questions that you're grappling with today are like at Datadog, because we can talk about this from the perspective of how individual customers are coping with their log management. But I'm actually quite interested in Datadog itself, because it's at such a high scale and it's built on cloud providers.

I've talked to a lot of different companies who just have really expensive cloud bills and sometimes the cloud bill like climbs up in ways that they didn't expect, they didn't foresee. You may see the rise of these cloud cost management companies. There's like 10 or 20 companies that are based around consulting and engineering solutions for cloud cost management, because cost just get unexpectedly out of control.

So how are you grappling with that internally at Datadog, like negotiating the cost management of Datadog itself?

**[00:36:26] RB:** Okay, and that, as you can imagine, is a very, very big topic for us as well. We are very, very close from AWS today. AWS is really the historical cloud provider of Datadog, and it's still like the very first one. Datadog provides three products, like we call them the three pillars of observability, like metrics. Basically, like it's monitoring the infrastructure and providing metrics. It's time series basically. So that's one type of storage that it costly for us to maintain. I'm not even mentioning the intake and everything.

We have APM, so application performance monitoring, where we have to just – Every time one of our customer generate a query, we actually keep what we call spans that are – Actually, where I lose each piece of code or processes where I lose some latency on the query. So it's actually tons of data, because each request, like every time you go – I don't know, on Airbnb for instance, you're going to just do a request and then you're going to generate without knowing 50 pieces of information that Datadog is going to keep and logs.

I'm going to focus on logs, because everybody knows or at least like the people working in the monitoring market, that log management is traditionally the one that is very, very sensitive, is very costly to index logs, and it's very difficult as a business to sell something at the right price without losing money.

Of course, that goes for like optimizing our own source code. I'll get back to the acquisition, but one of the reason acquired Logmatic is that we were one that they've seen that we were doing a pretty good job, like doing optimizing and like the cost of each log on the system. But that also goes through negotiating with the cloud provider. Each companies, what we call an EDP, which is a special contract, where because of the volumes that we have, like the capacity that we are requiring, that we can negotiate the price with them. So we do that.

Something that we do as well is Datadog is not multi-cloud. So we have multiple clusters ring around the world working on multiple cloud providers. So we are not relying on a specific service on each cloud provider. We're actually running entirely on Kubernetes now, and we are able to just move from one cloud provider to another. Of course, that gives us a leverage in

negotiating prices, of course. Also, if a cloud provider does a bit of job or had a bit of price, we can just move faster.

**[00:39:15] JM:** Tell me more about the multi-cloud architecture. So one question I'm curious about is I'm sure you have a failover in the case of an entire cloud provider going down, but what I'm curious about is if you have tested that failover.

**[00:39:31] RB:** Yeah. So, okay. So we have multi-region and multi-cloud. So multi-region is like we have an instance in U.S. and we have an instance in EU, and the reasons for that is because we have lots of regulatory – The companies walking with us, they have lots of regulatory compliances to fulfill and they need all their data to stay in Europe. So we need it to a multi-region.

We also do what we call disaster recovery, and also – So providing the ability that if AWS is going to fail one day in North America, we're going to be able to just run the same services on another cloud provider. So we have this today in place for some of our customers. So we tested it internally, but we never had to use it, because we never had a major failure that forced us to use it. Of course, this is really the customers with the highest level of requirements in very critical businesses that are requiring this capabilities.

I hope I answered the question here.

**[00:40:40] JM:** Yeah. Yeah, that's enough of an answer. So let's zoom out. You're the VP of product management at Datadog. Describe your long term vision for the product.

**[00:40:53] RB:** So actually, there are two VP of products. So one that is just sounding like everything related to metrics and time series, and I'm really focusing on log management. But for us, logs is actually – The pure definition of what is a log is it's an event with a timestamp basically. So if you see that like this, there are actually tons of other products that we can do at the top of this platform.

For me as a VP of product of east log management platform, we are currently building actually a platform to build other products. Datadog really aims to serve like old observability challenges



and needs that are customers have. So we just released Synthetics, for instance. So Synthetics is like to test that an endpoint, an HTTP endpoint, or your webpage, is up and the performance like according to where you are in the world.

So we just released that as Datadog, and this feature is not using the log management platform, but we're also working in what we call real user monitoring, which is collecting like logs from mobile and web browsers for troubleshooting purposes as well and just because sometimes a JavaScript is not going well for some reasons on some version of the browser, etc. So we are building that, and we are even thinking about now network monitoring, and network monitoring is based on a very special kind of logs that we call flow logs, where to describe it, it's really which process, like which host is communicate to where and the amount of data that was received or transmitted from a process to another.

So it's very structured logs, but they are very interesting, because for secret reasons, but also just to see what is going on in your infrastructure. Sometimes services communicate with each other and you don't even know. So it's important to provide maps of the infrastructure automatically generated by these flow logs.

Another use case I've seen with flow logs for instance is cost management. As you said, everybody right now has some challenges with like the cost of these flexible infrastructures provided by cloud providers, and like just knowing the amount of network I'm using provide fine grained reports to say, "Actually, these applications can be moved from that [inaudible 00:43:28] so I can save money on that." So that's for flow logs. We're also building some other products.

For me, as a VP of product, the log management solution is now reaching a state where we are transforming it into a platform and lots of older product managers and teams are going to leverage these capabilities that we have to do other things that they're going to be even-based.

**[00:43:56] JM:** Has the roadmap changed at all, or have you had to make any significant adjustments as people have used more and more serverless infrastructure? Because sometimes with like serverless stuff, like AWS Lambda, the underpinnings can be a little opaque and they can be harder to debug. Things can go wrong unexpectedly. What's your sense of how logging around serverless is advancing?

**[00:44:25] RB:** Yes. So it's also things like we have dedicated teams for serverless now and we also released a product that we call cloud foundations internally that is actually about collecting the metrics and the logs and the traces from functions, because there is definitely a gap here. Lambda functions for instance are – They are not new, but the monitoring is still challenging. Logs, because we don't have underlying infrastructure, is like really what people use today to understand what's going on with the hundreds of thousands of functions that they have and communicate.

So, yeah, we definitely connect logs from these cloud functions, and I say cloud functions because it can be Lambda if it's AWS. It can be Azure functions. I don't think we use something right now for Google cloud, but we are most likely going to do something as well. The goal of Datadog is being agnostic, because we see most of our customers moving also being multi-cloud. So they cannot rely on some monitoring provided by the cloud providers themselves. They need like a full visibility across providers.

So, yes, Lambda function, and we have dedicated teams for that that are actually working across products, not necessarily on logs, but logs is just one of the recipient of what the Lambda function can provide.

**[00:45:49] JM:** You mentioned the O world earlier, the observability, which is this vague kind of buzz-wordy term that has nonetheless entered our vernacular as developers. Are things changing around what people want out of "observability"? I mean, one thing I can imagine is there's just a deluge of observability data. So maybe people need more narrow interfaces, but what are the changes in observability requirements that you're seeing?

**[00:46:20] RB:** The industry in general is actually moving to what we call – Datadog was one of the first to provide the three pillars of observability. I think what people want first is really to have a seamless experience between the different solutions, because one other thing we just provided and that's very unique is that when you – If you are a Java developer, you're going to use what we call a Java tracer, a tracing library. Basically, it's a library that you're going to pull into your application, and this library is doing the magic.

APM is – You should do another show with APM, because APM is like an incredibly, fascinating product as well. The tracing libraries here are actually doing magic, because to auto-instrument the codes to tell you which methods is consuming more – Is slower to execute and provide you a breakdown without doing anything.

So what we just added here is if we have these transactions that are going through all these layers of functions inside your code and which measures everything and we are able to just propagate this request between microservices and everything, because this is really the magic the APM is doing. There is one thing that I heard, like at Logmatic and at Datadog as well, is that, “Oh, do you have a solution to propagate the request ID into the log so I can have a full visibility on the transaction from a log perspective?”

Usually, companies are trying to do that, but very unsuccessfully, because it’s very hard to propagate context when a microservice is calling another and it’s calling another. But APM, this is what the APM does. So what we did is you just have to turn on a configuration on the tracing library and then all the logs that are generated inside the application are now going to know the transaction context, and we’re going to just inject that.

When these logs are going to come to Datadog, then we can exactly tell you which logs was generated by which thread into your application, in which function, in which transaction, and that’s totally unique. This is what people are trying to do on the last few years at least, where I work on that market, right? I think people now are looking for synergies. They are really looking for – And this kind of incredible value that we provide with that feature, for instance, is it’s only possible if you are able to just run the two solutions all by yourself.

So all the industry understand that, and if you see companies in that industry that are starting on log management or infrastructure monitoring or APM, they are all going into that direction, because this is really what people want, all solutions under one roof. I think like the companies that are going to do well in that market are the ones that find the little things, like the synergies between products. That’s the bet Datadog is making right now, and apparently our users really love what they are seeing now.

**[00:49:43] JM:** Okay. Renaud, thank you for coming on the show. It’s been great talking to you.

**[00:49:46] RB:** Yeah, thank you. Thank you for having me.

[END OF INTERVIEW]

**[00:49:51] JM:** GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use, and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plug-ins. Use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

With GoCD running on Kubernetes, you define your build workflow and let GoCD provision and scale your infrastructure on-the-fly. GoCD agents use Kubernetes to scale as needed. Check out [gocd.org/sedaily](http://gocd.org/sedaily) and learn about how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team who have talked about building the product in previous episodes of Software Engineering Daily, and it's great to see the continued progress on GoCD with the new Kubernetes integrations. You can check it out for yourself at [gocd.org/sedaily](http://gocd.org/sedaily).

Thank you so much to ThoughtWorks for being a longtime sponsor of Software Engineering Daily. We are proud to have ThoughtWorks and GoCD as sponsors of the show.

[END]