



## The Deloitte On Cloud Podcast

**David Linthicum, Managing Director, Chief Cloud Strategy Officer, Deloitte Consulting LLP**

**Title:** CaaS: democratizing access to containers to power cloud value

**Description:** Containers power cloud flexibility and portability, but over time container technology has become complex and expensive to deploy. In this podcast, David Linthicum talks with Portainer.io CEO Neil Cresswell about how Portainer enables Containers-as-a-Service (CaaS) to democratize access to container technology by centralizing container configuration, management, and security. With CaaS, companies can gain the benefits of container technology without having to navigate all of its complexity.

**Duration:** 00:24:37

**David Linthicum:**

Welcome back to the On Cloud podcast, your one place to find out how to make cloud computing work for your enterprise. This is an objective discussion with industry thought leaders who provide their own unique perspective around the pragmatic use of cloud-based technology. Today on the show we are

joined by Neil Cresswell. Neil is the CEO and cofounder of Portainer.io. A New Zealand-based provider and powerful management platform that enables organizations to deliver containers as a service to their users with over 25 years of experience in systems engineering, IT consulting and IT management. Neil is dedicated to enabling businesses to successfully adopt containers as part of their transition from legacy the cloud, building on this background. Neil built Portainer in 2017, relatively young company, to address the most pressing needs of container adoption. So, I have some more details here about the company, but why don't you tell me what the company's all about, Neil?

**Neil Cresswell:**

I like to say we are designed to democratize access to container technology. And what does that mean? It simply means make the technology available to the masses, to the mainstream, technology like Docker, like Kubernetes. It's absolutely staggering. It's an absolute game-changer in delivering digital services to end users, but it's complicated. And, unfortunately, it seems to be only those that can adopt the technology are organizations with enough people and money to be able to invest in the skills required to adopt the tech. So, those organizations that are a bit smaller, that can't afford to pay for consultants, or can't afford to retrain their staff are kind of left out. And so our goal is to include them into this Docker and Kubernetes technologies through Portainer by making it dramatically easier for them.

**David Linthicum:**

Now, what I like to hear is the stories in terms of how this was thought up. And, in other words, I come from a background as being a CEO of a tech company, CTO of many tech companies in the past. And there's always that eureka moment, where someone has an idea for a company, and not only just an idea, but the ability to kind of put the effort, money, and capital, and risk behind it to make it a success. So, when did you come up with the idea? Who came up with the idea? Was it you, a bunch of other people, yourself? How did you decide to start the company? I love hearing those stories.

**Neil Cresswell:**

Well, some of that you actually went through when you went through my background, right? So, I'm a career consultant and have spent a lot of time advising organizations how to build infrastructure platforms, how to run those platforms, Day-Two operations. And that was predominately individualization, network-storage technologies, but that was how I built my career. I transitioned from a consultant, into customer-land, when I became CEO of a cloud-service provider, and in that provider I had to go and deliver virtual-machines-as-a-service initially, but through that journey got myself exposed to Docker technology and thought, "Oh, my goodness, this is yet another wave of transformational tech like virtualization was. This is the next wave. I need to get a piece of this action," and started trying to deliver containers-as-a-service, as a service offering back in 2016, 2017.

But it was well before containers-as-a-service really existed as anything other than in someone's brain, and did it like everyone was doing. Here is an API endpoint. Customers, have fun. And the adoption of that technology was just very, very low. The market could not understand how to embrace a cloud service that was just an API endpoint and some auth tokens. How do I use this? Everyone was so used to Web portals and—yeah, it's a very Microsoft world. People are used to logging into Windows and clicking away. To transition from that to pure command line is the realm of genius, and there was a real struggle getting uptake and adoption.

At the time, I went and looked for products on the market to say, "Well, what can I do in regards to a self-service portal that I can put across this Docker service?" And there were not really many in the market that suited delivering a container service offering. And, so, I went about building the precursor to Portainer. Through that journey I met my cofounder, Anthony Lapenna, and he helped me build the early version of Portainer. And from that point on, we scaled it ever since. So, the two of us caught on to this idea of a really simple self-service portal that can help users engage with and consume our container services. And that's really where Portainer came from.

**David Linthicum:**

Yeah. One of my mentors many years ago once said something I always think of, and I thought it was very profound: "You need to make simple things, simple." And the reality is, as technologists, we have a tendency to make simple things complex. And I was doing that in building some of the products out, and this person came to me, and he said he liked the product, liked the capabilities, but no one could use it. Reality is we have to get this out to a larger market. We have to make this technology work in the context of how people want to consume it. And that was something that I missed at the time, but I kind of take as a life lesson moving forward. Was it a similar epiphany that you guys found and really kind of—just simplifying the use of containers?

**Neil Cresswell:**

Absolutely, and at times we're, even today, we're still ridiculed for wanting to make this tech easy. There seems to be this love affair of how complicated this tech has become. This is one of the first times in my entire career where I've watched our technology get more complicated, rather than less. If you looked at the early days of virtualization, it was the realm of the expert. It really, really was. I went through a massive amount of training to learn how to deploy a virtual-based infrastructure, and then in the five years after that, it became easier and easier and easier, where a three-day training course, and you were off.

If you look at Docker and Kubernetes technologies, it's the inverse. It's become so powerful. It does so many things now that the complexity is just scaling and scaling and scaling, and it's harder now than it was four years ago to actually use this tech, and that's crazy. And that's the whole premise of Portainer. It's like, "Okay, yes, we agree that this technology is amazingly capable, infinitely flexible."

The users of the system don't necessarily need to be exposed to all of their complexity all of the time. We really need to reduce the mental load on these people and say, "How can we actually for 90 percent of the things they need to do during the day, make it as simple as possible while still letting them get into the nuts and bolts when they need to?"

**David Linthicum:**

Yeah. I think it's an absolute problem that we need to solve. I mean, one of the things I kind of discovered in doing my previous research in multi-cloud and how people were leveraging deployment, they're not necessarily running into walls where the technology doesn't work, but they're running into the

complexity challenges and building systems that're so complex that they can't operationalize them. You have to remember there's a limited amount of money that's available to move these things into an operational state and run them longer term.

And if we're going to deploy these things, they're going to bring many different complexity points. We can't hire additional skills. We can't, in essence, double the operations budget. We have to figure out how to make this work with a reasonable amount of money. And I think most of the folks out there who work for rank-and-file companies that're just trying to solve problems with technology, are jumping up and down around this and agreeing with us right now. So, the ability to not only simplify things that're very powerful, but do so without reducing their power—and I think that's maybe something you guys are looking and doing with Portainer.

And I think that's absolutely something that everybody needs to do. People that're in the technology business need to understand there has to be realistic expectations that we can operate these things with less, or the same, costs that we're doing now with some of the on-premise systems. And that seems to be escaping everybody. Everything is moving into very complex states, and the reality is I get it. I'm a technologist, and my propeller spins fast just like everybody else's on my head, but we've got to figure out how to make this stuff work within the context of reasonable amount of budgets. And since I became a consultant years ago, I kind of am very empathetic for these folks within these organizations trying to make these things work. And I think that taking a pragmatic look at leveraging this stuff, making simple things simple, is something that we're missing, I think, as an industry. What're your thoughts on that? I bet you have thoughts on that.

**Neil Cresswell:**

Believe me now, I do. One of the things that really irks me—we know as the CEO of this cloud provider, one of the worst things that happens in life is an outage. And everyone has outages, let's be honest, and at 3:00 in the morning, when you cannot get hold of the consultant, or the expert that designed this particular piece of your cloud service that is now failing, you will regret signing off on such a complicated, perfect solution. And almost always when you are—the next day or the next week when you're doing the RCA to say, "Okay, what went wrong? What did we learn from this? What're we going to do?" almost always the answer is we're going to take away this huge piece of beautifully-made solution that was overly complicated and go back to something that far more people in my team can actually triage and fault-find, because at 3:00 in the morning, and the expert is not answering their phone, or able to be reached for whatever reason, you genuinely value simplicity.

And all throughout Docker, Kubernetes top technologies, it requires an expert to design and build it. When it fails, not if—when it fails, it needs an expert to triage and restore service. And that is just a sad state of affairs. So, again, our whole job is to take away the complexity, take away the operational—the Day-Two burden, make it easier for people—for everyone inside IT that uses the system to know what's going on and give actionable insights to say, "This has failed. Go and do this," as opposed to giving you a barrage of metrics and saying, "You need to go figure out what's going wrong. Here's all the metrics. It's up to you, Mr. SRE, to go and figure out what the heck's going on." We're like, "No, no, no. That's crazy. We need to give you genuine help to say this is what's failed. Go and do this to fix the problem."

**David Linthicum:**

So, what's a typical problem that you're trying to solve with your technology and a typical customer that's going to leverage your technology as the first multiplier to solve their problems?

**Neil Cresswell:**

There's those people who are just getting started with container technology, which almost always translates to Docker on their machine. So, they've come from either a virtual-machine world, or they're coders who have lived their life inside a coding IDE, and now need to make it available inside containers. And they've managed to make the transition to Docker on their machine, and that is basically—it is a journey. It's a journey of education, but it's relatively constrained. You're not trying to deal with scale out and persistence and advanced networking. It's just on your machine.

And that is basically the level of skill that we say is the entry point into Portainer. So, as long as you have started the journey and have mentally understood Docker on your machine, then we want you to be able to use Portainer and deploy against production, against Kubernetes, load balancers, ingress controllers, reverse proxies, persistence. All of that stuff we'll take care of. As long as you know how to use Docker on your machine, we'll take care of the rest for you. So, one of the huge use cases is helping people go from, "I know Docker," to, "I can deploy anywhere."

The second one is helping people in a getopts world. Getopts and automation in general is seen as the key to unlock mainstream adoption, but that tech itself is also quite complicated and requires a real change in behavior to say, "I now control my entire environment from one place." From conflict files, I don't now touch the end systems. The end systems read their state. So, we're trying to make that whole automation easier as well. In the end, it's all around identity and access management. Docker and Kubernetes out of the box doesn't actually do anything to help you enforce identity-and-access management. You need to configure that. If you don't know how to configure it correctly, it's dangerous.

There's been so many well-documented exploits, where there's bitcoin miners running on exposed Kubernetes clusters, because people haven't correctly secured them, and that's scary. That is genuinely scary, so we're saying let us take care of that for you. Let us make sure your cluster is correctly configured, authenticated, secured, locked down so that only authorized users can use it. So, they're kind of the three main drivers, where we're trying to help people with getting started, getting started with automation, and making sure the thing is actually secured and governed correctly.

**David Linthicum:**

So, what industries are more interested in technologies from Protainer.io? Is it going to be finance? Is it going to be retail, health care?

**Neil Cresswell:**

Interestingly enough, it is a monumentally broad-spectrum. We've got users everywhere, from manufacturing, so there's Portainers sitting out there on the shop floor running robots. We've got Portainer inside banks, insurance. We've got just normal enterprises, home labs, NAS hardware. I don't know if you've bought yourself a recent NAS box, but almost always now it'll come with Docker installed and likely Portainer preconfigured. So, it's everywhere, but that's

our open-source community edition, which is designed to have broad appeal. Our commercial products, because the commercial variant of Portainer is focused very tightly on governance and security, that naturally is attractive to those organizations that are in regulated environments, so finance, government, and large enterprise.

**David Linthicum:**

So, I'm interested in the selling process, when you take this thing—not necessarily doing a sales pitch, but I'm talking about how you present this to the market in general. So, do you do so through more thought-leadership marketing, introductory marketing, relationship marketing, all of the above? What's your most successful way to get people aware of your technology and leverage your technology?

**Neil Cresswell:**

So, we're what's known as a commercial open-source company. So, we use our open-source products as our marketing means. So, we say, "Here is the open-source product that does 90 percent of what our paid version does. So, use this, but here are its limits. It has natural boundaries." As a CIO, there's no way a CIO could sign off Portainer CE as being compliant to corporate policy, because it's missing out a whole bunch of stuff around governance/security/audit, so that the goal is to get Portainer in, get it used by teams, loved by teams, and then naturally it gravitates and moves up the food chain and says, "Okay, now, actually, this thing is actually very, very useful to our team. It helps us with efficiency. It helps us better secure and govern the platform. Can we now expand it company-wide?" at which point the CIO says, "No problem. Tell me how it does these 17 things around governance. Oh, it doesn't? Okay, but here's a paid version that does." So, it's the same product. You simply upgrade to the paid version, and you can now tick all the boxes that the CIO and the CISO are demanding.

**David Linthicum:**

Yeah. I love the way open-source, in essence, sells itself, because you're giving away a version of the product that can do most of what the product does, and therefore they're able to test-drive the product and also see what the product does. There's no secrets. You can look at the code. You can look at the architecture, look at the way it's done. It's published out there. Other ecosystems support it, so it's not just customers out there, but you have an ecosystem, the developers that may be building existing things on it and selling on—not just your product but other open-source products. It seems like this is kind of the wave of the future, and I think it has been for some time. And it certainly is affecting how we consume cloud, too. Do you think this will become the majority of the models out there in how people deploy technology? Do you think this is something that will be a little problematic to people who're looking to drive pure profit and a proprietary approach?

**Neil Cresswell:**

Making money from open-source is definitely not a walk in the park, so it's definitely not for everybody. You have to invest in the open-source products. You have to build it, pay for people to build it to make it available for free. Then you have to support the people who are using it for free, for free. So, not only are you saying, "Here's my free product, but here is access to my support and engineering team for free as well." So, it's quite an investment in this open-source product, which is basically an investment in marketing, which will eventually turn into sales in a commercial entity. Now, that probably takes a bit longer, arguably, than a proprietary product or a SaaS offering, but it is completely viable, because it's so open and transparent. And there is so much positive returns that you get from the open-source community, more than just money.

It's amazing. We've got over 500,000 active monthly users of our free products. That's 500,000 people giving me feedback every single day on how I can make their experience better. That feeds back into the product's development, which in turn makes the commercial variant even better as well. So, if people are confused using the UI and they're opening support tickets, we don't just say, okay—we don't teach them how to use it. We say, "Well, hang on. We've actually made a mistake here. We need to make this better." So, we use all these inputs to build a better product. So, that's actually monumentally valuable. So, we may not get money from them directly, but we get something better than money, which is direct input to help us build a better product.

**David Linthicum:**

Yeah, someone who's paid for those services, the ability to kind of get an advisory board in place and have people test the products—and if it's a proprietary product and they can get it for free, there's no incentive for them to do it. And, so, in essence, you have to engage these firms to go off and do that. And I had to pay for that as a CTO or someone who is selling proprietary software at the time. Yeah, but you do with this open-source community. You get the immediate feedback, and therefore it's almost like a built-in product-management team that's able to, by committee, tell you where you're going right and where you're going wrong with your product, how to steer it in better directions, quality issues you're able to get to before you would deploy them out with the next version and the ability to do this all on demand.

So, everything's available, so you're able to get to the goodness of whatever the latest version is, as they need it. And, so, that seems to be nirvana moving forward, the ability to kind of get to this point where you're dealing with an ecosystem of individuals, a bunch of folks who're able to influence the marketplace. And you don't have to pay them. They are, in essence, leveraging your product, because it's within their best interest for their own career aspirations or what they're trying to do with their business to leverage your product. So, it's a symbiotic relationship we're kind of moving to right now. Would you agree?

**Neil Cresswell:**

Absolutely, yeah. I'm amazed every day when we get inbound e-mails from some of our users, saying, "Hey, my Dev team used Portainer internally to help deploy applications for my own company. They've asked whether we can actually give back an hour of their time a month to help build a better Portainer. Can you guys give us some features that they can work on in their spare time?" They really feel like they owe a debt of gratitude to Portainer for making their lives easier, and the way that they can pay back that gratitude is by helping build a better product through feature development. So, it's amazing how symbiotic the whole relationship is between us as creators of the product and the users of the product, wanting to help co-create it. So, it's actually a beautiful thing.

**David Linthicum:**

Yeah, it's a very beautiful thing, especially if that symbiotic relationship is going to lead to everybody succeeding, win-win. So, what's the process of implementing your technology? Am I downloading things? It leverages a service off of a site? What do I need to support it in terms of platforms—some of the details, not all the details, but what basically are the requirements for me to get into the Portainer.io business?

**Neil Cresswell:**

So, it's not a SaaS service. This is all self-hosted. So, you run it in your environment. As you would rightfully expect, because we are a tool to help manage container platforms, we ourselves ship as a container. So, you simply Docker run Portainer somewhere in your environment, which becomes the central Portainer instance, and that manages all of your other environments. Or you can use Kubernetes and deploy it inside a Kub cluster and then manage all your environments. So, it takes mere seconds to get Portainer up and running in your environment, and it's completely within your firewall, controlled by you, governed by you. All the data is inside your network. There's no SaaS offering. So, it's very safe and secure.

**David Linthicum:**

So, where can listeners go find out more about Portainer.io? I guess that's your website as well. And where would you point them in terms of descriptions of the product, where they can get access to the product right now and start getting into the Portainer.io business?

**Neil Cresswell:**

Yeah, so definitely go to the website, Portainer.io. There's a lot of information on the website. We have docs.Portainer.io, which is a very, very comprehensive and recently rewritten catalogue of admin guides and user guides. But also we've been investing very heavily in creating content. We have a whole content team now, who are creating how-to guides, instructional videos for how to use the product and get maximum benefit from it. So, we have our own YouTube channel as well, and so on there, there is an array of content right across the spectrum of what the product can do, so definitely go there and join up to our community if you want us to send you monthly newsletters on all of the cool stuff that we're doing, the new features we've bought, and access and intros and invites to webinars and training sessions.

**David Linthicum:**

Yeah, I know from listener feedback that the majority of folks who listen to this podcast are moving into containers fast and furious, have done it over the last several years and are moving into it as a net-new system. So, right now is the time to figure out how you can leverage the best enabling technology to get to these solutions and get to the value that you need without overcomplicating things. I think that has a tendency to be a direction that we go as a technology team or as technology advocates, and that won't scale. So, we have to think differently in terms of how we leverage technology and how we make this stuff work.

So, anyway, if you enjoyed this podcast, make sure to like and subscribe on iTunes, wherever you get your podcasts. Also, don't forget to rate us. Also, check out our past episodes, including the On Cloud podcast, hosted by my good friend Mike Kavis and his show, Architecting the Cloud. And if you'd like to learn more about Deloitte's cloud capabilities, check out deloittecloudpodcast, all one word, dot-com. And if you'd like to contact me directly, you can reach me at DLinthicum, L-I-N-T-H-I-C-U-M @deloitte.com. So until next time, best of luck with your cloud projects. We'll talk again real soon. You guys take good care and be safe.

**Operator:**

This podcast is produced by Deloitte. The views and opinions expressed by podcast speakers and guests are solely their own and do not reflect the opinions of Deloitte. This podcast provides general information only and is not intended to constitute advice or services of any kind. For additional information about Deloitte, go to Deloitte.com/about.

Visit the On Cloud library  
[www.deloitte.com/us/cloud-podcast](http://www.deloitte.com/us/cloud-podcast)

About Deloitte

-----  
As used in this podcast, "Deloitte" means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see [www.deloitte.com/us/about](http://www.deloitte.com/us/about) for a detailed description of our legal structure. Certain services may not be available to attest clients under the rules and regulations of public accounting. Please see [www.deloitte.com/about](http://www.deloitte.com/about) to learn more about our global network of member firms. Copyright © 2021 Deloitte Development LLC. All rights reserved.