Simon Elisha (00:02.29)
Hello everyone and welcome back to the AWS podcast. Simon Leige here with you. Great to have you back. I'm joined by a very special guest. I'm joined by my good mate, Brett Looney, who is an AWS solution architect and all -round nice guy. G'day Brett, welcome to the podcast.

Brett (00:14.35)
Aw, thanks mate, I appreciate that.

Simon Elisha (00:16.72)
Now Brett and I do go way back, although Brett is located on the exact opposite side of Australia that I'm at. So he's from a little town we call Perth in Western Australia.

Brett (00:26.57)
It is definitely a little town for sure.

Simon Elisha (00:30.41)
And Western Australia, for those of you who don't understand Australian nuances, often wants to secede from the rest of Australia thinking they're much bigger and better than us, but only when there's a mining boom. So, you know, it depends.

Brett (00:42.92)
Exactly, exactly. The rest of the time we actually need the culture that comes from Melbourne.

Simon Elisha (00:48.91)
Absolutely. Absolutely. Now, Brett is not just here because of his winning personality, but also because he probably knows more about networking than most people do and has probably forgotten more than I've ever learned. And we wanted to do a deep dive, super deep dive on a topic that comes up quite a lot for a lot of our customers, which is how to manage internet ingress and egress patterns when you're using AWS. Now,

Brett (00:54.57)
you

Brett (01:16.43)
for sure.

Simon Elisha (01:17.49)
This is kind of a big deal because for a lot of customers that have a strong mix of workloads, some of them are publicly facing and some of them are internally focused. But in most cases, all of them need some kind of access to the internet. And I guess Brett, what we want us to do was to demystify this. But before we do that, tell us a bit about yourself, your credentials and why this is a topic that you've really spent a lot of time talking and thinking about.

Brett (01:43.82)

Yeah, so I've spent a lot of time in the networking world back in the 90s. Yes, I really am that old. Did a lot of stuff with Cisco and have worked lots with customers on premises. And then coming into AWS, of course, have continued being the networking nerd. And of course, as you say, a lot of the patterns that we have with customers are how do I connect my apps to the internet or how do I deliver content to the internet in a secure way?

This happens for new applications, also happens for customers who are migrating to AWS and they want to have the same level of security and assurance that they feel that they have on premises. And luckily we have a lot of tools to do that. But there are different ways of doing it because naturally the cloud can be better and it can be different. And that's where this conversation comes from.

Simon Elisha (02:38.09)
Absolutely. We're dealing with a, I guess a different set of constraints and some unconstrained thinking that if you apply previous mental models to the situation is not going to work as well. So let's maybe start at the start, which are, what are some of the common patterns in terms of ingress and egress?

Brett (02:56.65)
That's really easy. There's two types. We have shared and distributed. So country and Western, we have both types, right? Absolutely. There are probably some people who aren't old enough to get that joke, but that's fine. So yes, so you can do shared egress or distributed egress. And similarly, you can do shared ingress and distributed ingress. And this is where most of the conversations come from. Customers come to us and say, well, which one is best?

Simon Elisha (02:59.82)
Country and wisdom.

Simon Elisha (03:08.62)
Ha ha ha.

Brett (03:25.77)
And of course, as always, as you know, being a solutions architect from many years ago, it depends.

Simon Elisha (03:31.72)
think if I ever got a tattoo, that's what it would say. It depends. So what are you seeing most customers using what tends to work out there?

Brett (03:34.22)
For sure.

Brett (03:41.61)

So let's cover egress first because it's the easy one. Shared egress is what most customers do, as in we have a single point where we send all of our traffic out to the internet. It turns out in AWS that's really easy to do. It's very similar to what customers did on premises, so it's pretty easy conceptually to understand. We have tools such as AWS Network Firewall and Gateway Load Balancer that make it easy for customers to funnel their traffic through.

that single egress point, inspect it if they need to, and then send it out to the internet, receive the responses, and away you go. So that one is actually pretty easy.

Simon Elisha (04:18.57)
So let's dive a little bit more into the why though. Why do I want to inspect stuff going out? I mean, stuff coming in, I can understand, but going out, why would I care about that?

Brett (04:29.35)
You might want to stop customers going to malicious websites. You might want to be doing data loss prevention, otherwise known as DLP, to make sure that there's not sensitive information leaving your organization. But it's also a way for you to check and make sure that there's nothing compromised inside your network. So you can use your firewalls to detect whether something is infected with a virus and trying to reach its command and control centers, or whether there's something doing Bitcoin mining, for example, and all those sorts of things.

So your ability to see what's going out is still really critical to maintain a strong security posture, because otherwise you don't have visibility of what's happening inside your network.

Simon Elisha (05:09.80)
It gives you some interesting signals. Paint for us a word picture of what this would look like in a sort of a VPC type structure. Are we talking about one shared VPC? Are we talking about transit gateways? Like conceptually, what are we thinking about if I'm architecting in my brain? How do the magical squares connect to each other?

Brett (05:19.88)
Mm -hmm. Yep.

Brett (05:31.15)
No, absolutely. So most large customers are going to be running Transit Gateway. Transit Gateway is, for all intents and purposes, a very large cloud -scale router. Under the hood, it's actually far more scalable and resilient than just being a single box. And if you're interested, go out and Google Hyperplane, which is the underlying service that runs Transit Gateway. It's pretty amazing. And so all of customers VPCs are generally connected to their Transit Gateway.

and they have a route that says, hey, send the traffic to this single egress VPC where all the traffic is going to go out. In that VPC, at its simplest, if you didn't want to do any filtering at all, you would have a NAT gateway and an internet gateway and just send all the traffic out. But

because we're talking about security, the piece we put in front of that, so in between where the traffic comes from the transit gateway into the VPC and the internet gateway, we put a

a gateway load balancer endpoint. So gateway load balancer is a service that came about in about 2018, if I remember correctly, and it has the ability to basically intercept traffic and send it off to a set of firewalls. And those firewalls can be run by you. They can be any kind of third party firewalls. I think Palo Alto, Cisco, FortiGate, there's lots of partners out there that do this. They can be firewalls run by those partners. So again,

Halo Alto have a cloud hosted firewall, managed service, absolutely. And AWS also has a service called AWS network firewall, which is essentially the same thing. Under the hood, you will actually see a gateway load balancer endpoint in your VPC. But again, we are running the firewall on your behalf. So what type of firewall you put there is reasonably unimportant as long as you are comfortable with its ability to do what you need it to do.

Simon Elisha (07:01.80)
like a managed service type situation. Is that? Yeah.

Brett (07:27.63)
and it is transparent. So you route the traffic basically through the firewalls and it decides what it's going to do with it. So yeah, so that pattern is workload VPC to transit gateway to egress VPC via the gateway load balance.

Simon Elisha (07:44.39)
And the choice of firewalls, I think is an interesting one because, you know, on the face of it, you may say, well, just use the AWS network firewall. Why wouldn't you? And one of the reasons is that often organizations have huge investments in firewall providers or those firewall providers provide particular capabilities that are very important to them. So this choice component is actually important, which I guess feeds into the it depends concept here.

Brett (07:47.63)
Mm -hmm.

Brett (08:09.87)
Absolutely. And I see customers sometimes run a mix of firewalls. So you're absolutely right. They might come from on -premises where they have, you know, firewall vendor X and they want to continue using that because that's where their expertise is and it offers them features that they are comfortable with and they use. But one of the interesting things about AWS is we give customers great power here. You can, in fact, use this same pattern to inspect all of the traffic in your organization. So all of the traffic going between

all of your VPCs. And a lot of customers start this way, but it ends up being very, very expensive because if you are looking at every single byte and every packet that goes around in your network, those firewalls have to be large in order to do that. And what some customers do is

they say, actually, you know what, we're going to run our third party firewalls for the risky stuff where it's connected to the internet, but we're going to run AWS network firewall for that, what we call East West inspection between VPCs.

because it's actually less expensive and we're a little bit less concerned about, you know, advanced persistent threats and things like that. But we still do want to do some level of filtering. So it comes down to comfort versus experience versus cost in the end.

Simon Elisha (09:23.37)
And there's also, I guess, one of the other elements to think about is capacity and capacity planning. I know in the old days when we would just literally set up a firewall running on an EC2 instance and we knew the different sizes of EC2 instances and how much network you could get through them, etc. This is a lot different to that, but we're still talking bump in the wire. So you've got to make sure that the devices that the traffic is flowing through can cope with the

Brett (09:29.32)
Mm -hmm.

Simon Elisha (09:52.20)
vagaries of traffic in terms of that egress.

Brett (09:55.43)
Absolutely. And the great thing here, and this is what is very different to operating on premises, is that in the cloud, in AWS, you have scalability. And again, Gateway Load Balancer can point to a set of firewalls that are attached to an auto -scaling group. So they can scale up in order to meet those traffic requirements. And more importantly, they can scale back down again and save money, right? Because that's what it comes down to. And that's amazing. And you can obviously say, well, actually, I need you to scale up preemptively.

at eight o 'clock in the morning before people are coming into work, and then just allow it to scale down organically at the end of the day. And this is great because you don't need your firewalls scaled for peak all the time, it saves you money. And it turns out there's an added benefit here because you're constantly rotating that fleet of firewalls, it actually has a side effect of making upgrades really, really easy because you can say, well, we're currently using version A for our firewalls.

But version B is coming along, so the next time we launch a firewall, we'll just launch version B. And before you know it, all the version A's have gone away. And this makes it.

Simon Elisha (10:58.44)
You kind of got a built in rolling upgrade going on there.

Brett (11:01.42)

Absolutely, absolutely. So it's very powerful. And it's one of those things that is, and I'm not being unkind to people, it's difficult for people to get their heads around because they're used to these firewalls being static devices and their replacement being high risk. And it isn't in this one.

Simon Elisha (11:15.00)
Yeah. Yes. It's not a box or the box or the cluster. It's a whole different concept. And I wanna continue on the performance line because it is important. One of the biggest challenges I see with customers or complaints or problems they have is when the user community are trying to access the internet to do their jobs.

Brett (11:21.93)
Mmm. Mmm.

Simon Elisha (11:37.64)
and it's slow because it's getting throttled through that egress point. So your point about the scalability of the egress is really important to reduce calls on the help desk and to have happy employees.

Brett (11:49.45)
Absolutely. And I've worked with customers previously, again, in the on premises world where they have these whole stack of firewall devices and they have a gigabit internet link and they're only able to push 100 megabits because all the devices are overloaded. And the cost of upgrading that stack and the risk in actually going in and touching it is really, really high. So those things get stalled for years and years and years. So I'd much rather be able to say, you know what, we just scale up the firewalls. Thanks very much. We're done.

Simon Elisha (12:16.20)
We like it. Now let's flip the script. We talked a bit about egress. Let's talk about the other way. Ingress. What do you see customers doing for this one?

Brett (12:18.99)
Yep.

Brett (12:22.38)
Mm -hmm.

So this is where I see a lot of disagreement, a lot of discussion, and it's where I spend a lot of time with customers. Again, you can have shared and distributed ingress. Shared ingress is what is most comfortable for customers because it's what we've always done on premises. You have a single, maybe two internet links coming into your organization if you're big, and you have this big...

conglomerate of firewalls that sit there and do all of the filtering and it's what people understand. It's a choke point, but deliberate. Now in AWS, you don't have that anymore. If you've got 20 VPCs, they can all be connected to the internet. You've got this distributed point and that makes people really uncomfortable. So they tend to, customers tend to go, we want to do the same thing we did on -prem because we understand it. And that's completely normal. That's what everyone does.

Simon Elisha (13:16.90)
It's a reasonable first start to say, well, I know and understand this, I can replicate the model. Let me do that first, you know.

Brett (13:23.37)
Right? Exactly. But there's a problem here, right? And so think of an application that is having a good day, as in it goes viral and all of a sudden your requests go from 1 ,000 requests an hour to a million requests a minute. Your firewalls are now that choke point. And if they don't scale quickly enough, then, in fact, this is true of any network device. It's not just the firewalls. But if your network devices don't scale fast enough,

Simon Elisha (13:38.56)
Mm.

Brett (13:50.51)
then all of a sudden every application coming through there is now also having a bad day. And that could happen for a DDoS as well, so a distributed denial of service attack. If one of your applications is having a bad day, then everyone is having a bad day. And sure, we just talked about firewalls scaling using Gateway Load Balancer, which you can absolutely do. But at the same time, they may not scale fast enough to meet that sort of thing. And you don't want to be knocked offline because of that.

Simon Elisha (13:58.18)
That's very true.

Brett (14:18.76)
And again, indeed taking a slight detour out of this story. A couple of years ago, we, AWS, had what we would call a large scale event in US East one, so in a North Virginia region, where we did in fact have a bunch of network devices that were overloaded. And that essentially took out the control plane for our customers. Now, all of the existing

Services kept running, but you couldn't make any changes. And this happened because it was a cascading failure. The devices get overloaded. So whether they're firewalls or routers or anything like that, they get overloaded. And the end systems that are trying to communicate across those devices, what do they do? They see packet loss, so they retransmit. And that makes the session worse. Every, sorry, it makes the problem worse. All the sessions get

dropped. We do retries. And the only thing you can do is start to shut down those end systems. And it was, it's a big problem to solve.

So we've gone out and said, you know what, we've actually moved away from a shared model where all of our services go through common network devices, so firewalls, routers, things like that, to a distributed model where we actually use AWS Network Firewall to do the filtering for all of those control plane activities. And every single service now scales independently of each other. So this is...

Simon Elisha (15:40.48)
And that's a critical importance because then you're talking about things like blast radius and what gets affected by what. And I think the interesting thing is firstly, that's a great example of there's no compression algorithm for experience and usually the experience is hard won. But also the fact that one of the reasons why we went traditionally on premises for a common choke point was it made it easy to manage.

Brett (15:43.66)
Mm -hmm. Yes.

Brett (15:53.55)
Yes.

Simon Elisha (16:04.77)
particularly if you were in a click ops world where you had a console and an expert and that's how you manage stuff. So you want less stuff to manage. Whereas in an infrastructure as code world, you can scale as big as you want, AKA AWS scale. It's the same amount of effort. It's more predictable and you're reducing that blast radius as well.

Brett (16:05.10)
Correct.

Brett (16:27.21)
Absolutely right. Again, it's not just about an application having a good or bad day. If somebody accidentally makes a change, you know, the good click ops and you've got this big shared firewall, they can affect everybody. And that's not to say anything about malicious stuff happening as well. If you have a failure, it affects everybody. So yeah, and the biggest pushback I get, and you're absolutely spot on, is where customers are saying, well, actually, we can't manage all that. If we go from having one ingress point to having 10 ingress points for all of our applications.

How do we manage all that? And I think one of the things we don't talk about enough at AWS as the technical focused architects is automation. The power of being in AWS is that everything can be automated. Everything has an API and you should absolutely do that. And so if you're using the AWS native tools, things like network firewall, if you're using security groups and

things like that, we have Firewall Manager, which is there to help you manage all of those things across all of those accounts.

And so you can say, well, I do have 10 ingress points, but I can update the policy for all of them really, really quickly just by changing one policy and then doing a deploy on top of that. And I really encourage customers to think heavily about network automation because, and being blunt as network people, automation is not something that we think about. We're like, we love getting into consoles and tapping away in topographic commands and stuff like that. We love getting in and clicking and changing rules and firewalls.

Simon Elisha (17:46.46)
No.

Simon Elisha (17:52.38)
And also, also we're very used to, you know, build it once, build it right, get it done. And then it sits there for a few years, you know, a few patching along the way, but you know, your fundamental routing rules, et cetera, pretty static, you know, quote unquote, good environment.

Brett (17:57.55)
Mm -hmm.

Brett (18:04.59)
Yes. Yeah, absolutely. So it's super powerful to be able to automate stuff. But I even encourage customers to go further than that. And that is, again, the common viewpoint is, well, we bring stuff in, and the very first thing that hits is the firewall. And I would argue that, again, firewalls are really expensive devices. And they don't scale quickly, and they don't scale easily. And they do, but it takes time to do that.

And to customers I'm saying, well, why don't we use the cloud native tools that we already have in order to blunt any of those bad things that might be happening to applications? So using things like CloudFront as a content distribution network, which automatically has DDoS protection built in, use standard WAF rules in some web application firewall rules in CloudFront to do that. Then deliver the traffic to your load balances where you can have additional filters and rules and WAF and things like that to make sure that

Only the traffic which is really, to use a word, cleaner, I guess, filtered, actually gets through to your workloads. And only at that point do you bring your firewalls in, because the firewalls are there to look at things like advanced persistent threats. They're there to look at behavioral type of attacks and things like that. Why use your firewalls to go, yeah, I just want to block traffic that's not coming in on port 80 or port 443. I want to block traffic which is coming from bots. Why not just do that at the layer where it's

done in a cloud scale way where we scale automatically for you and you only pay for what you use. And then use your firewalls for all the stuff that is really, really important. And that way you actually pay less for your firewalls because they're handling less traffic. They can scale easier and it's generally a less expensive but more scalable solution for you.

Simon Elisha (19:51.58)
Exactly. And you get it. You also giving your customers a better, a better experience as well, because they're accessing your service through those cloud front points of presence around the globe. They're getting faster, faster response times. It's a, there's a, there's a lot of benefit beyond just the defensive side as well. There's also sort of a marketing thing. And I don't know about you, Brett, but I still notice when, you know, if I go to someone's website and it snaps up really quick, it's like, wow, okay. Someone's, someone's thought about this versus the one that sort of laboriously paints.

Brett (19:57.19)
Mm -hmm.

Brett (20:15.98)
Yeah.

Brett (20:22.47)
No, absolutely. And again, you mentioned earlier that I'm from Perth. One of the things that we're completely aware of here is that we are a long way from anywhere. And network latency actually matters. So reducing latency using CloudFront and things like that is absolutely key. And you really do notice it when websites are slow and when they are fast and responsive.

Simon Elisha (20:44.32)
So if we're talking, you know, CloudFront, ALB, WAF, and I agree, I think if you're doing anything publicly facing, if you're not using CloudFront, you're doing it wrong. You'd want to have really good reasons not to be using that. But if that's the way, quote unquote, the way, what's the downside of this distributed approach?

Brett (21:03.24)
So it can be more costly, right? Because you're now having multiple places where you need to insert firewalls rather than just one place. One of the great things about Gateway Load Balancer is that you can say, well, I'm going to put endpoints to inspect my traffic in multiple places, but use a common core of firewalls to inspect it. But...

that ends up getting you back to the same challenge where you have this set of firewalls that have to scale in response to all of your applications at once rather than just the one application. So it is probably slightly higher cost, but that is offset by the fact that as you say, everyone's going to get a better experience and those applications are going to be on the whole in the long -term more reliable because less things are going to happen to them based around the other applications around.

Simon Elisha (21:53.92)
Exactly. There's definitely a resilience aspect there too. So what, what are some of the mistakes you see people making? It's, it's, it's easier to, you know, and again, I'm using the words right and wrong very loosely here, and I'm not using mistake in the pejorative sense. I'm just saying, you know, maybe suboptimal things that you see people do you say, well, if you did this, things could be better.

Brett (21:57.93)
Yeah.

Brett (22:05.93)
Yes.

Brett (22:14.63)
Absolutely. So I see customers doing what they did on -prem as in they put their firewalls as that front line guard against attack and against traffic coming in. And what happens then is they end up in a position where they have to be scaled for peak. Those firewalls have to handle all the traffic. Upgrades are really, really hard. Resilience becomes a problem. So swapping traffic from one firewall to another is very, very difficult at that point.

And all of these things lead to operational delays. It leads to, because we've got these shared firewalls and some of them have to make a change, all of a sudden we've got to go to a change board. We can't, you know, it takes two or three weeks to get things approved. And it affects agility, not just for the team managing the firewalls, but also the application teams as well. And on top of that, all of those things really do cost money and they cost time. So it's...

Simon Elisha (23:10.11)
Yeah, money is a real thing. But also risk is a real thing. So it's not just the money, but it's how you managing the risk of the threats you're facing too.

Brett (23:13.13)
Yeah. Yes.

Brett (23:20.65)
Absolutely. And the thing is that I often, again, talk to customers about this topic and I'm saying, look, I really would encourage you to go to a distributed ingress or even, and there's a middle ground there where you can kind of distribute some things about others and have those shared firewalls if you want to. But often it's a step too far. And so customers are going, we're going to go with what we're familiar with. Even though they might say, yeah, we can see the benefit of that, but we are not ready for that yet.

Simon Elisha (23:47.48)

Hmm. Hmm. But I guess they've got to have a trajectory towards what better looks like. And then you sort of, you touched on that middle ground. Is there a, you know, at the moment we're sort of talking about terms in terms of, you know, soft or crunchy tacos. is, is there a middle ground of sort of a semi soft or semi crunchy?

Brett (24:00.78)
Hehehehe

Yeah, actually there is. And I've been working on a demo for this as well. So basically what happens is if you think about the standard team setup in any organization, you would normally have the team that takes care of the application, but then you have the team that takes care of the network and probably that ingress in from the internet as well. And so again, we give you great power in AWS to be very, very flexible with things. And so what we can do is have the application team running in their own account.

they can have their own isolated VPC where their application is running. And then what happens is we have a shared ingress VPC where for each application we set up a CloudFront distribution, a load balancer, and WAF, and even a firewall that all come into that one central VPC where you have your networking team taking care of things.

And then we deliver the traffic from that Ingress VPC to the Workload VPC via PrivateLink. Now that's a service I've mentioned up until now. PrivateLink, ta -da, right, good. So PrivateLink was actually developed for service providers, as in I'm a software as a service provider and I want to offer you a connection to my service. And it does a couple of things that are really amazing. First of all, it only allows you to connect to me.

Simon Elisha (25:10.68)
new service, new service alert.

Brett (25:30.25)
I cannot connect back to you. So as a service provider, this is great. It protects you from me. I cannot connect back to anything on your side. You just get a basically a network interface that you talk to my service on, which is great. And the second thing it does is it actually fixes an IP overlap problem. So you can have the same IP address as me and we can still talk. It does this fancy double -sided NAT thing in the middle, which is really cool.

And so by using PrivateLink in a different way for this, I can set up a PrivateLink endpoint between the application and the Ingress VPC, which means that traffic can come from the internet through CloudFront, ALB, Firewalls, lots of stuff. And it hits PrivateLink and is magically transported across to the application VPC where the application team can operate completely independently. They can do whatever they like in there. They run up EC2 instances, they can run up containers, whatever they like to host their application.

But what they can't do is connect back out to the internet via that ingress path. And so now we have this nicely isolated VPC, which you can't really do anything at all, but it can receive traffic from the outside world that is appropriately filtered. And so what we end up with is inside that ingress VPC is effectively an application stack per application that can scale

independently of the other applications. So because you've got CloudFront ALB private link dedicated to that app. But it's all in one VPC where it can be managed by a single team.

Simon Elisha (27:07.35)
Yeah, so you've still got network control, but you've also got that classic DevOps application control as well. And the other thing to point out about PrivateLink is traffic for PrivateLink only traverses the AWS network. So that's the other magical part of it is that the data does not go across any internet network. It only goes across the AWS network.

Brett (27:19.98)
Absolutely.

Brett (27:29.29)
Definitely. Private Link is really cool. And again, I mentioned Hyperplane before. Private Link is another Hyperplane service, which means it is highly scalable out to tens or hundreds of gigabits of traffic per second. And also is what we call a cell -based architecture, which means it is cell, which is dedicated to a customer, scales independently of every other customer.

Simon Elisha (27:53.27)
That's another way of managing risk. Now, Brett, you talk to lots of customers, not just in Perth, which is of course the most remote capital city in the most remote country in the world, just to help folks understand the latency challenges that one of our network experts deals with on a daily basis. But you obviously speak to global customers as well. And one of the things I love about our role and talking to customers and particularly technology customers is they're not shy on providing opinions and feedback.

Brett (27:55.31)
Mm -hmm.

Brett (28:03.72)
Hahaha.

Simon Elisha (28:20.02)
When you talk to customers about this, what's some of the feedback you get? What are typically the talk tracks that start to happen?

Brett (28:20.04)
Mm -hmm.

Brett (28:29.56)
On the whole, when I talk about the shared to semi -distributed to fully distributed ingress, the feedback is, yeah, we can absolutely see the advantage of that. Customers see the sense in it. It makes sense. They go, okay, we get it. We get scalability and we get agility. We get this ability to completely deliver a unique experience for each of our application users in terms of the people building it and the end users who are using it.

But it comes down to what I said earlier, it sometimes is a step too far. And customers like Worko, we need to work towards this. And I've had this conversation with customers two years ago and only now are they like, right, we now have the maturity level internally where our teams who are building the applications are fully on board with DevOps practices and we are building that level of automation into our networking and security teams as well. I think it's great to see customers on that journey.

And the thing that I tend to point customers to, there's a great re -invent talk in 2017 from Eric Bradwine. So he's a principal director of security, but that's not his correct title, but he's a really important dude inside Amazon. Right.

Simon Elisha (29:43.63)
No, he's a dude with a brain the size of a planet and a wall of patents.

Brett (29:51.37)
Yeah. And so he actually has a talk which is called the Amazon philosophy of security. And in that he talks about the security team at Amazon who started out doing everything manually and then realized that they could not scale because there was just too much work for them to do. And so they ended up automating their own jobs and paraphrasing his story is like, we wrote a bunch of bad code that didn't work and then we went back and we made it work. And now

Simon Elisha (30:15.41)
You

Brett (30:18.95)
95 % of the security tickets that are raised on the security team are resolved automatically through the automation so that the humans can go off and do more valuable things. They can go off and actually answer questions with other humans rather than dealing with all the minutiae that you shouldn't have to do.

Simon Elisha (30:33.75)
Yeah, you left an RDP port open to world type stuff.

Brett (30:36.94)
Exactly, exactly. And so that means, you know, if they can do that, then anybody can do that and you can automate away the bits of the job that you shouldn't have to do.

Simon Elisha (30:47.47)
Well, let's talk quickly just in closing again about that mental model shift, but also the skills shift. Let's say I'm a experienced network person or I've just got a few years under my belt. Maybe I've just got my CCIE. I'm really comfortable with the technology of networking. I love to talk about spanning tree protocol, all that sort of stuff. You know, I'm working on memorizing strings of IPv6 addresses.

Brett (31:09.64)
You

Simon Elisha (31:16.47)
What do I do to, I guess, make my skills more cloud suitable and be able to use my networking knowledge, but make it supercharged by what I can do in the cloud?

Brett (31:29.35)
Yeah, it's a good question because I think there are many, many paths here. My own journey, taking the one has been, first of all, get into the AWS console and figure out how those AWS networking components work. We've been talking about VPC and Transit Gateway and Private Link and things like that. Play with them and get them working so that you understand what they do. And then the second step is, okay, how would I automate the build of this? And the first tool...

anybody should use is CloudFormation because you get to say, well, here's a file that describes what I want my environment to look like. It's amazing and it's unbelievably powerful when you start digging down into it. And then after that, it becomes, okay, I've now got this environment. How do I monitor it? How can I take care of it? How can I look at alerts and things like that? So I am not a professional software developer.

Simon Elisha (32:10.42)
Hmm.

Brett (32:25.26)
but I know enough to be really, really dangerous. You should not let me get anywhere near your code. So choose a language that you're comfortable with, whether it's Python or Node or Go or Java, anything like that, and start to write things that actually automate those alerts coming out from threshold alerts that come out of CloudWatch metrics and things like that. Or how do I change this network to be responsive to events? Now, this is great in theory. What I find, again,

Simon Elisha (32:28.88)
Hehehe

Brett (32:54.25)

the best thing to do is find a project to solve problems on and then go and use your knowledge there, which is way better than I'm just going to think up an imaginary scenario. For me, it's easy. Customers come to me all the time and they say, hey, I've got this problem. How do I solve it? And I'm like, let me do a prototype for you. Right. And I get to play with all those things, which is great. And that is an ideal way of testing your knowledge, putting it into practice. And again, as with the Eric Bramwine story,

Simon Elisha (33:11.86)
Yeah, yeah.

Brett (33:23.59)
You start off small and then you work towards something which is bigger. And before you know it, you've got a system which is hopefully running itself.

Simon Elisha (33:32.14)
Exactly, exactly. So how do folks get more information about this if they want to dive a little deeper?

Brett (33:37.61)
If you're listening to this and going like, wow, I want more details, contact your local AWS Solutions Architect. Mention my name and I'm happy to share with them the artifacts that I have about this whole ingress, egress thing and stuff like that. If you want to get more into network automation, again, there's a whole bunch of us inside AWS who are only more than happy to talk to you about that. And so please, again, your local Solutions Architect is a good entry point and then we can...

have a conversation. But there is also a huge amount of resources available in the AWS Samples GitHub repo.

Simon Elisha (34:13.42)
Absolutely. There's also some fantastic reinvent presentations as well. They're all up on YouTube as well. That's a fantastic resource. Even things like, I think there's one, you know, the trip of a billion packets and there's all these different aspects. So you can dive super deep and often it's even more easy to understand with all the animated diagrams and such. But...

Brett (34:17.87)
Yes.

Brett (34:25.58)
Yes.

Simon Elisha (34:36.17)

Brett, we're definitely going to have you back on again to talk about all things networking at even deeper levels in the future. But thanks for coming on the show and demystifying ingress and ingress for us.

Brett (34:40.14)
Awesome.

Brett (34:45.48)
Thank you for having me and letting me espouse my opinions.

Simon Elisha (34:49.23)
Anyway, of course, we do love to get your feedback. aws.amazon.com/podcasts is the place to do it. And until next time, keep on building.