



For Cloud professionals, part of the On Cloud Podcast

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

Episode_03_LoriMacvittie

Duration: 0:23:37

David: Hey, guys welcome back to the podcast and with me I have a special guest Lori Macvittie. I always say MacVittie that's right.

Lori MacVittie: That's close enough yes. I think.

David: You see, it's not McVittie, its MacVittie right?

Lori: It depends on which member of the family you are talking to, some are actually McVitties and some are MacVittie's in the same generation so yes, take your pick.

David: Yes, anytime I mispronounce somebody's last name I say "Hey, pronounce mine buddy" you know go ahead and give it a try. So, Lori introduce yourself, this is a new audience being the Deloitte podcast, so tell us what you do at F5 and really some of the things you are focused on.

Lori MacVittie: Okay, well exciting to be in front of a new audience today, I'm Lori MacVittie, I'm the Principal Technical Evangelist for

F5 Networks and what that means is I have to be familiar with everything that we do in at least as much detail as is required to be able to communicate that, whether it's speaking or PowerPoint or writing blogs and articles just to explain the technology as well as market technology. So, a lot of my focus is on trends, architectures, what's going on in things that F5 doesn't normally look at. Like what's an Application Architecture doing today, because that impacts what we do and how applications are delivered not only from the perspective of what F5 does but just in general how you are going to architecture your network, your data center, the Cloud, what you are doing. So, I spend a lot of time just reading the Internet and trying to turn that into something that makes sense in the written word usually.

David: Yes, Lori is a very prolific blogger, and has some great content out there and I love stealing from her. So ultimately how often you blog?

Lori MacVittie: I am currently on it, at least a once a week schedule.

David: Okay, you are putting out some pretty good content. And so, we're going to talk about one particular blog post she did and this is entitled "Imperative or Declarative? The question of DevOps in the network". So, this is a pretty interesting topic, so give me a summary of the post and kind of tell me the argument you are making here?

Lori MacVittie: The summary is... so there are two different ways to interface with anything whether it's a server or a router or a switch, when you are doing automation. One of those is through an API, which is "Imperative" you are telling it exactly what to do in a step-by-step, write script or program if you go that far. And the other is "Declarative" where you give it kind of a configuration you tell it what you want it to do, but you don't necessarily step-by-step explain what to do and that's declarative. So, there are these two models that we can use to automate the network infrastructure whichever pipeline you are looking at. And the argument is if you use imperative methods, if you do everything via an API you are going to accumulate a lot of technical debt because you are essentially writing code that is binding your automation and your process to a specific API version and specific device in the network. So, we would like to see that move to a more declarative model where you are using more of a configuration approach, a DevOps like infrastructure as code, if you will, to tell it this is what I want the device to look like when I'm done, figure out how to do it, to alleviate a lot of that tight coupling that goes on with an API to the specific device.

David: So, what we are doing is putting an abstraction layer in between the program, the automation program that is would typically be calling the API's and the API's itself so in essence its interpreting what the intent is for the automation and then converting that into API calls?

Lori MacVittie: That sounds pretty good, yes, sure.

David: Okay, we'll go with that.

Lori MacVittie: So off for the network.

David: So, this is – yes this is kind of – yes absolutely and this is kind of near and dear to my heart because I think people have a tendency to kind of take the path of least resistance and just couple everything together and tightly coupled stuff is bad where loosely coupled stuff is not bad and ultimately the ability to kind of put complexity behind the domain, put volatility in the domain is a much better way to do it but it takes additional time and energy to make it happen. What percentage do you think would be the – or if there is extra time and energy and cost in doing this versus doing it the imperative way?

Lori MacVittie: So, doing it declarative well, I mean there is tradeoffs, usually it's the upfront cost, if you are going to do declarative you have to really look at what you have and you have to build something or buy something that's going to be able to do that translation, be that engine for you to actually do that translation, talk to all those devices. So that can be time consuming and costly upfront, right, in the long run it's going to be better because you are decoupled, you are more agile, you can change those processes a lot easier, add new ones, remove old ones and as well switch out the other side, right, you can switch out different versions of your networking hardware, your infrastructure a lot easier if it's not tightly coupled. But it is still much faster to just write a Bash script or a Python script to log in to a certain device using API and make this change. (0:06:00) So you are going for quick return but you are accumulating all that debt you are going to have to work at keeping track of all those scripts and what if Bob quits and it only worked on his device and you know all the questions that you have, same thing that's been done in the application world for 20 years where we've gone through this process of well, we really have to decouple things, we have to learn how to be more service oriented, no matter what the architecture is called, and consider what we're trying to do rather than specifying how to do everything. Talk through API's that are easily manipulated a façade if you will, to provide that kind of buffer between us and what we are actually doing.

David: So, going forward this is kind of a better way to do something we've doing for a while and then it's going to save you money in the long run. It's another story with in IT. In other words, you are in for you are looking at the ROI in terms of what you are able to do and it's better to spend the money upfront and get this declarative technology in place so you get these layers in place other than just doing the tightly coupled binding we were doing in the past. So, understanding that how does this kind of relate to DevOps in the network?

Lori MacVittie: Wow, well isn't that really what we're doing, we are trying to make it more agile; we are trying to share more. There is a – I don't want a culture in the network, engineers are highly skilled. They know things about networking that most of us would just – we're just awed by what they know, they can do things in the network can make things flow and it's really kind of amazing it's almost like thinking about a Neurosurgeon how does he know where those all those little things go it's fascinating, I don't want to do it. So, I want the ability to do some of these things with an eye toward being able to take advantage of their skills without being able to have to shadow them forever and learn how to do it directly. So, I'm going to use scripting, I'm going to use automation, I'm going to share, I'm going to pull out information, I'm going to make this reusable, I'm going to pull all of these things from DevOps the concepts of just being faster, sharing more, communicating and not keeping everything as pets on my desktop, and share that out. So, it's partly about doing that, because that kind of tribal knowledge gets locked up inside network engineers.

David: Got it. So, moving forward I mean DevOps is so much about sharing and developers have done such a poor job in sharing in the last 30 years. So I guess you can go back even 40 years in some instances because its we're dealing with different generations of technology it's obviously different languages, different enabling technologies that's one kind of barrier to sharing but the big thing is in the noggins of the people who are going to be working in the Dev organization to in order to kind of get the fact they were writing something for a lots of other people to leverage which is something that's relatively new which shouldn't be relatively new but each time we get generations and developers out of college I do notice they still have that mentality, so how do we break that bad habit?

Lori MacVittie: We break that in Dev? I don't know if we can. There is a mindset to being a developer and I feel like I can say this, because that is where my heart still lies is development. There is a – not invented here, I didn't write it, I don't trust that code, it didn't do it the way I wanted it to, it's the wrong language, its old. There are so many things that go through your mind when you are a new developer fresh out of college you just want to change everything and do it your own way. So, it's really hard to break that because it's kind of built into us coming out of school, we know better, we are new, we are fresh. I think actually in the network what I'm seeing on that side at least is that there is a more – it's not really reluctance but it's a more considerate approach to how this is going to happen and I think that stems from the fact that if I write some bad code today and it doesn't compile, well I don't commit it and nothing happens. If I make a mistake in the network doing this, the blast radius is huge; it can significantly impact the entire business. Maybe just one App if I'm lucky, but it could be all Apps it could be the entire business, you are talking about a very scary thing to do. So, I think on the network side they take a little bit more of a considerate approach to how they are going to do this kind of automation and the sharing of information and they are taking their time. I don't see them jumping in, they are not – we need to change all these things it's a – how do we make this more efficient and do this in a way that's going to keep stability but give developers and business the speed that they need, and they are trying to balance those two, because they are very aware of just how fragile and important the network really is. So I think on that side, it's a lot easier to get that we want to talk, we need to communicate, we have to have this kind of conversation before we jump into it because it is so impactful to other people and I think they recognize that already so there is – it's a lot easier I think to build that sharing culture and that collaboration on the network side as oppose to the DevOp side and the Dev side where even our architectures are leaning toward the cowboy coder, microservices, code local, code small, this is mine. I mean it almost feeds my need to be like in complete control of little code world, nobody else can have it, nobody else needs it, I don't have to, I just have to share my API's you don't need to see behind the kimono. So even those architectures are kind of lending themselves toward that kind of an attitude even if they are sharing more on the external side they are not really – it still kind of feeding that environment. So, I think there is a different culture already that's kind of helping the network side be a little bit more open about what they are doing and how these things are going to work.

David: So, I'm an enterprise CIO and I just read this blog post and I go this is amazing I want to figure out how to utilize this kind of methodology, this kind of approach within my own organization. Lori, tell me how to do it?

Lori MacVittie: Wow, yes just do it. No jump in no – I like to start with strategy. I don't think that you can just grab this and say "Okay, I'm going to start automating everything in my network and use this" whether it's declarative or imperative you can't do that. You have to have a strategy and you have to kind of learn from the failures that you mentioned on the Dev side. We have to understand standardization is going to be a boon, we shouldn't have people using Bash and Python and Pearl and Go and Ruby and 15 different languages on the network side because these guys aren't developers to begin with otherwise they would be developers, they are network engineers. So, you need to standardize and get your platforms in order, understand how you are going to do it first, actually have a strategy for what you are going to do first, how you are going to do go about it, the languages that you want to use and the techniques you want to use. So, you need to learn those tool chains, understand how to integrate them, make decisions and put a strategy in place before you jump in. Because otherwise you are going to run into mistakes and you are going to have errors and you are going to have downtime and it's not going to be pretty. So, you want to take a strategic approach to automation rather than just either accidentally or tactically fall into it.

David: So, we are dealing with tools and technology ultimately you are going to assist us in doing this so after we create the strategy, when you figure out the tactics, then how we are going to do it and assuming we can change hearts and minds let's say in sort of enabling technology should we be looking at.

Lori MacVittie: I think there are several you first need to understand some of the basics about how you are going to go about automation and integration which is going to be – I need to understand HTTP, I need to understand GET. How do I use playbooks and Ansible and what's the core of how an automation engine works, how does it take this thing in and then spit out all these commands and do these things. So, there are some very basic concepts that have to be embraced and learned and taught to the people that are going to be doing this. And I think one of the mistakes that we always make no matter what technology it is, I'm just going to run out and I'm going to hire the talent to do this, because that's the way we do these things. But when it comes to things like Cloud and automation and all of these things we are running at a deficit right now in terms of talent. And it's very hard to attract that talent and then keep it in the organization and it's much better to train up your people, get them into programs, buy them the books they need, there are online courses, there are super NetOps, take some kind of training that's going to give you those basics, invest in your people because they are also the ones that understand not only your systems but your processes. And it's very hard to teach outsiders the business and the processes so that they could do the work, it's much better to invest in those people you already have who have that expertise and just give them more skills plus then you are building a talent pool that's going to help everyone in your geographical area where you couldn't find a talent to begin with. So, start with the basics and just getting familiar with those kinds of things before you jump into – all right now here we are going to build the system. You got to start with the building blocks.

David: So ultimately this has been discussed in one way shape or form in the last 20 years I think I wrote three books on this kind of topic in the ability to kind of leverage services and reuse them and put volatility in domain and configuration and abstraction layer and abstracting people away from the complexity – and we kind of went through this with the component based stuff and certainly distributed object based stuff and certainly the service oriented stuff kind of brought it home. And so here we are in 2018 and we are looking at micro-services everything you just mentioned Code local and all the things I hear about all the time. People declare their religion to me now and they say “I code local” “Okay, well what does that mean?” Ultimately you got to get something done, you got to get something in place and by the way I prefer it to be done the right way where you can actually create something that's reusable and discoverable and then I can abstract into other things and we're always leveraging the same abstractions to the APIs versus various kind of proprietary one-offs which people have a tendency to do. So, I wouldn't say we failed, but we certainly didn't live up to expectations around the SOA stuff and kind of early Cloud here we are in 2018 we just have a vast array of additional tools and technology, storage is almost free, compute almost free we have an understanding of how to do DevOps right and agile right and how to integrate these various systems. What's going to change now in 2018 to 2020 and that's going to get us onboard with this approach and methodology and way in doing something that we just kind of missed in the past.

Lori MacVittie: Just the continued explosion, the pressure we cannot continue to grow like we have. You can't just throw more hardware at the problem, you can't just throw more people at the problem. There are three million different laws that tell us we can't do that at some point it just becomes unwieldy and it actually slows down and impedes the process. So, I think just the volume of applications and data connections, devices, things, the demands of consumers and like corporate employees is going to force adoption of this kind of methodology and this kind of movement forward, because they just won't be able to keep up otherwise. And it's going to impact the bottom line and that's really what will drive adoption is it's going to impact the bottom line. This is an App economy that's the currency and if you can't get an application developed which they can they are doing it really fast and then it kind of runs into the network and it just hangs there for a while, it's got to get through that pipeline faster we have got to make changes that are going to make that happen not only faster but to do it smarter. Get it out there so that we can make those kinds of changes and keep up with the business. Otherwise you are going to fail. And it's not going to be a nice fail, it's going to be a business fail and that's really, I think the change that's going on is that IT is no longer just a loss center, it's not just a support group, it's part of the business, it is the business. I had a CIO tell me, “20 years ago we are not in the business of IT” and I don't think there is a business out there today that can say that. Everybody is in the business of IT today, by virtue of the fact that we have to get applications out there. APIs it's how we build our supply chains, our distribution centers, if you are B2C, if you are B2B it doesn't matter you are using technology and applications to make your business run so you absolutely have to be in the business of IT in order to succeed moving forward.

David: Yes, I think we are going to see some businesses that are going to go down over the next 10 years. Some of the brand names we've been dealing with for a 100 years are just kind of going to go away because they are going to get their lunch eaten by other people who are able to do this well and disrupt them in the market which is something we haven't seen, we haven't looked at IT as a strategic enabler and then suddenly we see the disruptors starting to emerge every point to Uber and Airbnb and Netflix and all the other things but there are 1000s of companies out there that are getting into spaces that are kind of held by traditional companies and that are able to disrupt the space because they are able to move in a much more nimble direction, they are able to take service orientation, Cloud enablement, able to take this technology kind of to the next level. And then suddenly I think the big companies never knew what hit them – Amazon coming from online

background in selling books and to disrupting the entire retail space and the ability to disrupt the insurance space I think that's coming the former space I think that's coming to healthcare space, I think that's coming where the new players are going to be very different than the older players and these guys who end up going out of business because we are not paying attention to technology like this and how it's really going to be a key game changer and enabling them to delight the customer more to automate things in the background to make things occur in an auto magical way and then also be able to get Apps out there at the speed of need which is something that I think the big companies can't do. Even if they move to the Cloud their development processes are so so bureaucratic and kind of politically charged that they can't seem to get the applications out the door to solve the problems that need to be solved and therefore they end up changing slower than their market is changing which is basically a sign of debt, what do you think?

Lori MacVittie: I think you are right. I mean there are certain things that are always going to be done or not always going to be done the way that they are done but there are certain you can't make a roll of paper a different way it's still a roll of paper you still have to the pulp and all the chemicals and the pressers and all the things that go into that. But the way that it's sold and bought and purchased and acquired and all of those things is changing. And it's possible for them to be disrupted to the point where that's all they are its just churning things out and there is no other business for them. There is also opportunities out there for businesses they are going to miss. So even if they continue to exist doing what they do they are never going to really grow or succeed or go beyond that and take advantage of new opportunities because they are going to miss them because they are not focusing on the Apps and the ways to connect with people.

David: I couldn't have said it better myself. So where can we find this blog online and your other blogs as well?

Lori MacVittie: So, this blog is at devops.com and I don't have the URL in front of me. So just search for my name I guess and it's out there and I have a whole bunch on "Automation and NetOps" on devops.com and then I write on for f5.com and that's accessible through devcentral.f5.com that will have a nice list of everything that's been written in the last 10 years.

David: I got it in right in front of me its devops.com/author/lmacvittie M-A-C-V-I-T-I-E how's that sound?

Lori MacVittie: That is perfect, yes.

David: There you go, Lori it's great to have you on the podcast we are going to have you again pretty soon and looking forward to your additional research and I'm looking forward to stealing from your blogs.

Lori MacVittie: All right, thanks this was great and I look forward to being on again and steal away.

David Linthicum: You got it.

Visit the On Cloud library

www.deloitte.com/us/cloud-podcast

About Deloitte

Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited, a UK private company limited by guarantee ("DTTL"), its network of member firms, and their related entities. DTTL and each of its member firms are legally separate and independent entities. DTTL (also referred to as "Deloitte Global") does not provide services to clients. In the United States, Deloitte refers to one or more of the US member firms of DTTL, their related entities that operate using the "Deloitte" name in the United States and their respective affiliates. Certain services may not be available to attest clients under the rules and regulations of public accounting. Please see www.deloitte.com/about to learn more about our global network of member firms.