Deloitte.

For Cloud Professionals
November 2021

The Deloitte On Cloud Podcast

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

Title: **Four essential tips to help build a sound cloud architecture**

Description: Cloud architecture is more art than science. Building a solid architecture takes a broad understanding of both IT in general and cloud specifically, and it also takes lots of up-front work. In this podcast, David Linthicum shares four key tips for building a sound cloud architecture: Focus on the business value, build in security and governance from the get-go, have a solid plan for ops and tech, and test as you go to ensure the final product meets your needs from day one.

Duration: **00:17:25**

David Linthicum:
Welcome to this Deloitte On Cloud Podcast Knowledge Short, exploring a specific topic related to cloud computing. This is a short tutorial talking about the real-world concepts in the emerging world of cloud computing. I'm your host Dave Linthicum, cloud computing subject matter expert, author, speaker, and managing director with Deloitte Consulting. And this is Four Cloud Architecture Tips to ensure success.

First, let's kind of define what architecture is. It really is the configuration of technology so that it's optimized to deliver value to the business. That's kind of the academic way that we define how we're designing, deploying different technology sets, and doing so in such a way where we're not only able to fulfill the needs of the business, but do so in an optimized and cost-efficient way. And, so, cloud architecture is really an art more so than science. It's about having a general knowledge about lots of things, including databases, networks, application development, cloud computing, edge computing, IoT computing, and having the ability to look at a business problem and configure technology successfully to solve that business problem. So, let's keep that definition in mind as we move through these four tips.

The first tip is focused on the business value, and not what tools and technology are hyped at the time. Keep in mind this is about serving the business. This is not about us being able to leverage whatever technology we think is going to be the most popular at the time; it's around the right technology to solve the problems for the business. And that's always going to be different, so there's no one-size-fits-all solution. It's going to be looking at the particular problems in the business and then backing the appropriate technology to solve the problem.

Now keep in mind if we're focused on business value, we're going to consider a couple of things including hard costs and soft costs. Hard costs are really the value that we save that we can see in terms of the efficiency that whatever technology we're leveraging is going to bring. Cloud computing typically is going to operate at a less cost than leveraging on-premise systems, but not always. So, it's the ability to define the fact that our storage is going to be 20 percent cheaper, our compute is going to be 30 percent cheaper, our networking is going to be 10 percent cheaper, and be able to define metrics that prove that those cost savings exist.

Second, and probably more difficult to define, is soft cost. Soft costs are really the business value of leveraging a technology that's typically harder to define. People say, "We're leveraging cloud computing for agility. We're leveraging cloud computing for innovation." Well, how do you measure that? How do you create the metrics and create the formulas that are going to show that you're bringing this value into the business by leveraging this technology? And, so, you need to figure out what the business is, the current as-is state, where the business is going, the to-be state, and what kind of innovation value, what kind of agility values are being created by moving to cloud computing or any technology for that matter.

In some cases, it's going to be fairly slight. Certainly, if you're a manufacturing organization, or you're some business that doesn't necessarily benefit from agility and innovation, or at least to a lower degree, your soft costs are going to be lower. However, if you're a bank, if you're a high tech company, if you're able to leverage technology as a force multiplier to take the business to the next level, then your soft costs, basically how you're going to provide strategic value to the business, are going to be a lot higher.

Also, we need to determine a few things, such as what does success look like? In other words, what metrics are going to be in play for us to define something as successful and something as not? So, is it saving ten percent off operational costs? Or is it leveraging technology as a force multiplier for the business so we can add value to the innovation processes and add value to Agile things? Are we able to change the business around the changes in the market?

And keep in mind, this is iterative and baby steps. So, in other words, we can't define holistically what cloud computing, or any technology, is going to bring in terms of value to the business. And, so, we're doing these things in iterative ways. We may be dealing with 100 applications at a time, 5 different databases at a time, those sorts of things. And in doing these things in incremental steps, or sprints, we're able to define smaller value that the technology is able to bring, which gets up to a more holistic value. And keep in mind, that's going to be a healthier way to do it because, as time goes on, technology is going to change, and you're going to find more value within the technology that you're able to leverage.

Number two is, understand that both security and governance need to be built into the systems and can't be added later. You have to remember that both security and governance are going to be systemic through pretty much everything you do as the role of an architect. So, they're systemic in understanding the requirements, in other words, where security needs to fit in—including compliance issues such as encryption systems that have to be in place because of laws and regulatory things, the ability to deal with policies within the organization in terms of how things are protected, how data is encrypted, dealing with certain vertical-industry security requirements such as dealing with PII information, dealing with HIPAA standards, things like that. PII, by the way, stands for personally-identifiable information. So, you need to understand many things in terms of your requirements. What are the regulations that are in place? What are the standards within the business that you have? What security is going to be required for the future state of the business? And all that needs to be defined before you start building the configuration of anything, before you start doing architecture.

So, we need to do this in a few steps. Number one, defining the migration of applications and data, as well as building net new applications and data stores. So, no matter if we're moving technology, applications, and data from one on-premise system, for example, into the public cloud, or building an application from scratch in the public cloud, we have to consider how security needs to be factored and built into both the applications and data storage systems. And this means defining how we're going to secure the application before we build it, and the reason why is because in many instances the application has to be designed from the ground up to deal with particular types of security.

So, it's difficult to add these things after you build and deploy the application, because you're going to have to undo lots of things you already did, versus building the application from the ground up with security in mind and designing these things into these systems. We have to consider security around deployment, how we're going to deploy the systems to the platform of choice, the platform-level security, application-level security, data-level security. All that has to be factored in.

And then also how security and governance is going to be played in the operational state. Keep in mind that security operations, or SecOps, is an important concept as we move into the cloud. And the reality is that security has to be operated like any other system in order to do things like spot breaches and the ability to spot vulnerabilities that we need to patch within the systems. This is all part of operating in the cloud and, really, all other aspects of dealing with cloud- and non-cloud-based systems. So, security should be considered at every step in the game. In fact, one of the things that I do, and other people who are cloud architects do, is they always have a security engineer that's part of the architecture team, someone who's responsible for looking at the requirements of security and making sure the right mechanisms are being built in or paying attention to the larger aspect in terms of how security's going to play a role in building our systems.

Next, and a very important concept, think about ops issues and technology upfront and how the holistic cloud solution can be operationalized. Now this means that we're doing ops planning, and as we're building and deploying these systems, we're migrating systems to the cloud, or building systems in the cloud as net new systems. We're figuring out how they're going to be operated. In other words, who's going to deal with data backup and disaster recovery systems? Who's going to deal with performance issues? Who's going to deal with outages? Who's going to deal with all the things that really pop in terms of operating a system longer term?

Now keep in mind that this is how success is going to be defined in the minds of most users out there. They want the system to be up and running when they need it, they want it to perform well, and they want outages to be few and far between, if not ever occurring—if they can be eliminated, or getting to a

state of operational excellence where virtually outages are unheard of. So, the way you get to that state is putting the time in to do operational planning that occurs at the beginning of the process. We need to understand how much ops will cost us before we build and deploy cloud-based systems.

We need to figure out how to mediate complexity in terms of operations, certainly if you're moving into a multicloud environment where you're dealing with two, sometimes three, cloud brands, cloud providers, and different security systems and monitoring and operations systems, governance systems, things like that. How are you going to operationalize those systems holistically? So, looking for opportunities such as leveraging AIOps tools and monitoring and management tools that are able to operate cross-cloud, in other words, abstracting you from the complexity so we're just dealing with a single set of interfaces which are able to account for the differences between the different native services that are running within the cloud provider.

All these tricks are available to you. The core message here is that you need to figure out what operational aspects of the system need to be built, need to be planned for, and the enabling technology that you can leverage to make operations easy.

The final tip is one that many architecture teams miss: the ability to allow plenty of time for testing, including performance testing, unit testing, smoke testing, all kinds of testing to ensure that what we've built and designed is going to live up to expectations, not only on paper but showing or proving that it has the ability to be something that can be operationalized and is going to provide value to the business for many years to come. You can only do that with testing.

So, ultimately, this does not only occur at the end of the project. This occurs during the project. So, you can't get to the end state and holistically deploy things on public cloud providers and different technologies and different databases, different application development platforms, and then suddenly test it, because if you do that, you're going to find that many things are wrong with the system. Some of the things that were assumed to work in a certain way don't work in that way, and they have to be reset and taken out.

The problem is that there's technologies within these architectures that are all interdependent. So, in other words, we may pick a technology that's dependent on a particular database that's dependent on a particular public cloud brand, and by doing so removing one of those components because it fails a test at the end of the process is going to require us to swap out lots of other things besides that component. So, the message here is that testing is continuous and should occur around the selection of tools, cloud providers, databases, internet of things platforms—anything that would be a component in the cloud solution as you pick them. So, in other words, we unit test them to make sure that the tool, the technology is able to live up to expectations—very important, because you don't want to find out that that tool or technology doesn't work or is incompatible with certain things that you've picked at the end of the process. You need to find that out before you pick it, before you invest the time and money into it, and before it becomes part of your solution.

Next, holistic testing occurs using simulators. So, in other words, the ability to do performance testing by doing performance modeling systems—lots of great tools out there to figure out what performance is likely to be, whether you're dealing with a database, application. Simulations provide the ability to do this without actually deploying the technology, so you're able to do it on the cheap.

However, you have to do real testing as well, and that includes segment testing, or the ability to test certain segments of a particular application. For example, just testing the application that's communicating with a single database, or testing how a container is going to work and running within a Kubernetes cluster and disconnected from the rest of the systems. It's just looking at these certain things, this small group of technology that's running in a particular segment.

And then finally, holistic testing and acceptance. In other words, this is where we're looking at the whole enchilada at the end of it. In other words, we have our architecture. We have all our systems; we may have 100 or 200 different components that sit within our architectures. We've component tested these things. We performance tested each one of the technologies. We've done some segment testing. How does the whole thing run?

Now, while it's disappointing to run into issues when you deal with holistic testing, it's not unusual. We're going to find something where a database may have a performance problem if it's communicating with a particular application development tool because of some sort of incompatibility that has been built within that tool but not necessarily discovered yet. So, you work with the vendor to fix the problem and kind of move on. So, this is about getting realistic in terms of how this thing is going to operate and how the users are going to be experiencing the system.

And more importantly, is the system able to provide the value that we've defined to the business? So, we need to consider security testing. We need to consider performance testing. We need to consider functionality testing—make sure the applications do what they're supposed to do. We need to consider the business value, and we need to consider human factors. In other words, how easy is it for human beings to interact with the system based on user interfaces we've built for human beings to interact with?

So, we covered a lot of stuff, and it's only really the tip of the iceberg in terms of how you do cloud architecture. And some of these tips are not going to be news to many of you. Some of them are maybe things that you need to add to your existing architecture process, procedures, methodologies, workbooks, plans, things like that. Ultimately, good architecture is about considering everything and not assuming that things are going to work. It's really about providing architectural designs. It's really about leveraging technology in a certain configuration where we're able to get to an optimized state. And keep in mind that we have lots of technology configurations that will work, but just working is only solving part of the problem. At the end of the day, we're looking to move to the most cost-optimized solution that we can, so—and that's typically one configuration. So, it's looking for that optimization. It's looking for the ability to bring as much value back to the business that we can through the configuration of this technology. So, I hope this helped.

So, if you enjoyed this podcast, make sure to like us, rate us, and subscribe. You can also check out our past episodes including those hosted by my good friend Mike Kavis. Find more out at DeloitteCloudPodcast.com. If you'd like to contact me directly, you can e-mail me at DLinthicum@Deloitte.com, L-I-N-T-H-I-C-U-M.

So, until next time, best of luck with your cloud journey. You guys stay safe. Cheers.

Operator:

Visit the On Cloud library
www.deloitte.com/us/cloud-podcast