



NoOps in a serverless world

Shift IT's focus from operations to outcomes

WE HAVE REACHED THE NEXT STAGE IN THE EVOLUTION OF cloud computing in which technical resources can now be completely abstracted from the underlying system infrastructure and enabling tooling. Cloud providers are continuing to climb the stack; rather than simply providing everything from the “hypervisor on down,” they are now—through their own focus on hyper-automation—taking on many core systems administration tasks including patching, backup, and database management, among others. Together, these capabilities create a NoOps environment where software and software-defined hardware are provisioned dynamically. Going further, with serverless computing, traditional infrastructure and security management tasks can be fully automated, either by cloud providers or solution development teams. Freed from server management responsibilities, operations talent can transition into new roles as computing farm engineers who help drive business outcomes.

Traditionally, the CIO's responsibility of keeping business-critical technology systems running has absorbed up to 70 percent of IT's budget as well as considerable amounts of labor bandwidth. Cheaper storage, cloud, and outsourcing have lowered this budgetary outlay by 20 percent or more. Yet in an era of perpetually tight IT budgets, finding ways to redirect financial and human assets from operations to innovation remains a top CIO goal.¹

In previous issues of *Tech Trends*, we have examined how CIOs are pursuing this goal by transforming their technology ecosystems from

collections of working parts into high-performance engines that deliver speed, impact, and value. From the bottom of the IT stack up, they are building infrastructure that is scalable and dynamic, and architecture that is open and extendable. From the top down, CIOs are rethinking the way their IT shops organise, staff, budget, and deliver services.²

In many reengineering initiatives, automation is the keystone that makes meaningful efficiency and cost reduction achievable. With more of IT becoming expressible as code—from underlying infrastructure to IT department tasks—organisations

are applying new architecture patterns and disciplines in ways that remove dependencies between business outcomes and underlying solutions. They are applying those patterns as well as engineering techniques to redeploy IT talent from rote, low-value work to the higher-order capabilities.

Now, as part of a growing trend, CIOs are taking their automation efforts to the next level with serverless computing. In this model, cloud vendors dynamically and automatically allocate the compute, storage, and memory based on the request for a higher-order service (such as a database or a function of code). In traditional cloud service models, organisations had to design and provision such allocations manually. The end goal: to create a NoOps IT environment that is automated and abstracted from underlying infrastructure to an extent that only very small teams are needed to manage it. CIOs can then invest the surplus human capacity in developing new, value-add capabilities that can enhance operational speed and efficiency.

Serverless' value proposition is driving considerable interest in the serverless market.³ A recent Cloud Foundry global survey of 600 IT decision-makers found that 19 percent of respondents were already using serverless computing, with another 42 percent planning to evaluate it within the next 24 months.⁴ Moreover, MarketsandMarkets, a B2B competitive research firm, projects that the value of the serverless architecture market will reach US\$14.93 billion by 2023, up from US\$4.25 billion in 2018.⁵

Thus far, a number of large companies including Netflix,⁶ Coca-Cola,⁷ and the New York Times Co.⁸ have been at the vanguard of the serverless trend. In the next 24 months, expect more organisations to begin following their lead, exploring ways to use serverless to scale their DevOps practices and to build greenfield applications. The goal of achieving pure NoOps environments may take several years

to achieve, but across industries, the transition, however preliminary, is underway.

Enough already with the “care and feeding”

For the purposes of discussing this trend, the terms *NoOps* and *serverless* are not interchangeable. “Ops” comprises any number of operational areas—think networking, security, management, and monitoring. In the marketplace and in the context of this technology trend, the term *serverless* basically describes server administration. To further muddy the definitional waters, both terms are misnomers: With the serverless model, there are still servers, but their functions are automated; likewise, in NoOps environments, traditional operations such as the code deployment and patching schedules remain internal responsibilities—they are simply automated to the extreme.

Both terms can trace their roots to the first as-

Serverless computing is an umbrella term for a spectrum of cloud-based options available to organisations wishing to get out of the business of managing servers.

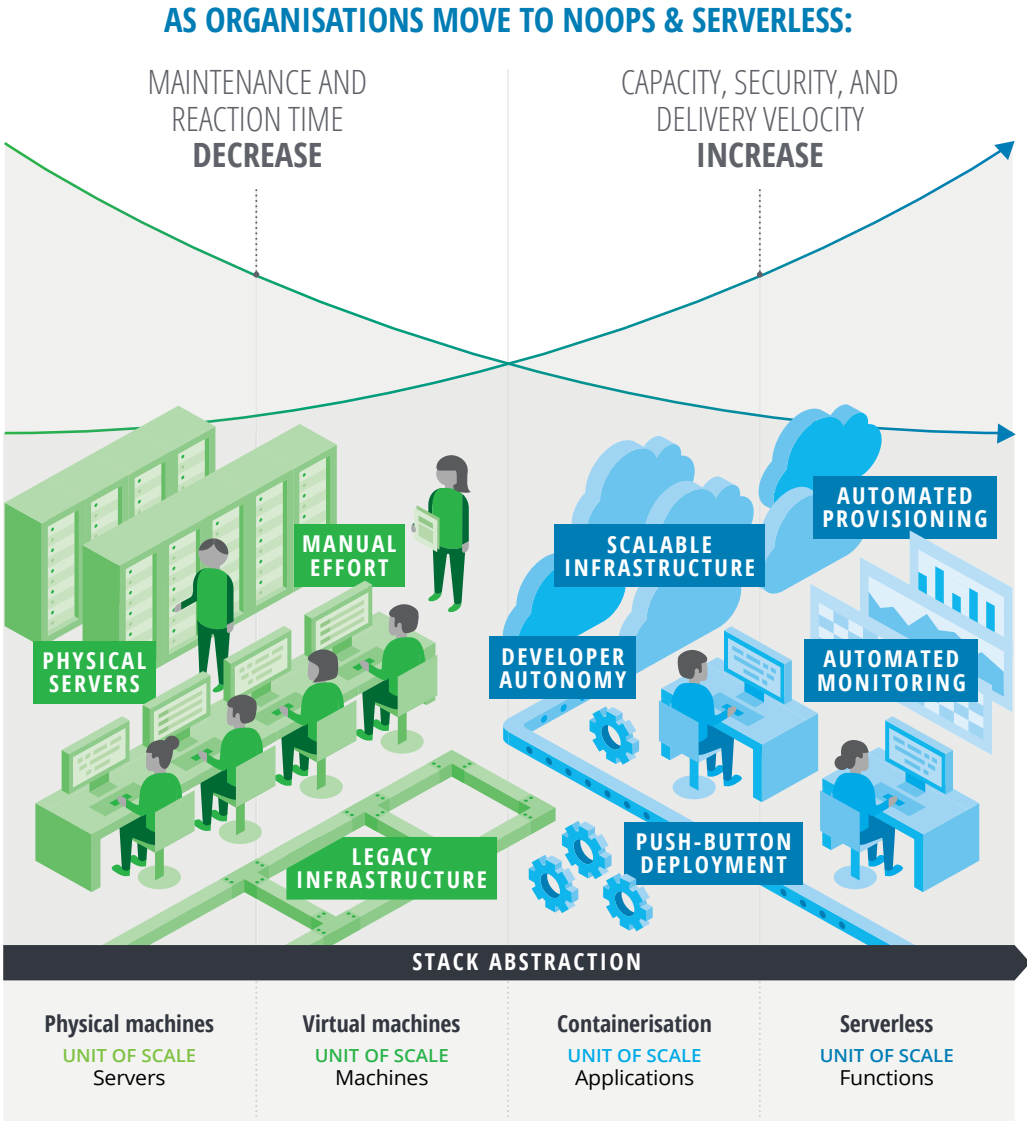
a-service offerings and the dream that one day IT organisations would be able to hand onerous care-and-feeding responsibilities of enterprise systems to someone else. Today, *serverless computing* is an umbrella term for a spectrum of cloud-based options available to organisations wishing to get out of the business of managing servers. At one end of this spectrum is the platform-as-a-service model in which customers buy always-on access to a database. At the other end is the function-as-a-service model, which offers a fine-grain pricing model, substantiating and running code only when a customer needs

it. As such, customers pay for only the requests they make.

Serverless computing offers CIOs a toolkit for transforming their IT operations. Its potential benefits include:

- **Infinite scalability and high availability.** Functions scale horizontally and elastically depending on user traffic.
- **NoOps (or at least less ops).** Though operational tasks such as debugging typically remain in-house, infrastructure management is fully outsourced.

FIGURE 1
The digitalisation of IT



Source: Deloitte analysis.

- **No idle time costs.** In a serverless computing model, consumers pay for only the duration of execution of a function and the number of functions executed. When a function is not executed, no charge is levied—thus eliminating any idle time. In terms of cost, this represents an improvement over legacy cloud computing models in which users are charged on an hourly basis for running virtual machines.

In many companies today, cloud strategies remain works in progress, as do efforts around virtualisation and containers. Likewise, serverless computing, with its promise of a NoOps future, does not suddenly render these efforts redundant, but it does offer a vision of a highly automated end state to which CIOs can aspire.

And aspire they do. In Deloitte's [2018 global CIO survey](#), 69 percent of respondents identified “process automation and transformation” as the primary focus of their digital agendas.⁹ In the coming years, serverless will likely be a key technology that many CIOs use to automate the deployment, scale, maintenance, and monitoring of applications. Today, cloud vendors are steadily adding new capabilities such as databases and natural language processing interfaces to their portfolios of serverless offerings. It is now possible to build greenfield applications without deploying a physical or virtual machine. By understanding the investments that companies are making today in serverless computing and by embracing the broader *NoOps in a serverless world* trend, they can make this transition foundational to their short- and long-term digital transformation agendas.

Chasing the elusive NoOps dream

The NoOps trend is gaining traction in part because it offers a new way of looking at an age-old problem: How can we make resources go further? For budget-strapped CIOs whose enterprise fief-

69%

of Deloitte's 2018 global CIO survey respondents identified process automation and transformation as the primary focus of their digital agendas.

doms generate no revenue directly, this question remains largely unanswered. But one thing is clear—there is not a lot of business value to be found in maintaining servers and data centers. Keeping on the payroll people whose expertise lies solely in patching servers has traditionally been just another cost of doing business.

The NoOps trend offers CIOs an opportunity to shift these employees' focus from patching, monitoring, and measuring to higher-value engineering and development tasks. More broadly, this trend makes it possible to manage IT operations more efficiently using automation and orchestration capabilities that others pioneered and have proven.¹⁰ Much of what we think of when we talk about *NoOps* and *serverless* are infrastructure components that Amazon, Google, and Microsoft developed to support their as-a-service offerings. These vendors realised that these same components could benefit their as-a-service customers too, particularly in the area of software development.¹¹ In a *NoOps* model, developers no longer have to coordinate with other teams to execute minor tasks involving underlying infrastructure, operating systems, middleware, or language runtime.

Transitions from traditional to serverless environments do not happen overnight, a fact that helps to mitigate the fears that some in IT may have about

job security. During these transitions, operations talent may still have to do some routine database tasks and make sure that core systems are tuned and maintained. But they will now have the bandwidth to upskill and redefine their roles; perhaps more importantly, they can begin approaching operations tasks less from a plumber's perspective than from an engineer's. Many may find this a much better place to be professionally: Writing software that monitors and heals certainly seems preferable to getting an urgent 2 a.m. text that a critical system is down. More broadly, think of this as transitioning operations talent from being reactive to proactive, and finding new opportunities to leverage automation. In the NoOps world, IT talent engineers variability out of operations, thus making things routine, repeatable, efficient, and effective.

Working with serverless platform vendors

At present, several major cloud providers are offering serverless platforms that can help users move ever closer to a NoOps state. Amazon, Google, and Microsoft dominate today's serverless market. Alibaba, IBM, Oracle, and a number of smaller vendors are bringing their own serverless platforms and enabling technologies to market.¹² Meanwhile, open-source projects such as OpenFaas and Kubeless are attempting to bring serverless technologies from the cloud to on-premises.¹³

The serverless model offers several advantages, particularly over the IaaS and SaaS models for which customers often pay a fixed monthly or yearly price whether or not they use the entire capacity provided. By contrast, serverless models charge customers for only the resources consumed for the life of the function that is called. It's a fine-grain, pay-as-you-go model, with significant projected cost savings over other cloud models for many workloads. For example, as competition in the serverless space heats up, it is not unreasonable for users to get up to a million free compute requests per month,

which provides a large amount of computing power without huge upfront costs.¹⁴

As you explore serverless offerings, be aware that the serverless computing model is still evolving—it should not be construed as a cure-all for development and operations problems. For example, the production tooling that provides visibility in serverless development environments is currently limited. Recently, cloud infrastructure provider DigitalOcean surveyed 5,000 development professionals about the challenges they have encountered when using serverless. Their responses varied, but respondents identified the following major areas:¹⁵

- **Monitoring and debugging.** Unsurprisingly, 27 percent of survey respondents cited monitoring and debugging in a serverless environment challenging, which is perhaps unsurprising given the ephemeral nature of serverless computing. Capturing the information needed to monitor and debug is more challenging with a serverless model because there is no machine to log onto. In some situations, developers working to debug tricky problems may be forced to log manually into a data store. The good news is that a new generation of debugging tools and applications that make it possible to run serverless functions locally are emerging.
- **Vendor lock-in.** Concern over vendor lock-in emerges often in the early stages of disruptive technology waves. Until industry standards are set and a single model becomes a market leader, early-stage customers are often concerned that they will pick the wrong horse. What if you are locked in an agreement with a vendor whose products eventually become nonstandard? If you want to switch vendors, you could face significant costs to retool and redesign your architecture. DigitalOcean found that 25 percent of respondents were concerned about getting locked into an agreement with their serverless vendor. In many cases, a suitable architecture can minimise your ties to a particular vendor. With function-as-a-service, for example, it is

possible to abstract your business logic from the serverless “handler” to make porting easier. With other serverless features, consider weighing their benefits against the potential costs of getting locked into an agreement you may neither want nor need.

- **Migration.** Roughly 16 percent of survey respondents cited migration as challenging. Indeed, for large companies, migrating at scale is no small task. For example, it can involve re-architecting

one or more of your applications (as in the case of function-as-a-service) or at least swapping major system components such as databases. For this reason, some companies may view full-scale migration of their application portfolios to serverless as overly costly and disruptive. They may opt instead for migrating select existing applications or those that are a part of greenfield development initiatives.

LESSONS FROM THE FRONT LINES

CARGILL'S FUTURE-READY FOUNDATION



By the year 2050, the planet will be home to some 9.5 billion people. That's a big number—one that requires companies to think and act differently. For Cargill, a leader in the food and agriculture industry, its sole purpose is to nourish the world in a safe, responsible, and sustainable way. Cargill recently placed an increased focus on innovation and technologies that will help the 153-year-old company transform and effectively address some of the world's greatest food challenges today and into the future.

One way Cargill is making improvements to its software engineering capability is by automating the development life cycle. This shift via technology not only drives the business, but also empowers Cargill's developers to write code without the worry of deployment or packaging. "A strong digital foundation helps us work better while serving our customers and businesses more efficiently," says Keith Narr, vice president, Cargill Digital Labs.¹⁶

The renewed investment in engineering has been an important part of Cargill's core modernisation and cloud journey and supports the ambitions of technology-minded but business-focused leaders within the organisation. Development and operations standards are being embedded in the technology platforms and automatically enforced behind the scenes. Additionally, Cargill developers have embraced automated security scans leading toward DevSecOps, as well as providing the backbone for API-based development, and paving a path toward open standards and open platform adoption.

Narr and his team worked to ensure that the resulting cloud journey wasn't relegated to a lift-and-shift exercise: the same old capabilities running on a new technology stack. He instead used the opportunity for Cargill to transform the IT landscape within the organisation by exploring new ways to work. A refactored core and a modern architecture, rooted in autonomy and DevOps with an eye toward

NoOps, enables the development of modern applications that run and operate on a scalable platform and can self-monitor and self-heal. "The technology is the easy part; it's the mental shift that's difficult," Narr says. "As part of this journey, we are retraining ourselves to think differently about our expectations of technology and how we consume it."

A big part of the NoOps journey is rooted in breaking down the walls between traditional IT and the business. While building the new platform, Narr and his team started by seeking out business units that already possessed a startup mentality, understood the value of the platform, were already using DevOps processes, and were eager to embrace a NoOps philosophy. The IT team continued to build out the platform capabilities based on what those business-unit teams needed.

Recognising the need to overcome the mental shift, Narr's team also looked to raise awareness beyond early adopters. To demonstrate the potential of the NoOps model, Narr took the entire IT leadership team through a six-hour DevOps 101 boot camp so they could see the impact the capabilities could have on their individual businesses. The team was fully engaged—writing code snippets, checking in source code, deploying, and witnessing firsthand the power of a fully-automated continuous integration and continuous deployment platform. The effect was eye-opening for the business.

"The first 12 months of the journey were intentionally grassroots and focused within IT, building out a core set of capabilities before people started asking for it," Narr says. "As adoption grows and success is mounting, the emphasis shifted from the platform to the outcome: the prototype, the business proof of concept, or the new product rapidly brought to market." As Cargill's strategy evolves, the platform is becoming an embodiment of the "future-ready foundation" that is a cornerstone of the company's technology road map for its business strategy.

MUTUAL DECISION: BUSINESS AND IT PARTNER IN COMMONWELL'S SYSTEM MODERNISATION



When the Commonwealth Mutual Insurance Group set a goal to significantly increase its premium growth rate, the company aimed to simultaneously maintain high standards of member value, member service, and employee engagement. Leadership recognised that this would require a fundamental shift in how supporting technology was deployed. To drive innovation, Commonwealth's small IT department built a partnership with the business, and their common goals led to a core systems modernisation project. This project was executed within a DevOps framework and is changing the way IT delivers services to the business. The success of that journey has given rise to the organisation's next major transformation: moving toward a NoOps model and serverless environment that will further enable business transformation and change how the IT team operates and manages its infrastructure.

"Some people think of DevOps as eliminating steps of the development life cycle, but we see it as expediting some of the more repetitive, time-consuming parts of workflow," says Commonwealth solution delivery manager Paul Stamou¹⁷ "We want to deliver on the promise of the business rather than just focus on keeping the lights on."

Because Commonwealth's IT department is streamlined, it has chosen to implement a cloud platform that provides an automated and secure foundation. The solution is based on an "infrastructure as code" approach, enabling better agility. The IT management processes—backup and security features, in particular—will be written into the configuration and deployed on containers. The serverless platform will allow the IT organisation

to build and deploy applications as cost-effective services that provide built-in availability and scalability. This will enable Commonwealth IT to focus on its business outcomes instead of managing servers, with automation provisioning capacity on demand.

"Our previous solution required a significant amount of infrastructure and human intervention to maintain and service operations," says IT vice president Jennifer Baziuk.¹⁸ "The platform and a NoOps model is a fundamental shift from that traditional approach; we're looking to leverage cloud computing, software as a service, and an ecosystem of partnerships to help us manage operations rather than internalising those costs with our own hardware and human capital."

According to Commonwealth solution architect Justin Davidson,¹⁹ it is less about accelerating infrastructure than about enabling IT to be as agile as the business in driving speed to market and delivering continuous value. While the transition to NoOps is still in progress, Commonwealth's IT team has already seen a significant impact from the move to DevOps and a serverless environment. Additionally, that success has built a groundswell of support and enthusiasm from employees and leadership for the shift to NoOps.

"From a qualitative perspective," Baziuk says, "one of IT's aspirations is to be a trusted adviser to the business. As a result of the success of our DevOps and NoOps strategies, we've earned a seat at the business strategy and development discussions, driving forward the conversation of how IT can be a strategic differentiator. We're freeing up our capacity so we can be the partner we aspire to be, and help make digital part of Commonwealth's business path forward."

VERIZON: FORGING NEW GROUND THROUGH CLOUD



To better serve millions of customers each day, Verizon continuously seeks to advance its networks' performance and efficiency. When leadership saw an opportunity to increase stability and reliability leveraging modern cloud computing technologies, the company launched its cloud migration journey, aiming to both deliver and exceed the performance levels its clients expect, while increasing the level of automation of its systems operations.

Verizon's architecture review board conducted legal and regulatory assessments to determine which of its network systems were eligible to move to the public cloud. Due to the variety and specificities of the applications and workloads, Verizon Network System's strategy combines public cloud, private cloud, and on-premises hardware with a phased migration approach.

In the initial exploratory phase, the teams started migrating nonproduction applications to the cloud while leaving production on-premises, with a goal to learn about cloud services technologies and the advanced automation they provide while developing new skill sets. Traditional cloud computing services in which Verizon's engineers instantiate and manage the equivalent of virtual machines were, in many cases, the initial technologies considered, as they were close to the on-premises environment that some teams were used to. But this bifurcated approach limited the potential benefits.

In the second phase, once the teams successfully deployed the first production applications into the cloud, everything accelerated. Verizon's engineers became proficient in leveraging the emerging serverless cloud services just released by the major public cloud providers. The advanced automation quickly generated benefits, enabled by letting the cloud service provider handle the lower-level cloud infrastructure operations such as database patching

or server instantiation and management. The engineers could focus on developing their application faster and delivering value to the business at an increased pace. As a result, the serverless environment became part of Verizon's directional tech stack and the footprint of such technologies expanded, as did the benefits of the cloud migration.

Some of the early, big wins include migrating a complex provisioning system that handles all of Verizon's service activation and provisioning for fiber-based services. Another was building a cloud-native, microservices-based provisioning gateway that provides a common interface for its many legacy businesses. The provisioning gateway serves as the team's model for building new applications.

"We're completing our first significant year of migration, and part of our portfolio has been moved to the public cloud," says Lynn Cox, senior VP and network CIO.²⁰ "While we still have work ahead of us, we've successfully transitioned several large, monolithic applications, so there are no excuses to say something is too complex. Additionally, I've seen a real culture shift within my team. We are now building new applications directly in the public cloud when possible—a big shift in mindset from where the team was even just one year ago." Verizon is already seeing financial and operational benefits, including greater stability and reliability of applications, increased automation, and auto-scaling of compute resources, which was of paramount importance to Verizon's engineers and technicians in the field.

Within the first year, Cox's team has realized benefits transitioning operational resources to more strategic activities. For example, several teams have been refocused on enabling the deployment of Verizon's next-generation converged core network and on meeting business timelines, as cloud has provided a higher level of automation and shorter lead time than deploying on-premises hardware.

Capabilities enabling 5G rollout and automation have also been delivered faster and with higher levels of reliability.”

The NoOps approach enabled by the serverless environment has been incredibly motivational for team members,” Cox says. “Rather than being limited to a production support role managing operations, they can focus on growing their skills. Now we can lay out a path to the future for them—developing strategic solutions with technologies like

machine learning and robotic process automation—and they see career potential for the long term.”

Prior to the move to serverless and cloud computing, Cox’s team sometimes struggled to keep pace with the needs of their internal customers. As applications have moved to the cloud, that went away. “Now, the question is: How fast can the client work?” she says. “It’s not about the system managing the pace of the client—it’s the client now managing the pace of the system, which has been a big win for us.”

MY TAKE

GENE KIM, AUTHOR, RESEARCHER, AND DEVOPS ENTHUSIAST

For almost 20 years, I've had the privilege of studying high-performing technology organisations, and all our research shows decisively that high performers are massively outperforming their peers, often by orders of magnitude. They ship software to their customers more quickly and safely, which enables them to rapidly innovate and experiment, so that they can win in the marketplace by outlearning the competition.

In the past, this topic was important mostly to technology executives. These days, almost every CEO is being asked about how they are responding to digital disruption, defending their market from the dominant technology platform players, and how they are investing in their software capabilities. In the age of software, almost every act of investment has something to do with software.

It's difficult to overstate the technology miracles that are now possible that would have been impossible a decade ago. Instagram had only 13 employees—six of whom were generalist developers—when Facebook acquired it for US\$1 billion.²¹ Pokémon Go broke the record for fastest time to gross US\$100 million among other records,²² and that was achieved with fewer than 40 employees.²³

I think these examples frame the most important goal of DevOps: to create the conditions in which small teams of developers in a modern business context can achieve the same type of amazing outcomes. Because it's not just for social media and games—it's to solve the business problems that will determine the next generation of winners and losers in the marketplace.

According to one study, developers will be able to raise global GDP by US\$3 trillion in the next decade.²⁴ In my opinion, the majority of this value will be created not by today's tech giants or by startups but, rather, by the largest brands in every industry. These are the organisations that have better access to capital, already have large customer bases, and can access the same amazing technology talent the tech giants can.

This is not a story about “small beats big.” Instead, it's “fast beats slow.” And the best of all worlds is being both fast and big—which is what DevOps enables.

I don't think that DevOps is just about developers, but it *is* about enabling developer productivity, which requires world-class infrastructure and operations skills. This goal is what led to NoOps—an unfortunate term, since it implies that Ops engineers will disappear, something I believe will never happen.

However, I do believe that the days when Ops can operate as a silo are going away. The same can be said about information security, compliance, and infrastructure in general. In this new era, the goal is not to interact with developers as adversaries or fellow sovereign states—instead, they act as fellow engineers, working together to achieve common business goals. Often it means creating platforms that developers use to do their work quickly, safely, and securely, without ever having to open up tickets and having work performed on their behalf.

Here's why I am certain Ops is so important: For two decades, I've self-identified primarily as an Ops person. This is despite being formally trained as a developer, having received my master's degree in computer science in 1995. I always gravitated to Ops because it's where I thought the real action was. But something changed about two years ago: I started self-identifying primarily as a developer. Without a doubt, this is because I learned the Clojure programming language.

It was one of the most difficult things I've ever learned: It's a functional programming language, which disallows mutation of state and encourages writing only pure functions. But I believe it's a safer and more productive way to build applications, and it's brought the joy of programming back into my life.

Here's the strange and unexpected thing that happened on this journey—I now hate dealing with infrastructure. It's so messy and unpredictable. I've become one of those developers, who just wants to live in a perfect little application bubble, despising having to deal with messy infrastructure.

And this is why I'm so convinced that the best days of infrastructure engineering are ahead of us. We need skilled engineers who can help ensure that developers can be truly productive, armed with the platforms that can help us build, test, secure, and deploy our code into production, without having to write custom scripts, manage security credentials, deal with logging, monitoring, connecting to databases, and so forth. These are the necessary things we need to do to create value in a messy and imperfect world, but they slow developers down.

That's why infrastructure is so important. And developer productivity isn't free. The reality is that most organisations are probably massively underinvesting in this area. Leading technology players invest heavily in their own technology.²⁵ In contrast, many traditional organisations do not. In an age where fast beats slow, these organisations are orders of magnitude slower than their peers.

But this is changing. The DevOps story in large, complex organisations is often one of rebellion, in which brave and courageous technologists seek to overthrow an ancient, powerful order—the conservative functional silos. My advice to top leadership is twofold: Identify engineering leaders who understand the value of the DevOps way, and partner them with passionate business leaders who want to reimagine how they create value. Get them together and empower them with a budget, autonomy, and authority. Magical things will happen.

RISK IMPLICATIONS

Many organisations may find daunting the cyber risks of working in a serverless and cloud computing environment. But there is a tremendous enterprise opportunity in leveraging automation to better protect itself against potential threats. Organisations should understand that, done properly, security protocols in a serverless environment supported by a NoOps operating model might significantly reduce cyber risk. Done poorly, it can accelerate cyber risk across the entire enterprise and at scale.

The key to overcoming the potential cyber risks associated with serverless environments is to shift your point of view—to see risk mitigation as an opportunity to develop and implement security and risk processes (or guardrails) within the code itself. Analyse the potential vulnerabilities in the code and the serverless environment to determine which threat vectors are most significant and most tolerable, then focus resources on protecting your most valuable assets and susceptible entry points. Embed security controls to detect and auto-respond to adverse events throughout your network and systems, as well as to automatically update configurations when new cyber risks are detected.

Where should an organisation start? First, it is crucial to understand how your network and infrastructure are designed, and identify the vulnerable points. This is no small task and should be given proper attention at the start of your serverless

In a serverless world, managing cyber risk requires daily revisions, tweaks, and tune-ups, but NoOps automation can make it easier.

journey. Applying the same approach that you've used during more traditional technology transformations is no longer sufficient: Logic, tools, and processes do not necessarily translate directly from your on-premises networks to serverless. Take the

example of reusing internal code and turning it into an external-facing API—without adding additional protections, you could accidentally expose your network to malicious attacks by using code that was never meant to be deployed in the hostile environment of the internet.

The good news is that cloud providers have built-in mechanisms that can be leveraged to enable strong authentication, proactive surveillance of the network, configuration monitoring, and more. To further shore up your defenses, seek to build security into various layers of your environment, throughout development, delivery, and operations—from the cloud platform management layer to the process and application layers. With your new automated ops capabilities, systems should be able to test for, detect, stop, and fix threats before they can affect your network, data, or reputation. It is worth noting that many organisations will continue to operate concurrently in the cloud and in traditional environments. They will face the additional challenge of maintaining the old controls and strategies, while designing and implementing new, very different controls within their modernised infrastructure.

There is no way to eliminate cyber risk altogether, so it is imperative that you reevaluate and redefine your risk tolerance in the light of serverless technology adoption. By leveraging the power of serverless technologies, security teams can deploy

solutions that will help contain and respond to threats, new and old, in a way previously not possible. Revisiting cyber risk events and proactively writing, tuning, and updating code to shield against newly discovered threats should be a regular process, not a quarterly or annual event. Gone are the days of setting

policy on paper and monitoring for, then managing, breaches after the fact. In a serverless world, managing cyber risk requires daily revisions, tweaks, and tune-ups, but NoOps automation can make it easier—and take active defense to a new level.

ARE YOU READY?

Like the macro cloud journey that many companies are currently on, any pursuit of NoOps or serverless can unfold in manageable stages. But where—and how—to begin? As you explore the NoOps in a serverless world trend's potential for your organisation, consider the following questions:

▶ **Is the trend right for me?**

At their core, the NoOps and serverless trends are grounded in two fundamental use cases: Infrastructure engineers can drive automation to new levels, and application developers can lessen their dependency on infrastructure engineers. It's also important to remember that while serverless won't be a good fit for every app in your stack, there's little downside to embracing automation and self-service to run and manage some of your solutions. Additionally, when done right, serverless architecture may yield faster time-to-market, more flexibility, a reduction in human error, and lower infrastructure and maintenance costs, all of which make good sense for the right workloads.

▶ **How do I go from doing this in small pockets to driving it across my organisation? How do I make movement at a tactical level?**

From a technical perspective, a serverless environment allows for faster and continuous scaling through automation, so the technology enables faster deployment across the enterprise. As the application load increases and more functions are executed, the cloud provider is responsible for scaling the underlying infrastructure. This may allow established organisations with monolithic legacy systems to stand up new capabilities as quickly as small startups. From an operational perspective, however, a NoOps environment requires a cultural shift in your organisation. You must be willing to break down silos, assign new roles, and reorganise your roster to gain the necessary traction to deploy at scale. Much like efforts to migrate to the cloud, a steering committee that establishes and enforces standards and lays out the road map—which at first glance may seem to counter the spirit of DevOps as well as NoOps—can keep the transformation on course.

▶ **What if I need to start from scratch? I'm not very far along with automation or DevOps.**

Serverless architectures may be your fastest way to embrace NoOps. With a serverless environment, software applications can be broken down into individual functions (that is, a microservices-based architecture) that are portable, cost-efficient, and, most importantly, not bound to a legacy infrastructure footprint. The separation of application functionality from supporting infrastructure provides the greatest opportunity for application modernisation.

▶ **What sort of workloads will be appropriate for serverless environments?**

A serverless approach is not one-size-fits-all but is often a good fit for applications that rely on microservices or APIs, such as Web applications, mobile backend, IoT backend, and real-time analytics and data processing. Applications well suited for serverless environments are ephemeral and stateless and don't require access to file-level systems. On the other hand, functions with high read-and-write volumes and those that require sustained computing power may be poor candidates. Longer-running, more complex computational tasks—such as data migration to NoSQL, applications requiring significant disk space or RAM, or those that require server-level operational access—may be better suited to a hybrid solution that employs servers and serverless capabilities.

▶ **Where do I start? Every company has several core, heart-and-lungs systems—do I begin there? Do I tackle it on the periphery first?**

Some companies are focusing their serverless efforts in areas where they have already made some progress on the digital front, such as customer-facing e-commerce applications and microservices. These areas are often ripe for the move to serverless because digital teams have probably begun the cultural shift (as well as some of the retraining and upskilling that may be necessary) that is an essential part of a NoOps transformation. As companies anchor their efforts in their digital foundations, they can simultaneously begin the NoOps and serverless transformation from both a top-down and bottom-up progression.

▶ **From an infrastructure perspective, what do I need to adopt? Do I have to go full cloud, or can I remain on-premises?**

You certainly can reap some benefits of DevOps practices on-premises, but unless you have a truly robust private cloud, your automation capabilities will likely be limited. And while you could deploy a hybrid solution of serverless and server-based components, you may realise only select serverless benefits. Even with just one on-premises server to manage, you will still perform anti-virus and vulnerability scanning and patching. To reach NoOps nirvana, you will likely need to go all in.

BOTTOM LINE

For years, basic care-and-feeding of critical systems claimed large portions of IT's budget and labor capacity. Today, the *NoOps in a serverless world* trend offers CIOs a way to redirect these precious resources away from operations and toward outcomes. It also offers development teams opportunities to learn new skills and work more independently. The journey from legacy internal servers to cloud-based compute, storage, and memory will not happen overnight. Nor will it be without unique challenges. But as more and more CIOs are realising, an opportunity to fundamentally transform IT from being reactive to proactive is just too good to ignore.



Contact



RICHARD REGAN

Senior Manager, Technology Consulting
Deloitte MCS Limited
+44 20 70072817
rregan@deloitte.co.uk

Authors

KEN CORLESS

Principal, Deloitte Consulting LLP

MIKE CAVIS

Managing Director, Cloud practice, Deloitte Consulting LLP

Risk implications

KIERAN NORTON

Principal, Deloitte LLP

Endnotes

1. Bill Briggs et al., *Follow the money: 2018 global CIO survey, chapter 3*, Deloitte Insights, August 8, 2018.
2. Ken Corless et al., *Reengineering technology: Building new IT delivery models from the top down and bottom up*, Deloitte Insights, December 5, 2017.
3. MarketsandMarkets, "Serverless architecture market worth \$14.93 billion by 2023," August 2018.
4. Cloud Foundry, "Where PaaS, containers and serverless stand in a multi-platform world," June 2018.
5. MarketsandMarkets, "Serverless architecture market worth \$14.93 billion by 2023."
6. Mayukh Nair, "How Netflix works: The (hugely simplified) complex stuff that happens every time you hit play," Medium, October 17, 2017.
7. John Demian, "Serverless case study: Coca-Cola," Dzone, July 10, 2018.
8. Keith Townsend, "NoOps: How serverless architecture introduces a third mode of IT operations," TechRepublic, January 19, 2018.
9. Bill Briggs et al., *2018 global CIO survey: Manifesting legacy*, Deloitte Insights, August 8, 2018.
10. Bill Kleyman, "Cloud automation: What you need to know and why it's important," *Data Center Knowledge*, November 3, 2014.
11. Himanshu Pant, "A brief history of serverless (or how I learned to stop worrying and start loving the cloud)," Medium, April 5, 2018.
12. MarketsandMarkets, "Serverless architecture market worth \$14.93 billion by 2023."
13. Yoav Leitersdorf et al., "Big opportunities in serverless computing," VentureBeat, October 27, 2017.
14. Scott Buchholz, Ashish Varan, and Gary Arora, "The many potential benefits of serverless computing," *WSJ CIO Journal*, November 9, 2017.
15. Digital Ocean, *Currents: A quarterly report on developer trends in the cloud*, June 2018.
16. Interview with Keith Narr, vice president, Cargill Digital Labs, October 18, 2018.
17. Paul Stamou, solution delivery manager, the Commonwealth Mutual Insurance Group, interviewed October 25, 2018.
18. Jennifer Baziuk, vice president, IT for Commonwealth, interviewed October 25, 2018.
19. Justin Davidson, Commonwealth solution architect, interviewed October 25, 2018.
20. Lynn Cox, senior vice president and network CIO at Verizon, interviewed on November 28, 2018.
21. "How Instagram co-founder Mike Krieger took its engineering org from 0 to 300 people," September 26, 2017.
22. Rachel Swatman, "Pokémon Go catches five new world records," Guinness World Records, August 10, 2016.
23. Bartie Scott, "What's next for the \$3.65 billion company behind Pokémon Go," Inc., November 18, 2016.
24. Stripe, "The developer coefficient," September 2018.
25. Christopher Mims, "Why do the biggest companies keep getting bigger? It's how they spend on tech," *Wall Street Journal*, July 26, 2018.