White Paper: Demystifying NoOps and Serverless Computing

Purpose

The purpose of this white paper is to provide an overview of NoOps and serverless computing for CIOs in the federal government and discuss the potential impact of these emerging trends on how agencies develop software and deliver services.

Background

Serverless computing is an emerging cloud computing model whereby cloud providers manage IT infrastructure and users are charged only for the resources they use.[1] The growth of serverless computing has given rise to the concept of NoOps, whereby service providers automate and manage IT infrastructure and operations to such an extent that IT operations staff can focus on higher-value work in the software development life cycle. NoOps is more of a theoretical concept than a development practice at this time, with few commercial IT organizations adopting a development approach that completely automates IT operations.[2]

The history of cloud computing is characterized by increasing levels of abstraction where responsibilities have shifted from consumers to service providers. Companies and government agencies have moved from managing their own on-premise and agency-owned equipment, to adopting Infrastructure-as-a-Service (IaaS) models that abstract physical hardware, to Platform-as-a-Service (PaaS) models that abstract operating systems. Companies are now experimenting with and implementing serverless computing architectures, which further abstract server and infrastructure management away from consumers and increase the speed to deploy microservices.[3] Serverless is similar to PaaS models in that in both, consumers do not have to worry about patching or monitoring individual servers. However, they differ in that with PaaS, consumers still have to choose their run-time environments and the size of their servers, while also having to pay for server capacity, even if that capacity is not fully utilized. With serverless, consumers no longer worry about servers and only pay for the computing resources required to run their code.[4]

---

[1] There are currently several vendors that offer serverless computing including: AWS Lambda, Google Cloud Functions, Microsoft Azure Functions, IBM/Apache OpenWhisk (open source), Oracle Cloud Fn (open source), iron.io, webtask.io.

[2] Adrian Cockcroft, former Cloud Architect at Netflix, claimed in his blog post titled “Ops, DevOps and PaaS (NoOps) at Netflix” that Netflix was operating under a NoOps paradigm. http://perfcap.blogspot.com/2012/03/ops-devops-and-noops-at-netflix.html

[3] Kroonenburg, Sam. The Next Layer of Abstraction in Cloud Computing is Serverless - read.acloud.guru/iaas-paas-serverless-the-next-big-deal-in-cloud-computing-34b8198c98a2

[4] Moss, Hannah. What is Serverless Computing - https://www.govloop.com/what-is-serverless-computing/

Serverless Computing

In serverless computing the management and allocation of resources, such as uptime, server maintenance, patching, backup, and security are all managed by cloud providers instead of systems administrators and IT operations staff.[5] In the serverless model, server management is outsourced to cloud providers and horizontal scaling is therefore completely automatic, elastic, and provider managed. Also, unlike traditional server-based applications, there is no need to constantly run servers so uptime occurs only when the serverless function is called to action.[6] This allows for a more dynamic, pay-as-you-go price model for compute, storage, and transmission capacity. In addition, it allows administrators and operations staff to focus on higher-value work, such as refactoring and integrating microservices into the architecture or increasing resiliency and security of the underlying business processes and data.

Serverless computing is enabled by two categories of cloud computing services:

- Function-as-a-Service (FaaS) allow users to develop, run, and manage applications without having to maintain or build infrastructure. Users upload and execute modular, event-driven functions that express discrete application logic, which allows automatic and independent scaling of functions, instead having to manage bulky, monolithic applications.[7]
- Backend-as-a-Service (BaaS) are backend, off-the-shelf, microservice products such as authentication, database management, remote updating, and cloud storage.[8]

Serverless computing raises the possibility that agencies will no longer have to worry as much about provisioning and managing infrastructure, hardware, or backend services and can focus increasingly on developing and delivering mission services and applications.

NoOps

The promise of serverless computing is that systems administration can become so automated and abstracted from the underlying infrastructure that there is no longer a need for systems administrators or IT operations teams to manage software, or operating systems, and these teams can focus on higher-value work. The growth of serverless computing has brought about the concept of No Operations (NoOps), whereby IT organizations can focus more on developing

[5] Han, Bowie. An Introduction to Serverless and FaaS (Functions as a Service) - https://medium.com/@BoweiHan/an-introduction-to-serverless-and-faas-functions-as-a-service-fb5cec0417b2

[6] Evolution of Serverless Functions - https://isometrik.io/evolution-serverless-functions/

[7] Watson, Matt. What is Function-as-a-Service? Serverless Architectures Are Here! - https://stackify.com/function-as-a-service-serverless-architecture/

[8] What is BaaS? Backend-as-a-Service vs. Serverless - https://www.cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/

new software features and leave the operations and management of underlying infrastructure to cloud service providers.

NoOps supports a more integrated delivery model of cross-functional expertise. Conventionally, developers built software features, while operations focused on the application's availability, reliability, performance, and security.[9] Accompanying growing levels of abstraction offered by cloud providers has been increasing demand for better and faster development processes and tools. In 2001, the Agile Manifesto kick-started the agile development movement and empowered small, cross-functional teams to build high-quality software faster.[10] In 2006, the first commercial cloud service providers launched publicly, which allowed development teams to outsource physical infrastructure entirely to cloud providers and no longer have to wait for hardware provisioning. And in 2009, John Allspaw and Paul Hammond set the groundwork for DevOps in their presentation titled "10+ Deploys Per Day: Dev and Ops Cooperation at Flickr".[11] DevOps is a set of standard operating procedures and practices designed to reduce barriers and friction between developers, operations, and other parts of the organization.[12] Some contend that NoOps is simply the next logical stage of software development evolution, whereby traditional IT operations is automated out of the development cycle. Others consider NoOps to be part of DevOps and contend that serverless computing requires closer collaboration and the blurring of traditional roles between development and operations.[13]

Regardless of how it's perceived, as an evolution of DevOps or something new entirely, the impact of NoOps on delivery is an emerging trend. The term "NoOps" is sometimes implied as the elimination of IT operations staff entirely. The problem with this conceptualization of NoOps is that it assumes IT operations can be reduced to a static set of procedural tasks, rather than a dynamic role that evolves in the face of changing technology and business requirements.[14] For instance, the emergence of edge computing and the Internet of Things (IoT) connected devices will increase the demand for IT operations and support, not eliminate it.[15] Leaders in government IT should be aware of the polarizing nature of the NoOps debate and how it relates to the broader issue of changing labor requirements for the federal workforce in the face of increasing automation. Because NoOps is a relatively new concept, the National Institute of Standards and Technology (NIST) does not have a technical definition of NoOps at this time.

---

[9] Rouse, Margaret. NoOps - https://searchcloudapplications.techtarget.com/definition/noops
[10] The Agile Manifesto - http://agilemanifesto.org/history.html
[11] Allspaw, John and Paul Hammond. 10+ Deploys Per Day: Dev and Ops Cooperation at Flickr - https://www.youtube.com/watch?v=LdOe18KhtT4
[12] Related concepts such as Continuous Integration (CI), Continuous Deployment (CD), Application Release Automation (ARA), DevSecOps, and others development methodologies attempt to increase development and production speed and quality.
[13] https://gist.github.com/jallspaw/2140086
[14] Redefining NoOps to better Inform IT Decision-Making - https://devops.com/redefining-noops-better-inform-decision-making/
[15] Greene, Travis. 7 Arguments against NoOps - https://techbeacon.com/enterprise-it/7-arguments-against-noops

Figure 1: Conventional Software Development Life Cycle
Conventionally, developers and IT operations staff had siloed functions, which caused friction and bottlenecks throughout the development life cycle, with each phase requiring its own isolated set of skills and processing,which imposed additional delays. Technological advancements in cloud computing has created a need for more agile and integrated development approaches.

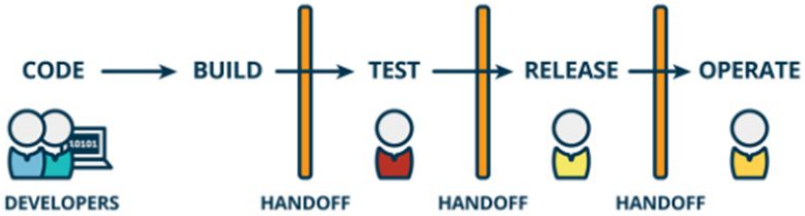
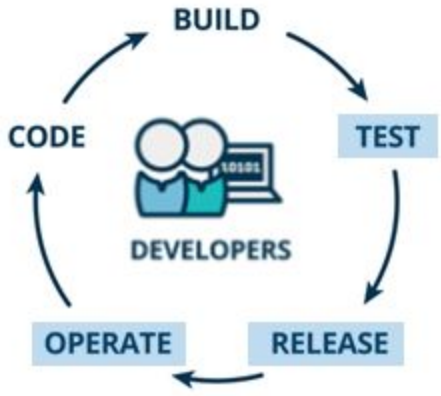


Figure 2: The DevOps Paradigm
Many organizations have adopted the DevOps principles of continuous development and collaboration between developers, operations, and other parts of the IT organization.Cross-functional DevOps teams and the automation of procedural tasks reduce the cycle time for delivery.

Figure 3: The NoOps Concept
Serverless computing allows delivery teams to automate many of the tasks traditionally isolated within IT operations, development, or operations staff silos. NoOps is a concept that could allow agencies to focus on delivering new features and not have to worry about isolated testing, release processes, or IT operations management.



Considerations for Government

The benefits and challenges of serverless computing and NoOps for federal agencies are highlighted below. Much of the direct benefits of serverless computing are increased speed and reduced costs to deploy new features and deliver business value. NoOps is more of a theoretical concept, with few examples of practical application in existing deployment models.

Benefits of Serverless Computing

Reduced Time to Deployment
By automating many IT operations tasks and allowing developers to code in a serverless environment, serverless computing can offer much faster development and deployment cycles than most government agencies or IT organization currently provide.

Reduced Costs
Serverless computing has the prospect of greatly reducing the cost of developing and deploying features because it reduces operational and development costs, lowers the cost to scale applications, and provides a more dynamic pay-as-you-go price model. Serverless computing models charge customers only for the resources used and have significant projected cost savings over other cloud models for many workloads.[16] In order to realize the speed advantages of serverless computing, agencies need innovative contracting approaches to ensure procurement does not become a process bottleneck.

Application Elasticity
Serverless computing allows applications to be elastic and automatically scale up to accommodate many concurrent users and scale down when traffic subsides. The cloud service provider manages infrastructure, which means services can scale faster than before.

Align Staff to Higher-Value Work
Serverless computing will likely change the role that IT operations staff traditionally play. Repurposing and retraining existing IT operations staff for more high-value work is both an area of opportunity and risk to IT organizations interested in implementing NoOps.[17] In a government context, serverless computing has the potential to enable agencies to realign staff to more high-value work, in line with the goals of the President's Management Agenda.[18]

---

[16] The Many Potential Benefits of Serverless Computing - https://deloitte.wsj.com/cio/2017/11/09/serverless-computings-many-potential-benefits/
[17] PMA CAP Goal 6: Shifting from Low-Value to High-Value Work - https://www.whitehouse.gov/wp-content/uploads/2018/04/ThePresidentsManagementAgenda.pdf
[18] President's Management Agenda. https://www.whitehouse.gov/wp-content/uploads/2018/03/Presidents-Management-Agenda.pdf

Challenges of Serverless Computing

Function Management
Serverless computing allows for the deployment of modular, event-driven functions instead of monolithic applications. However, as deployment units become smaller, the number of deployment units increase as well. The management of functions remains a challenge for many organizations implementing serverless computing, and this is where existing operations and administration staff could begin to focus their energy, and less on configuring servers.[19]

Monitoring and Security
Many monitoring and security tools work at the OS or the VM level. This isn't an option with serverless computing. Serverless requires IT operations staff to monitor performance at the application level, which may be a new paradigm for some government agencies.

Vendor Lock-In
Currently, the majority of serverless solutions rely on vendor's cloud-specific services. This means that if an agency builds an application on one vendor's serverless computing infrastructure, it would be difficult to transfer that work to another vendor without additional re-engineering.[20] With serverless, the cloud service provider controls the underlying infrastructure, which means agencies have less ability to customize or optimize infrastructure to meet the unique needs of the agency. However, there are several multi-cloud serverless platforms that make it easier to move applications between vendors.[21]

Migration at Scale
For large corporations and agencies alike, migrating and adopting new serverless computing environments is a monumental challenge. Such a shift requires new contract vehicles, re-architecting and re-engineering existing applications, and large-scale change management strategies. Federal agencies should leverage best practices and lessons learned from industry partners that migrate to serverless computing environments at large scale before undergoing similar undertakings.[22]

Culture Change
NoOps will require large-scale cultural changes for most organizations. NoOps proposes to fundamentally alter how software development is performed. To implement NoOps with serverless computing environments, federal agencies will have to invest in the reskilling of existing IT staff and change governance and acquisitions approaches. Agencies should focus on training acquisitions staff to be able to procure more dynamic, pay-as-you-go serverless computing services, upskilling IT operations staff to be able to support emerging technologies

[19] Kanowitz, Stephanie. When serverless makes sense - https://gcn.com/Articles/2019/03/12/serverless-computing.aspx
[20] The Many Potential Benefits of Serverless Computing - https://deloitte.wsj.com/cio/2017/11/09/serverless-computings-many-potential-benefits/
[21] Subramanian, Krishnan. On Serverless and Lock-In - https://stacksense.io/krishnan/philosophy/on-serverless-and-lock-in/
[22] Case studies of companies using AWS Lambda - https://aws.amazon.com/lambda/

such as edge computing and IoT devices, and ensuring developers have the requisite skills to be able to deploy on the serverless computing platforms.

Agency Serverless Use Case

The National Science Foundation's (NSF) Office of Integrated Activities (OIA) needed to acquire and provision scientific bibliometric data, including citations and publication information, from Clarivate Analytics[23] on a weekly basis. Early in the planning phase for this project, NSF leadership included serverless computing architectures as a possible solution within their evaluation framework. Based on several factors, including that data was pulled at regularly scheduled intervals and the system did not have to constantly pull data, NSF was already along their AWS journey, and the relatively low security risks associated with working with publicly-available data, NSF determined that a serverless solution using AWS Lambda[24] was the best option to fulfill their requirements. Furthermore, NSF previously centralized its IT operations, which streamlined governance and decision making throughout the process. NSF leveraged AWS Lambda to connect and move data from Clarivate Web of Science Databases into an Amazon Simple Storage Service (S3) bucket, then used AWS Command Line Interface (CLI) to securely move data from S3 to NSF's data warehouse. Using Lambda allowed NSF to reduce O&M overheads and allowed on-demand execution of data pulls. Additionally, NSF mentioned several use cases in which serverlesss would not have been the optimal solution including for managing large authenticated web applications that require continuous monitoring, patching and updating, as well as applications dealing with more sensitive information than Clarivate's publicly-facing data.[25]

Additional Considerations for Agencies

ATO Process and FedRAMP approved serverless offerings

The Authority to Operate (ATO) process has been a persistent hurdle for cloud adoption and IT modernization because it is a slow and burdensome regulatory framework that vendors have to navigate in order to offer new services to agencies. Although there has been recent progress to decrease ATO process times,[26] the process remains a barrier for agencies to procure cloud services. Several cloud service providers have their serverless offerings FedRAMP approved,[27]

[23] Clarivate is an external vendor that provides scientific citation indexing and other services. More information can be found at: https://clarivate.com/

[24] More information on AWS Lambda can be found at:https://aws.amazon.com/lambda/

[25] This use case was developed in discussions with senior members of NSF's Office of Integrated Activities (OIA) within the Division of Information Systems (DIS).

[26] Cordell, Carten. 18F Slices ATO Times from 6 Months to 30 Days. https://www.fedscoop.com/18f-slices-ato-times-6-months-30-days/

[27] AWS Lambda, Google Functions, and Microsoft Azure Functions are all listed as FedRAMP approved services on each company's compliance website: https://aws.amazon.com/compliance/services-in-scope/;

which enables agencies and federal contractors to move more quickly through the authorization process to achieve an ATO.[28]

Zero Trust Architecture and Identity and Access Management
Traditional perimeter-based security approaches are being tested as agencies continue to expand mobile and cloud-enabled environments, including serverless computing. Zero Trust is a new security paradigm that has the potential to substantially change and improve the ability of agencies to protect their systems and data.[29] Zero Trust is a cybersecurity approach that assumes threats are already on an agency's network, rather than relying on traditional perimeter-based security to keep intruders out. Therefore, users are constantly authenticated, which blocks threats already inside the network from moving laterally and infecting the entire system. Furthermore, security responsibilities shift from the user to the cloud service provider, which requires agencies have trust in the vendors' ability to handle security and risks. One possible solution is to adopt a DevSecOps approach to application development, whereby agencies' security teams are intimately involved in the development and deployment of code and features.

Serverless computing requires agencies reduce security risks by managing identity and access privileges.[30] Per M-19-17, "the interwoven architecture of the Federal Government creates complexity in managing access to resources, safeguarding network, and protecting information. While hardening the perimeter is important, agencies must shift from simply managing access inside and outside of the perimeter to using identity as the underpinning for managing the risk posed by attempts to access Federal resources made by users and information systems."[31] For agencies exploring serverless computing, comprehensive identity and access management combined with a Zero Trust security approach are critical to securing agency IT systems.

Suggested Actions for Agencies

- Include serverless computing as a consideration for your agency's evaluation framework. Applications that do not require constant running, that have lower security demands controls, and for which speed to deploy is paramount to business or mission delivery are good candidates for serverless computing. NSF's AWS Lambda use case that pulled

https://cloud.google.com/security/compliance/fedramp/; and https://www.microsoft.com/en-us/trustcenter/compliance/fedramp

[28] Barr, Jeff. AWS FedRAMP ATO: Difficult to Achieve, Easily Misunderstood, Valuable to All AWS Customers. https://aws.amazon.com/blogs/aws/aws-fedramp-ato-difficult-to-achieve-easily-misunderstood-valuable-to-all-aws-customers/

[29] American Council for Technology Industry Advisory Council (ACT-IAC). Zero Trust Cybersecurity Current Trends. 18 April 2019. https://www.actiac.org/system/files/ACT-IAC%20Zero%20Trust%20Project%20Report%20041820 19.pdf

[30] Kerner, Sean Michael. Serverless Cloud Security: How to Secure Serverless Computing. https://www.esecurityplanet.com/cloud/serverless-cloud-security.html

[31] M-19-17, Enabling Mission Delivery through Improved Identity, Credential, and Access Management. https://www.whitehouse.gov/wp-content/uploads/2019/05/M-19-17.pdf

scientific bibliometric data from Clarivate Analytics on a weekly basis, is an example of a low-risk serverless application not required to be always running.

- Work with development and operations teams within your agency to pilot serverless computing. In the NSF use case provided above, NSF went through several iterations before finalizing their process. This included close coordination with senior IT team leaders, integration of security teams from the beginning of the process, and collaboration with data warehousing teams, desktops teams, and the engineering review board.
- Work with security teams to understand ATO requirements for running serverless computing on your agency's systems. NSF engaged their security teams early in their adoption process.

Acknowledgements

This White Paper was produced by the CIO Council's Innovation Committee and would not have been possible without contributions from the Office of Management and Budget's Office of the Federal Chief Information Officer, the General Services Administration's Office of Government-wide Policy, and support from REI Systems, Inc. and Incapsulate, LLC.

Originally established by Executive Order 13011 (Federal Information Technology) and later codified by the E-Government Act of 2002, the CIO Council is the principal interagency forum for improving agency practices related to the design, acquisition, development, modernization, use, sharing, and performance of Federal information resources.

The CIO Council is one element of an interagency support structure established to achieve information resource management objectives delineated in legislation including the E-Government Act of 2002, Government Paperwork Elimination Act, Paperwork Reduction Act, Government Performance and Results Act, and the Information Technology Management Reform Act of 1996.

FOR MORE INFORMATION

Contact feedback@cio.gov
visit www.cio.gov