

Why the Future of Open Hybrid Cloud Should Include a Serverless Approach

MARKET TRENDS REPORT



Introduction

As priorities and requirements change, agencies are looking for ways to adapt their mission needs and requirements. This has motivated many agencies to adopt more modern, cloud-friendly application development methods including Agile development, DevOps, containers and microservices.

These modern development practices have gone a long way toward speeding up development, reducing application downtime and increasing overall productivity. They are ideal for many uses, such as complex applications that must be highly scalable and large applications with short release cycles. But they aren't the best choice for every scenario. They can fall short of the mark for quick-turnaround requests and situations where the development infrastructure sits idle for long periods of time, even though agencies continue paying for the resources.

Agencies are discovering a better way to address these issues: the serverless approach. This application development method — which actually does use servers on the back end — allows agencies to execute cloud-based development environments on demand, triggered by specific events. That way, developers have the resources they need to complete their tasks quickly and efficiently. Depending on the situation, serverless application development can be faster and much more cost-effective and efficient.

To learn more about how agencies can get the most benefit out of a serverless architecture, GovLoop teamed with Red Hat, which provides an open platform for developing and deploying event-driven applications in an open, hybrid cloud environment. This report will discuss when and how a serverless approach can help agencies improve application development.

By the Numbers

1.7 million

developers worldwide use Kubernetes

47%

of organizations currently use serverless technologies

48%

of development professionals believe serverless is the future of cloud

The top 3 functions for which organizations currently turn to serverless computing are:

1. Web apps
 2. Business logic
 3. Database changes
-

26%

is the amount serverless computing is expected to grow in the next 5 years

46%

of organizations use or plan to use serverless architecture in the next 12 months

60%

is how much serverless architectures can reduce costs

8 to 10

hours are typically spent by an application developer per month on application provisioning, security implementation and patching, and OS updates

“Serverless computing allows development teams to devote more time to core products and allows them to build scalable, reliable systems more quickly and easily than when using server-based architectures”

Mike Deck,
Principal Solutions Architect at Amazon Web Services

Improved Agility Through Serverless Computing

The Challenge: Inefficiencies Bog Down Applications

Cloud Smart, digital modernization and a push toward digital services in general have prompted most agencies to adopt some form of Agile software development. Often, it is microservices-based, which breaks applications down to very small components for development. This approach has gone a long way toward making government more efficient, but in some cases, application development still takes too long, and it's too expensive.

When tasks must execute quickly and change often, the microservices model may not be efficient enough. For example, a task that requires fast data aggregation or analysis and will execute for only a few seconds may be hampered by the thousands of functions making up that application that surrounds it. The same is true of scalability. Applications that need to handle large increases in the number of requests, such as apps with seasonal spikes, face similar issues. For example, an image processing app could include a microservice that would allow it to scale to handle the load, but it could be bogged down by other functions, because in the microservices model, all functions stand by, ready to execute at all times.

That challenge — being bogged down by other processes running at all times — translates into dollars. Because microservices are on standby waiting to be activated, resources and infrastructure are constantly in use, costing money both in terms of the host servers they reside on and the care and feeding of those resources.

The Solution: An Event-Driven Approach to Service Delivery

Depending on the use case, serverless application development can improve efficiency and agility while reducing cost.

While there are still servers in the serverless model, developers don't have to manage them; cloud providers do. And unlike the microservices approach, which breaks

everything into individual components, the serverless approach breaks things down even further. In fact, one microservice can have several serverless functions. The serverless approach also allows those functions to occur only when needed, reducing infrastructure needs and cost.

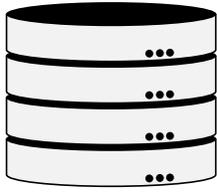
“Serverless development has the unique capability of being able to turn your software off by using a mechanism to make the code only wake up and run when it's needed,” explained Jason Dudash, a Principal Solutions Architect at Red Hat. “If there are no events coming to trigger that code, then the code just lies dormant. That means you can use your infrastructure more effectively because you aren't running something that's idle.”

That event-driven approach is what can make serverless application development extremely agile. Instead of standing ready to work, it springs into action only when events trigger those actions.

For example, if a healthcare-related agency wanted to develop a mobile app that can analyze skin blemishes to determine whether they are cancerous, it might use this approach. A patient could use that app to upload a picture of a mole for analysis. Uploading the photo is an event, and it would trigger the function of sending the photo to some type of cloud-native storage, along with a function that would strip the data of all personally identifiable information (PII). In turn, that would trigger a notification that the data exists, which would trigger functions that would run the algorithm, form a theory and return results to the user.

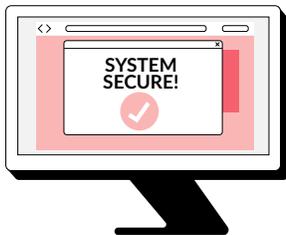
These types of efficiencies do more than increase agility. By reducing the time needed for expensive resources to perform tasks, they can reduce costs, and by using a single application programming interface (API) to perform multiple tasks — such as telling an application how to scale and how it should be exposed to the outside world — they can accelerate the entire development process.

Best Practices in Serverless Computing



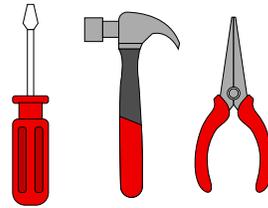
If you have sensitive data or applications that can't leave your premises, make sure the solution you choose can accommodate that.

Many serverless solutions run only in the cloud. Although that's fine for many applications, it can be a real problem for agencies with sensitive information that must remain within the on-premise environment. In that case, it's important to look for a platform that allows you to run serverless workloads on premises the same way it works in the cloud, using the same API.



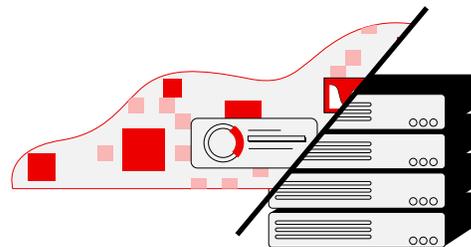
Build your serverless apps on a secure base.

In addition to taking security into account at every step of the development process — building security around each function, securing application dependencies, and running the runtime that executes the serverless code in a secure place — make sure that you are building on top of a secure framework, including trusted containers. “When you are writing your code, you don't want to have to worry about patching the operating system or version of the runtime environment. You just want to write your functions,” said Chris Kang, a Staff Solutions Architect at Red Hat.



Don't reinvent the wheel.

It's possible to go it alone with serverless development, but there are plenty of reasons to consider using a platform that can take much of the guesswork out of the process. Choose a platform that enables your developers to more easily build software that aligns to an event-driven architecture. The platform should provide ready-to-use APIs, for example, that will allow developers to connect functions smoothly. It should also handle as much of the business logic as possible and provide easy connectors for distributed services.



Remember, not every workload is a good candidate for serverless development.

There are many valid use cases for serverless development, including backend APIs; web, mobile and Internet of Things applications; and process automation. Applications with variable capacity and high scalability requirements are particularly well-suited, but not everything is a good fit. It's not ideal for extremely performance-sensitive workloads that require real-time capability or for applications with complicated computational tasks, data migrations or long-running batch jobs.

Case study: Research Agency Uses Serverless to Pinpoint Study Candidates

A federal healthcare research organization chose the serverless approach to build a service that analyzes chest X-rays for risks of pneumonia and routes good research candidates to the research team.

To prepare, the team first assembled and prepared its tools: a Ceph cluster for storage, an Apache Kafka cluster for distributing data in real time and Knative to create safe storage for the access key and bucket keys, along with a Knative service that will use the container image of the function to perform image processing. All of this runs on Red Hat's OpenShift Serverless platform. Knative is an open source community project that allows developers to deploy, run and manage serverless containers on Kubernetes.

The service launches when a new X-ray image is uploaded to a Ceph bucket. The bucket is configured with a notification feature that the image upload triggers. This

Knative event triggers a Knative Eventing function that retrieves the image from the bucket and uses an image-recognition algorithm to detect if there is a risk of pneumonia, assess the possibility of the risk in percentage form, and print that number over the image.

At that point, the process can go in two directions. If the risk associated with the image is less than 80%, it is immediately saved in a specific repository for clinic use. If the risk is greater than 80%, the processed image is then anonymized by blurring some data inside the image and changing the image's name, and served to another bucket for use by the research team. When the process is finished, image processing containers stop.

This example shows how the serverless approach can help organizations quickly and efficiently build automated data pipelines without additional infrastructure or tooling.

HOW RED HAT HELPS

Red Hat's solutions for hybrid cloud environments are built with open source code while pushing the boundaries of what's possible. Red Hat's OpenShift development platform, used by countless federal agencies and departments, is an important building block for all types of application development. OpenShift Serverless provides developers with a production-ready platform for serverless development, helping simplify the process of delivering code from development to production.

OpenShift Serverless encompasses microservices, functions and applications, and events, relying on containers and taking advantage of cutting-edge open source tools including Knative for deploying, running and managing serverless, cloud-native applications to Kubernetes; service mesh technology; and event-driven auto-scaling for containers. With this combination of technologies, developers can more easily and quickly build modern, scalable serverless applications while also supporting legacy applications.

Learn more: www.redhat.com/government

Conclusion

Agencies looking for ways to increase productivity and efficiency are naturally turning to more agile development methods, including microservices and serverless development. Both have a place in the developer’s arsenal. It’s just a matter of knowing when to use each method.

While the microservices approach is ideal for time-consuming data processing, reliable response times and always-on services, the serverless approach is more suited to routine tasks such as database access, applications with unpredictable spikes and applications that must be updated or changed quickly.

The best approach may be to have both options available. That way, developers can pick the best method for the job, or even combine the two. In either case, it’s important to find a platform that makes the job as simple, automated and secure as possible.



ABOUT RED HAT

The adoption of open principles helps the U.S. government start, accelerate, and improve the art of digital transformation—people, process, and technology. As the world’s leading provider of enterprise open source solutions, Red Hat uses a community-powered approach to deliver reliable and high-performing Linux , hybrid cloud, container, and Kubernetes technologies. Red Hat helps customers integrate new and existing IT applications, develop cloud-native applications, standardize on our industry-leading operating system, and automate, secure, and manage complex environments. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500 and 100% of U.S. executive departments. As a strategic partner to cloud providers, systems integrators, application vendors, customers, and open source communities, Red Hat can help organizations prepare for the digital future.

<https://www.redhat.com/government>



ABOUT GOVLOOP

GovLoop’s mission is to “connect government to improve government.” We aim to inspire public-sector professionals by serving as the knowledge network for government. GovLoop connects more than 300,000 members, fostering cross-government collaboration, solving common problems and advancing government careers. GovLoop is headquartered in Washington, D.C., with a team of dedicated professionals who share a commitment to connect and improve government.

For more information about this report, please reach out to info@govloop.com.



1152 15th St. NW Suite 800
Washington, DC 20005

P: (202) 407-7421 | F: (202) 407-7501

www.govloop.com
@GovLoop