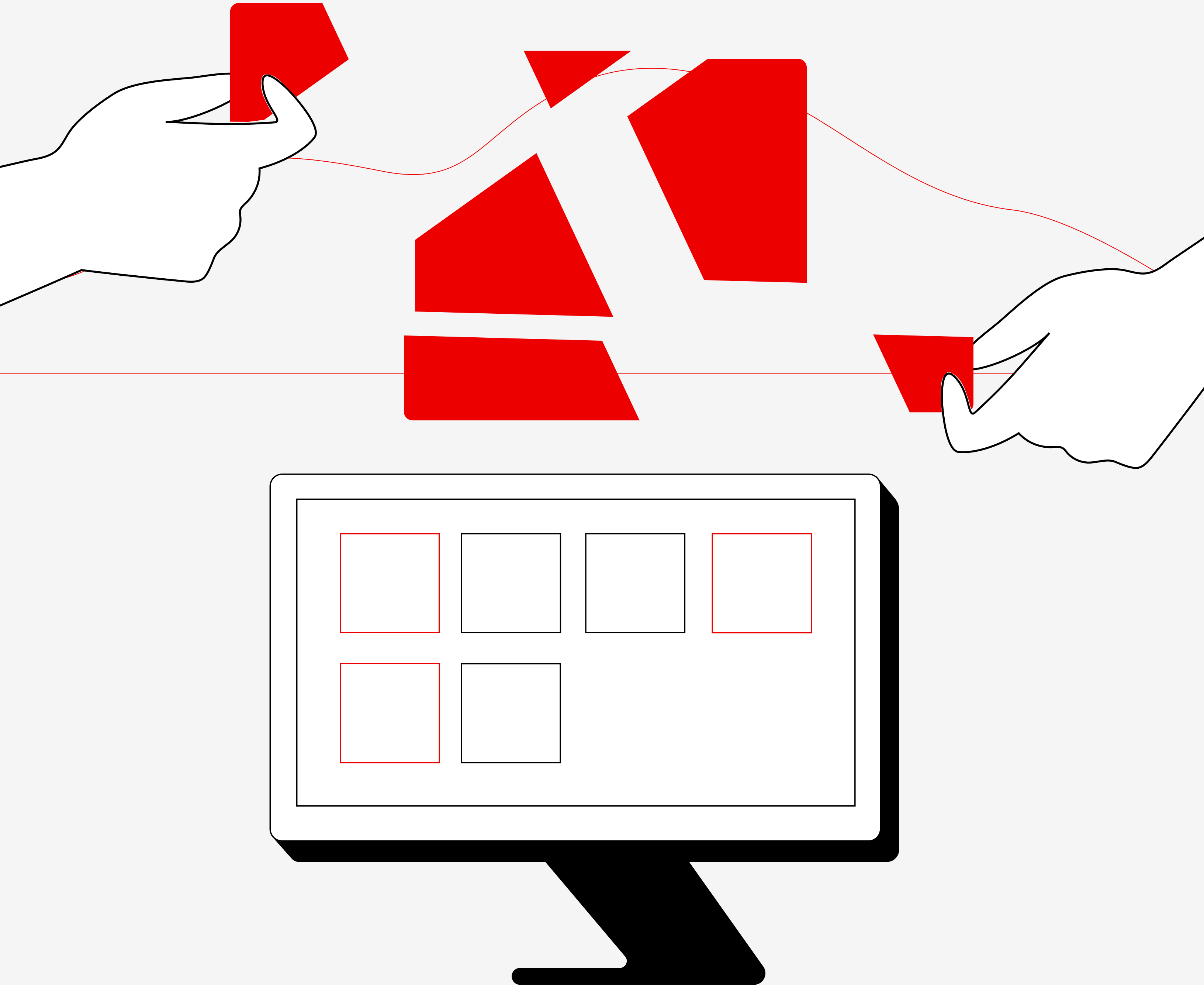


Getting the Most out of Microservices

MARKET TRENDS REPORT

govloop Red Hat

Introduction

Agencies today still run many older applications, often developed on a single codebase and running on outdated technology or infrastructure. That made sense a decade ago when client/server computing and desktop technology were the backbones of most agencies, but it's less effective today. With the growth of cloud computing, more ingenious cyberthreats, and the need for fast and easily changeable applications, agencies are realizing that the traditional method of application development doesn't serve their needs.

For more agencies, the answer is microservices. This application development approach breaks applications into smaller components for development. As a result, agencies can more easily and quickly develop cloud-ready applications, and developers can share similar processes across multiple applications, update applications without affecting other parts of the codebase and take advantage of cloud's inherent capabilities.

While microservices are becoming an important part of the application fabric governmentwide, the process also can make visibility, manageability and security more challenging. One of the most effective ways to address those issues is by implementing a service mesh — a layer of infrastructure that manages microservices without requiring changes to application code. This can result in better traffic control and load balancing, increased transparency among microservices, improved service-to-service communication, reduced downtime, and better security.

To learn more about how agencies can get the most from microservices, GovLoop teamed with Red Hat, which provides an open platform for running microservices on a cloud-based platform. This report will discuss the challenges with effective microservices development and management, and how to best address them.

By the Numbers

92%

of software engineers report at least some success with their microservices projects

62%

of software engineers use containers to deploy at least some of their microservices

40%

of microservices adopters say culture or mindset are the biggest barriers to microservices adoption

83%

of organizations rely on open source software to become more agile

62%

of IT decision-makers say they expect containers to be mainstreamed at their organizations within one year

35%

of all production apps will be cloud-native by 2022

90%

of new apps will feature microservices architectures by 2022

"Re-platforming to cloud-native is....an imperative for digital transformation strategies, and it will sweep through the market over the next decade, much like the re-platforming to the internet and web in the 1990s and 2000s."

How to Overcome the Complexities of Microservices

The Challenge: Providing Visibility and Security

Microservices are individual services that together, make up an application. It's not unusual for one application to consist of hundreds of microservices, all of which must be continually discovered and tracked, along with their dependencies, performance and status. Without the ability to see what is happening at all times, it is very difficult for developers to resolve issues as they occur. For example, a failure in one microservice can cause cascading failures in other parts of the application. The longer it takes to identify the source of the failure, the longer a service will be disrupted.

Take the simple example of a payment service performing slower than usual. Getting to the bottom of the situation requires a deep understanding and tracing of all components. For example, if a developer rolls out a new version of the payment system and notices that the end-to-end request now takes 1.5 seconds instead of the previous 700 milliseconds, that might prompt them to ask the payment team to look at the change it made. The payment team might not find anything wrong with the code. The next step would be asking each team that touches the request — the front-end team, the analytics team and all of the others — to check their code.

Microservices also can complicate security. Unlike a traditional application, which has very few entry points, applications built with microservices have dozens or hundreds of entry points. Each exposed application programming interface (API), port and component is a potential attack vector, and every entry point must have the appropriate access controls to protect against denial-of-service attacks, man-in-the-middle attacks and so on.

The Solution: The Service Mesh

Adding a service mesh to microservices can improve visibility, monitoring, management and security.

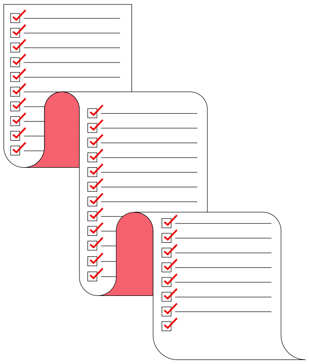
A service mesh allows developers to make changes without touching the application code itself. It provides the ability to mirror and monitor traffic on multiple versions of the same service, which lets developers test capabilities before deployment and determine the best way to route traffic through the system for specific types of use patterns. Most importantly, it provides automated ways of monitoring what is happening between services at all times, providing important metrics that can help quickly determine the cause of failures or performance issues.

"Microservices tend to get a lot of bloat from all of the things you have to add on to make sure it is secure and can handle different types of failures," explained Jason Dudash, Principal Solutions Architect at Red Hat. "The service mesh builds that capability into the platform, so it can handle these things without the developer having to import libraries or make source code changes."

A service mesh also can drastically improve security in microservices-based development through authentication and authorization. When services talk to one another, each must ensure that the others are who they say they are. Without a security layer, your microservices are essentially talking over an unencrypted open channel.

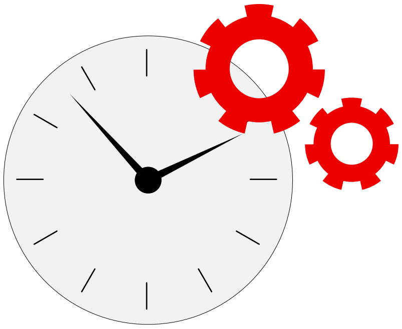
Authorization is equally important. In an application, there are certain services that should talk to one another, and others that should not. For example, although a checkout service talks to a payment service, the front-end service should not talk directly to the payment service. Maintaining secure authorization is difficult if done manually, but when incorporated into a service mesh, the process can be customized and automated.

Best Practices in Microservices



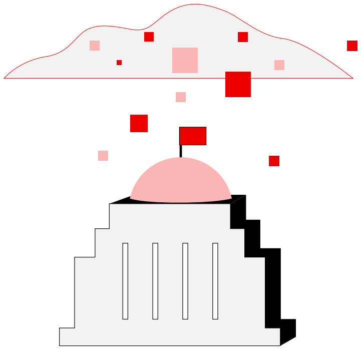
Make sure you have a reason and a plan for microservices.

Don't just jump on the microservices bandwagon because it's popular, said Chris Kang, Staff Solutions Architect at Red Hat. "It's important to plan around the measurable outcomes your agency is trying to achieve, and to define the business capabilities that you want your microservices to achieve," he said. Kang also stressed the importance of tying a microservice to people and processes to make that microservice a success.



Choose metrics that matter.

The best way to improve microservices over time is by measuring the metrics that are most important to your agency. Some of the most popular are tracking deployment frequency, which indicates agility or delivery speed; measuring change failure rate, or the percentage of deployments that must be rolled back because they failed in production; and how long it takes to get a change into production, or how many times you can release in a day. Those types of specifics give you a good idea of how well you are succeeding, and what needs improvement.



Insist on an open, cloud-based container platform for running microservices.

In addition to avoiding vendor lock-in, relying on an open source approach can reduce the risk of incompatible code, obsolescence and problematic upgrades, while improving developer productivity. At the same time, the flexibility and broad reach of an open cloud-based platform help developers deploy consistently in hybrid or multi-cloud environments.

"As microservices-based applications are increasingly deployed within large enterprises and cloud-based environments, dedicated and scalable infrastructure is needed to support a comprehensive set of security services. This infrastructure is called the Service Mesh, which can support authentication, authorization, secure service discovery, secure communication, security monitoring, as well as other security services."

NIST Publication SP 800-204A,
"Building Secure Microservices-based Applications Using Service-Mesh Architecture"

Case study:
The Air Force is Betting Big on Microservices

With a mission to improve the software development process, the Air Force is increasingly depending on more Agile development approaches that involve containers, microservices and other open source cloud tools.

The result is the Defense Department (DoD) Enterprise DevSecOps Initiative, a joint initiative of the Office of the Under Secretary of Defense, the DoD chief information officer, the Defense Information Systems Agency and the military services. It lays out the importance of relying on secure, hardened containers, with a focus on avoiding vendor lock-in. The team selected Kubernetes for that reason, along with technologies such as Istio providing security for the networking layer of the DoD stack and Knative for serverless development, said Air Force Chief Software Officer Nicolas Chaillan.

The application layer of the development stack allows teams to build microservices using hardened containers, while the service mesh layer provides baked-in security and microservices architecture enablement. The continuous integration/continuous delivery layer uses DoD-approved containers, while the platform layer uses Cloud Native Computing Foundation-compliant Kubernetes, and the infrastructure layer is environment-agnostic.

According to Chaillan, managing microservices with the help of a service mesh will improve API management, service discovery, dynamic request routing, gradual rollouts and Layer 7 load balancing, while enforcing a zero-trust model down to the container level.

Speed is equally important. With these capabilities, Chaillan said DoD can deploy software in production within about 28 hours of deciding to develop software. “That’s something we would never have been able to do with DevSecOps, at least not without taking significant risk when it comes to cybersecurity, which in this case it’s baked in into the stack,” he said.

HOW RED HAT HELPS

Development teams struggle with building, debugging and connecting services properly, while application operations teams face increasing challenges with hybrid deployments, scaling bottlenecks, recovering from failure and gathering metrics. Red Hat’s OpenShift Service Mesh, specifically designed to simplify the development, deployment and management of microservices-based applications on OpenShift, addresses these issues.

OpenShift Service Mesh provides a uniform way of connecting, managing and observing microservices-based applications by providing behavior insight into and control of the networked microservices in a service mesh. As a result, agencies can more effectively connect, secure, control and observe their microservice-based applications, while helping improve modernization outcomes.

Learn more: www.redhat.com/government

Conclusion

Whether the motivator is modernization, a shift to cloud computing or increased productivity, the result is often the same: moving to the microservices application development approach. This application development method, which breaks out smaller units of an application's function for development, can deliver on many of these promises, but it's not a set-it-and-forget-it option. Getting the best value out of microservices while ensuring security and visibility requires the right processes. With a service mesh approach to microservices, agencies can get the value they need without being stopped short by performance or security issues.

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.

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.

govloop

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

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

www.govloop.com
@GovLoop