Integrating Security From End to End With DevSecOps

GOVLOOP
E-BOOK
2019

Executive Summary

Demand for speed and innovation is everywhere in government these days. But now more than ever, in the face of rising and sophisticated cyberattacks, governments at all levels need to be integrating security into their application development and digital services.

This can be difficult, given that security is often perceived as slowing down innovation. In the past, the role of security was isolated to a specific team in the final stage of development. That wasn't as problematic when development cycles lasted months or even years, but those days are long over – and the days of increased cyberattacks on government are here.

In **Integrating Security From End to End With DevSecOps**, we explain how leading governments are working to integrate security into their DevOps practices and culture, ensuring that public sector innovation can be delivered securely – and creating an evolved approach called DevSecOps.

Today, in the collaborative framework of DevOps, security is a shared responsibility integrated from end to end. It's a mindset that is so important, it has led to this term and approach, DevSecOps, to emphasize the need to build a security foundation into DevOps initiatives. DevSecOps means thinking about application and infrastructure security from the start. It also means automating some security gates to keep the DevOps workflow from slowing down.

There are new, innovative technology solutions like containers, automation and cloud computing that can help your agency meet its security goals. But effective DevOps security requires more than new tools — it builds on the cultural changes of DevOps to integrate the work of security teams sooner rather than later.

In this ebook, we'll explore how DevSecOps highlights the need to invite security teams at the outset of DevOps initiatives to build in information security and set a plan for security automation. We'll also look at how it underscores the need to help developers code with security in mind, a process that involves security teams sharing visibility, feedback and insights on known threats.

Finally, we'll hear from experts in government who are using these approaches, and explore steps you and your team can take to fully leverage DevSecOps in your IT and development journey.

DevSecOps at a Glance

What impact does DevSecOps have on government? How does it help you achieve your agency's mission, and where did it evolve from? These stats will help set the context for why DevSecOps is more important than ever for governments at all levels.

30x

Today's high-performing IT teams using DevOps practices deploy code up to 30 times faster, experience 60% fewer failures and recover from development issues 168 times faster than their peers.

DevSecOps is "a cultural and engineering practice that breaks down barriers and opens collaboration between development, security and operations organizations using automation," according to the GSA.

338%

Mature DevOps practices are 338% more likely to integrate automated security.

75%

of CIOs identified DevOps as a top priority.

48%

of developers don't have enough time to spend on security.

73%

Of organizations surveyed, 33% suffered verified breaches stemming from vulnerabilities in open source components or web applications within the last 12 months. These and other high-profile breaches led 73% of respondents to affirm an increased interest in DevSecOps practices, including increased investment and implementation.

56%

of respondents with a mature DevOps practice rated container and application security tools as critical to their organizations.

"Something that helped the adoption of DevSecOps was to increase collaboration to create an environment of sharing."

- Darryl Peek, Director of Digital, Innovation and Solutions, Department of Homeland Security

5.9 billion

The DevSecOps market is expected to expand at a compound annual growth rate of 31.2% from 1.5 billion in 2018 to 5.9 billion 2023.

The How and Why of DevSecOps

The Definition of DevSecOps

Managing a seamless and secure IT enterprise is no small task in today's complex environment.

For starters, your agency relies on a host of systems and applications to meet daily demands from internal and external customers – everything from online services, to email, to ticket services, web applications and more. Ensuring that those systems are updated with the latest code, operating smoothly and running securely requires a joint effort across multiple teams.

But those teams' varying priorities can clash at times. Developers work to push code that corrects glitches, providing user enhancements and fixing software vulnerabilities. The IT operations team keeps these systems running and functional for the hundreds or thousands of people who depend on them. And equally important is the security team that must ensure the same systems are secure, up to date and compliant with federal standards.

To bridge the divide between development, operations and security teams and ensure that systems stay updated, running and secure all at the same time, agencies are investing in a new approach known as DevSecOps.

At its core, DevSecOps is **"a cultural and engineering practice that breaks down barriers and opens collaboration between development, security and operations organizations using automation,"** according to the General Services Administration's definition. The focus is on rapid, frequent delivery of secure infrastructure and software to production, which a growing number of agencies are prioritizing.

The purpose and intent of DevSecOps is to build on the mindset that everyone is responsible for security with the goal of safely distributing security decisions at speed and scale.

The Evolution to DevSecOps

Many organizations have, since the early 2000s, adopted agile methodologies for iterative, incremental and evolutionary software development.

Agile software development refers to software development methodologies centered on the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

Agile, in many instances, has evolved into a DevOps approach.

The word "DevOps" is a mashup of "development' and "operations" but it represents a set of ideas and practices much larger than those two terms alone, or together. DevOps includes collaborative ways of working, data analytics and many other things. It describes approaches to speeding up the processes by which an idea (like a new software feature, a request for enhancement or a bug fix) goes from development to deployment in a production environment where it can provide value to the user.

These approaches require that development teams and operations teams communicate frequently and approach their work with empathy for their teammates. Scalability and flexible provisioning are also necessary. With DevOps, those who need power the most get it — through self-service and automation. Developers work closely with IT operations to speed software builds, tests and releases — without sacrificing reliability.

And now, those using DevOps approaches have realized the need to integrate security into the development and operations flow, leading to a call for DevSecOps.

While DevOps was a drastic shift of mindset from agile, DevSecOps isn't that significant of a change if your agency has already started adopting DevOps practices. Essentially, DevSecOps focuses on the often neglected pillar of security across the development pipeline. Often in DevOps, the focus is on automation, collaboration and other core pillars of the DevOps methodology, and in the process, security can be forgotten.

DevSecOps aims to shift this by giving security the focus and priority it needs and deserves. It doesn't necessarily call for a separate security team, but rather looks to infuse security principles at every step, and into every collaborator, including developers and QA.

In part, DevSecOps highlights the need to invite security teams at the outset of DevOps initiatives to build in information security and set a plan for security automation.

Whether you call it "DevOps" or "DevSecOps," it has always been a best practice to include security as an integral part of the entire application lifecycle. DevSecOps is about built-in security, not security that functions as a perimeter around apps and data. If security remains at the end of the development pipeline, agencies adopting DevOps can find themselves back to the long development cycles they were trying to avoid in the first place.

The Tools and Tactics to Achieve DevSecOps

Selecting the right tools to continuously integrate security can help meet your security goals, but effective DevOps security requires more than new tools — it builds on the cultural changes of DevOps to integrate the work of security teams sooner rather than later. Let's take a look at what tools and tactics are necessary to bring DevSecOps to your agency.

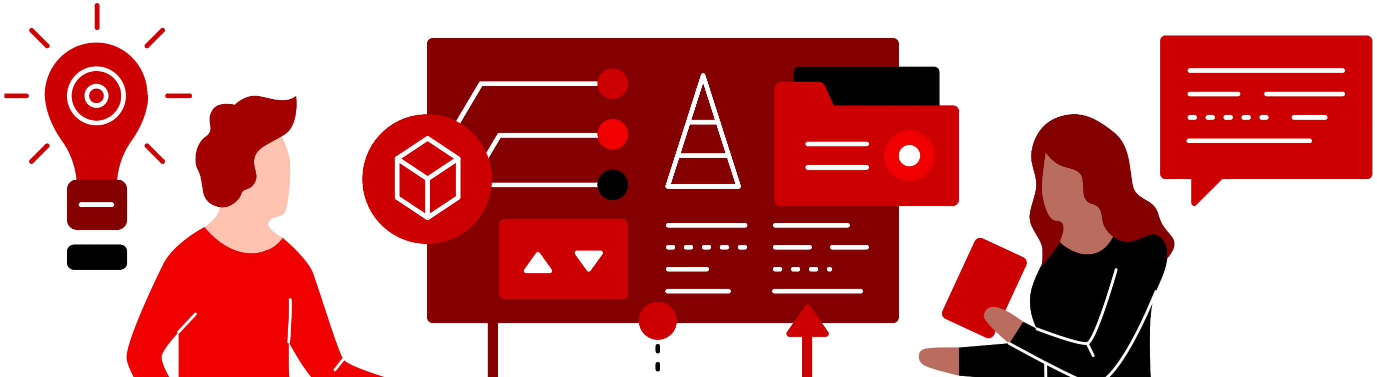
Containers and Microservices

The greater scale and more dynamic infrastructure enabled by containers have changed the way many organizations do business. Because of this, DevOps security practices must adapt to the new landscape and align with container-specific security guidelines. Cloud-native technologies don't lend themselves to static security policies and checklists. Rather, security must be continuous and integrated at every stage of the app and infrastructure lifecycle.

DevSecOps speeds up how an idea goes from development to deployment. At its core, DevSecOps relies on automating routine operational tasks and standardizing environments across an app's lifecycle. Containers offer these necessary standardized environments. They make it easier to move applications between development, testing and production environments. Using containers lets developers package and isolate their apps with everything they need to run, including application files, runtime environments, dependent libraries and configurations.

Automation

IT architectures are continually changing and must be infinitely flexible. Automation in software – and in security, in particular – helps with efficiency, delivering value faster and solving IT and business workflow challenges. As organizations adopt containers, an automated approach to security, testing and application development is needed to increase productivity and reduce risk. By automating security capabilities like enterprise firewalls, intrusion detection systems (IDS) and security information and event management (SIEM), organizations can better unify responses to cyberattacks through the coordination of multiple, disparate security solutions, helping these technologies act as one in the face of an IT security event.



Open Source

DevSecOps relies on a culture of collaboration that values openness and transparency. Implementing this approach means applying open source principles and practices because the cultural values of DevSecOps are tightly intertwined with the values of open source communities and agile approaches to work.

The culture of open source software projects can be a blueprint for how to build a DevOps or DevSecOps culture. Freely sharing information is the default approach to collaboration in open source communities. It can help to implement cultural changes like promoting transparency in decision-making, encouraging experimentation by eliminating the fear of failure or implementing a reward system that encourages trust and collaboration.

Culture

The key to looking at software or technology as an evolutionary change is that evolution is a natural function of its environment. In a business, evolution comes about with the culture. The changes necessary to support evolutionary change can be supported by management, but they can't be dictated by management. People need to want to change. It's a matter of free will, not force. Gartner notes that "90% of organizations attempting to use DevOps without specifically addressing their cultural foundations will fail."

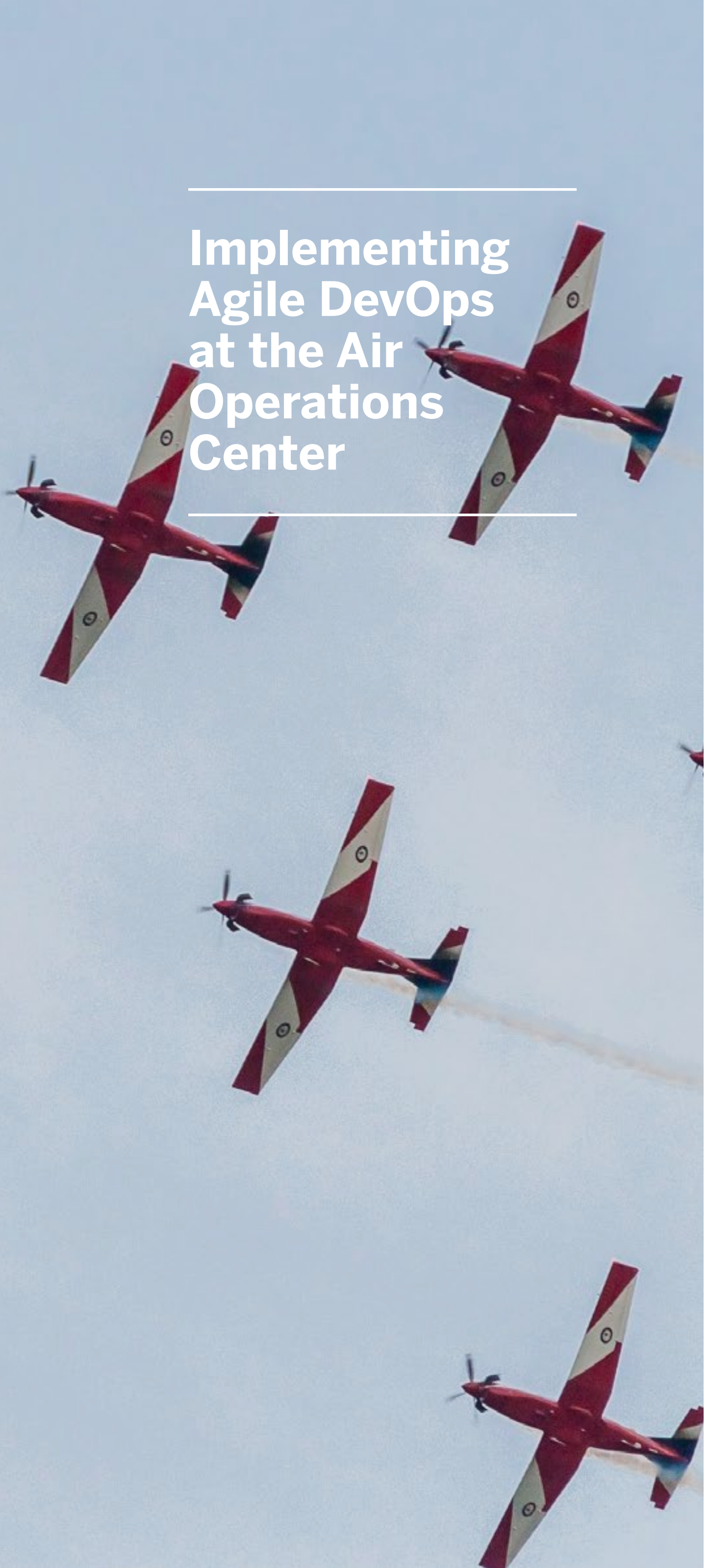
Changing the infrastructure or the application architecture is easy. To effectively change what you produce, you need to change your culture first. And cultural change goes even deeper than DevSecOps or agile or other methodologies. It is a commitment to actually putting everyone on the same team.

Major changes can begin with very simple steps. Cultural changes underpin all of the technological and process changes. If you're struggling to build a DevOps or DevSecOps culture, try two things:

- Have your developers spend the weekend with operations and security teams watching a production rollout and learning what they go through.
- Track how many steps or service tickets it takes for a developer to request a new virtual system.

Seeing how other teams are functioning in real time can be a powerful force to encourage teams to change their processes or to open up communication.

Implementing Agile DevOps at the Air Operations Center



Lean product development, user-centered design and extreme programing are terms usually associated with Silicon Valley startups, not the Air Force.

Similar to other military branches — and the Defense Department as a whole — acquisition strategies have been notoriously complex, cumbersome and in some cases, years long, only to be scrapped without providing value. That was the case with the Air Operations Center –Weapon System (AOC) Increment 10.2 program. AOC is a major automated information system used by the Joint Forces Air Component Commander to plan, execute, monitor and assess air, space and cyberspace operations. The AOC 10.2 contract was being developed to address application integration problems and cybersecurity vulnerabilities with the previous iteration of the program.

But those efforts were scrapped in 2017 after the Air Force spent half a billion dollars over a 10-year period. "[It] really was the impetus behind trying a different way and got us on the path toward Kessel Run," **Lt. Col. Jeremiah Sanders**, Program Manager, AOC, and Deputy Director of Kessel Run.

The Kessel Run team prides itself on integrating DevOps into Air Force acquisition practices and describes its work as revolutionizing "the way the Air Force builds and delivers software capabilities." It does this by taking industry-proven software development practices, such as Agile and DevOps, and empowering airmen to use those practices.

GovLoop recently sat down with Sanders and **Steve Wert**, Program Executive Officer (PEO), Digital, to discuss about what's new for Kessel Run and the Air Force, tangible outcomes, and Wert's role in evangelizing DevOps across the department.

The interview was lightly edited for clarity and brevity.

GovLoop: Can you talk about the success of Project Kessel Run and how PEO Digital builds on that success?

Sanders: Inheriting the challenge of modernizing the AOC, the opportunity exists now where we're able to take very small bite-sized chunks of that problem set, as opposed to the big bang, decade-long approach that was AOC 10.2. We stood up several very autonomous software development teams. We have about 20 teams that are building on specific operational capabilities or outcomes that we have to deliver to the AOC enterprise. We started that primarily with the 609th Combined Air Operations Center in Al Udeid [Air Base in Qatar] because they are fighting the current real fight over there.

And we were able to see significant operational improvements having an impact on the war fight within a matter of months. We were saving millions of dollars a month — in fact $13 million a month — in fuel, within just a few months, [and] also significantly reducing timelines and increasing validity in the air tasking cycle to include how we deliberately or dynamically target activities in in the theater, as well as on the mission planning side of the house, how we plan out and execute the air campaign in the theater. And then we were able to expand those capabilities to other geographic air operation centers across the world to the point that now, we're pushing new capability out of our development operations environment and into real-world operations every 15 hours. And we're getting faster.

What Agile and DevOps, or DevSecOps, are talking about is really the ability to continuously deliver capability so that we burn down the risk of delivering the wrong thing. We're able to get feedback from end users and iterate on that capability. The underlying technology infrastructure that allows distributing the software continuously on a worldwide scale, including multiple classification levels, was really a big enabler of what we've been able to do from a war fight perspective. We have subsequently onboarded another 20-plus teams from other activities within the Air Force and [are] also supporting the F-35 [aircraft program].

Can you make the connection between the work you do and how that is leading to fuel savings?

Sanders: The work that we did to create software enabled them a much more efficient [approach], both in terms of human capacity and the use of our refueling aircraft for aerial refueling. And now we've taken that same capability and are using it in the Korean Theater, Pacific Theater and others.

Wert: Col. Sanders, is it fair to say that people are using the time that they would have dedicated fighting an antiquated system to more methodically plotting out how to put tankers where they're needed most?

Sanders: I would say that the software enables what used to be humans doing that on calculators and whiteboards to the point that now there are fewer humans involved. It used to take six people six to eight hours a day to do that plan. Now it takes two people 30 minutes.

Wert: In parallel with the journey that we've been on with Kessel Run, as a Program Executive Officer, I can direct our other software efforts to start transitioning to this Agile DevOps approach, and I have been. So, we actually now have many examples we can talk about of software efforts, like Personnel Recovery Command and Control is releasing updates every two weeks.

It's an important part of software. When we have a downed airman, this is used to report and locate. So that's one way that we've been building on the success of Kessel Run. We're working to apply it everywhere within my portfolio.

The other way we're spreading this is helping other PEOs establish software factories, somewhat akin to Kessel Run. There's a small software factory that's been stood up in Colorado Springs, [Colorado] called Space Camp. There's an organic mobile apps team in place in Montgomery, Alabama, working for PEO Business Enterprise Systems, and he's working to stand up a larger software factory there. And then FMC is working to stand up a software factory out in Los Angeles.

SURPASS THE COMPETITION
Accelerate your business with DevSecOps.
LEARN MORE
Red Hat

Open Source as the Driver for DevSecOps

An interview with Dave Cohn, Cloud Native Subject Matter Expert, Red Hat

DevOps is taking off in government project management today.

But where is security in this growth? According to a Gartner report, by 2019, more than 70% of enterprise DevOps initiatives will have incorporated automated security vulnerability and configuration scanning for open source components and commercial packages, up from less than 10% in 2016.

Given the advance of cyberattacks and security issues throughout government today, this focus on security needs to continue – and it makes it more important than ever that security teams are incorporated into the DevOps culture that is developing. This leads to DevSecOps – meaning thinking about application and infrastructure security from the start.

To learn more about how DevSecOps can evolve in government, and how open source can drive that evolution, GovLoop spoke with Dave Cohn of Red Hat. Red Hat is a leader in open source technology for government.

Cohn noted that DevOps isn't just about development and operations teams. If you want to take full advantage of the agility and responsiveness of a DevOps approach, IT security must also play an integrated role in the full life cycle of your apps. Effective DevOps ensures rapid and frequent development cycles (sometimes weeks or days), but outdated security practices can undo even the most efficient DevOps initiatives.

That's why open source needs to comes in, Cohn explained. DevSecOps relies on a culture of collaboration that values openness and transparency. Implementing DevSecOps means applying open source principles and practices because the cultural values of DevSecOps are tightly intertwined with the values of open source communities and agile approaches to work.

"Open source is the fastest way to drive innovation, especially through the government, and making use of enterprise-grade, secure and hardened open sourced products. That's how you really accelerate the use of DevSecOps and get moving fast on innovation."

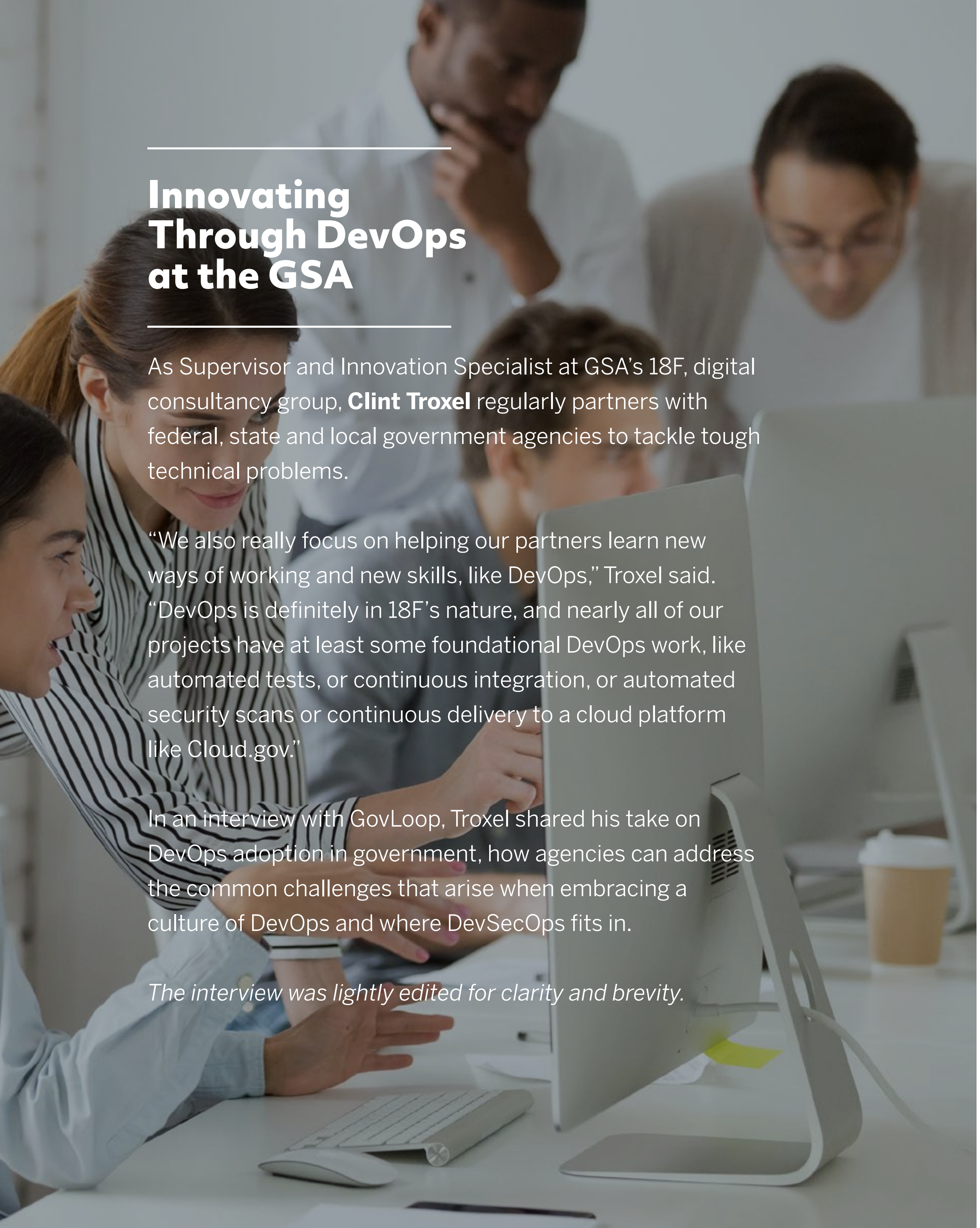
That's where Red Hat and their OpenShift platform comes in. As agencies move toward cloud environments, DevSecOps, and modern architectures, microservices and containers have become an essential utility. Containers have broad appeal because they allow users to easily package an application with all its dependencies into a single image that can be promoted from development to test environments and production—without change. Agencies now face the challenges of keeping their containers secure and trying to deliver DevSecOps.

OpenShift is a family of containerization software developed by Red Hat that addresses the security issues.

"Red Hat OpenShift Container Platform is designed with DevOps teams in mind, letting you automate the container application life cycle and integrate security into the container pipeline," Cohn said.

"Security isn't an afterthought, it's continuous in the process," Cohn said. "And that's what it means to really do DevSecOps. As long as you have developers, security, and operations working together cohesively, you're going to be in a good place."

Takeaway: Implementing DevSecOps means applying open source principles and practices because the cultural values of DevOps and DevSecOps are tightly intertwined with the values of open source communities and agile approaches to work.



Innovating Through DevOps at the GSA

As Supervisor and Innovation Specialist at GSA's 18F, digital consultancy group, **Clint Troxel** regularly partners with federal, state and local government agencies to tackle tough technical problems.

"We also really focus on helping our partners learn new ways of working and new skills, like DevOps," Troxel said. "DevOps is definitely in 18F's nature, and nearly all of our projects have at least some foundational DevOps work, like automated tests, or continuous integration, or automated security scans or continuous delivery to a cloud platform like Cloud.gov."

In an interview with GovLoop, Troxel shared his take on DevOps adoption in government, how agencies can address the common challenges that arise when embracing a culture of DevOps and where DevSecOps fits in.

The interview was lightly edited for clarity and brevity.

GovLoop: How would you describe the maturity or use of DevOps in government?

Troxel: Some places in government are not mature. There are other places in government that are on a rocket ship and working at the edge of what we can do with DevOps. I've seen a huge increase in the use of DevOps and the interest in DevOps tools and practices in federal and state and local government IT organizations, too. But one thing that I can say for certain is that there's still a lot of work to do — a lot of very foundational work.

As a government leader in the DevOps space, what do you see as the biggest challenge that organizations face, especially if they're transitioning?

One of the really big challenges in DevOps is just to make sure that we're all using the same words to speak the same language. DevOps has been a little hard to define historically. One way to think of DevOps is like a natural extension to Agile, not a transition. I think of DevOps as sort of extending Agile principles past the development of code into the entire system, into things like operations and security — and being Agile everywhere.

The best way that I know to start solving problems or making sense of DevOps, or these new tools that it brings, or these new concepts, is to just start — to start small and to start as simply as possible and to start now. I think one of my favorite things about DevOps is that there are often some very low-hanging fruit. So, you can begin very simply and still make real improvements.

There have been horror stories of developers wasting hours because they ultimately didn't have permission to deploy code. What questions or red flags should employees consider before diving into DevOps?

You can find horror stories on both sides, with people who are not using DevOps or people who are using DevOps. I think that the really important thing to note here is that if you're learning a thing — I don't care what it is — you're going to make mistakes. And it's important to acknowledge that up front, and to make it OK. In DevOps, a way that we do that is to create a safe environment or create an environment where consequences for mistakes are low – a place where you can go practice a bunch.

What practical advice can you share to help others avoid those horror stories and ensure everyone is on the same page?

Automation and collaboration are core tenets of DevOps. Even in a situation where you have these wonderful automated scripts that do everything for you and don't require any handholding, things still break. It's software; there are bugs. Nothing works forever. And so, again, having other environments where you can practice and where you can make these mistakes and learn from them [is critical]. In fact, where you can, try to make these mistakes. [For example,] what happens if I lose my password for the deployment or whatever it is? How are we going to handle that in the future if that problem arises?

In a previous blog post you wrote, "Agile without DevOps is a bundle of potential energy with no outlet." Can you provide an example of how you've seen that play out in government, and what agencies can do to provide that outlet, obviously using DevOps?

Agile taught us to quickly adapt to changes like new features in our code. That was great, and the cycle requires lots of feedback from users as part of that Agile cycle. You make changes and then you check with users, and then see what they thought and then you incorporate those changes. A couple examples of how DevOps can really unlock that cycle is by creating these automated environments where potential changes can be tested and reviewed. That's the thing DevOps can do.

Git is at the root of a whole host of collaboration in automation in DevOps. It's where we talk about our code, where changes to code trigger other events in other systems. So, yes, agencies that are practicing Agile software development should also invest in DevOps because to say DevOps is to say applying Agile practices to operations.

You also wrote, "When bringing DevOps into government, making it DevSecOps instead may be the difference between failure and success." Can you elaborate on that?

In the federal government, security requirements are very complicated and demanding. Security offices can stop a project dead in the water. So, we learned what helps a lot in our project is for us to interface very early with the security apparatus, whatever it is. To try and, in a friendly way, break down those walls around security like we did for development and operations.

What has DevOps enabled that you otherwise think would not have been possible?

I don't think 18F would be here if we ignored DevOps. Processes like automaton, I see them as a lever that really enables small teams like 18F to have an oversized effect. Our entire Platform-as-a-Service, Cloud.gov, is run by a small team of humans. That would be impossible without large amounts of automation. Operating these platforms, and operating these services with small teams is only possible because of deployment automation, because of test automation, because of DevOps.

Conclusion & Next Steps

Before agencies dive into the next stages of becoming an intelligent enterprise, they must understand a variety of technologies, practices and approaches to changing the culture that support the way forward.

Today, it's clear: In the collaborative framework of DevSecOps, security is a shared responsibility integrated from end to end. To get started, a good DevSecOps strategy is to determine risk tolerance and conduct a risk/benefit analysis. What amount of security controls are necessary within a given app? How important is speed to market for different apps? Automating repeated tasks is key to DevSecOps, since running manual security checks in the pipeline can be time-intensive.

For now, take a look at the possibilities and benefits that DevSecOps could bring to your agency, as well as some questions you and your team need to be asking yourselves as you move toward this new way of digital transformation and innovation.

With DevSecOps, it's important to determine where your organization can realistically go. This doesn't mean "easily," since the goal of digital transformation is to significantly change the culture, processes, architecture and technology of your organization. It means understanding what you are trying to achieve with that change and then clearly assessing what it would take to move toward that goal.

Ask yourself and your team these questions:

- What are your current team or group divisions?
- What are the communication patterns between those groups?
- Who is currently involved in planning cycles?
- Looking at functionality, how close is your current application architecture to your desired application architecture?
- What is the level of risk or failure tolerance within your organization?
- How well understood are your material and information flows? (This is value-mapping your organization.)
- How frequently do you need to be able to release an update to meet customer or operational needs?
- What new functionality is required by either business objectives or development needs?

Benefits of DevSecOps

- Faster application development lifecycles
- Increased developer productivity
- Reduced IT time per application developed
- Lower costs through greater efficiencies
- Improved service quality and reliability
- Reduced risk of deployments
- Faster adaptation to market changes
- Competitive advantage through speed to market
- Improved customer satisfaction
- Higher return on investment (ROI) with more applications in less time

Red Hat

Thank you to Red Hat for their support of this valuable resource for public sector professionals.

govloop

About GovLoop

GovLoop's mission is 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 the public sector.

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

govloop.com | @govloop