GEOCENT

# Navigating DevSecOps

Nearly everything you need to know to adopt a better security posture for your mission.

# Table of Contents

# Introduction

Many federal agencies still utilize the “waterfall” method when it comes to software systems and development. One department completes a task and hands the project off to the next department, who builds the next item on the list. This is the way of the past. Today’s modern agency calls for the flexibility to transform software rapidly while ensuring that security tools and processes are baked into the software delivery lifecycle.

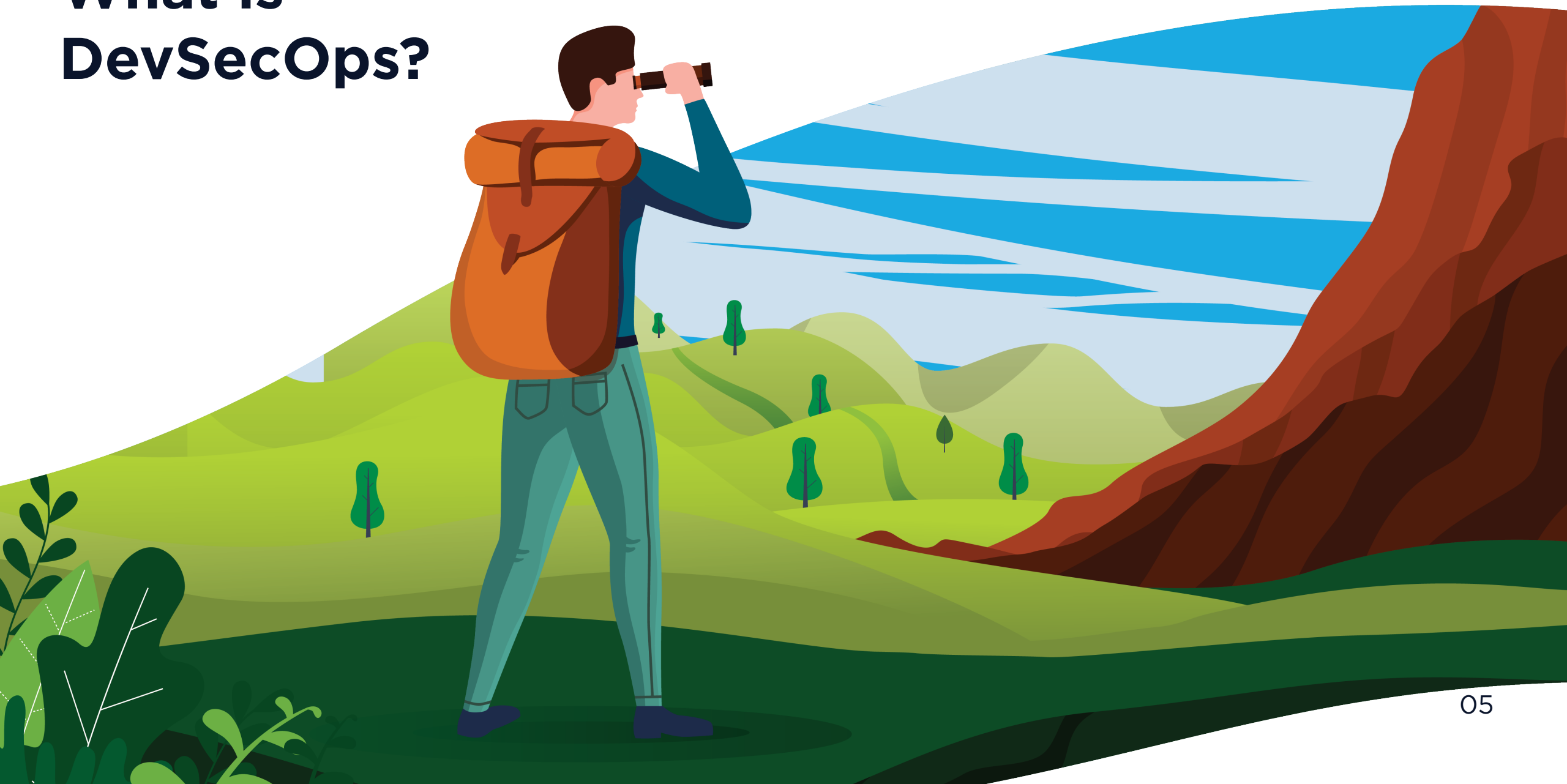
Not sure how to navigate to a DevSecOps approach? Not really sure what it is and how it fits into your modernization plans? Don’t worry. You’re not alone.

By adopting a DevSecOps mindset, you’ll set your development, security, and operations teams up for success with the right tools necessary to scale the mountain of transformational change required for your mission goals.

Still lost?

**We’ll show you the way, help you maneuver around risks, and spot problems before they spot you.**

# What is DevSecOps?



# DevSecOps

DevSecOps is an evolution from the old waterfall method. It's the pivotal response to older, bottleneck-riddled security models on the modern continuous delivery pipeline. Building and automating the development process bridges the traditional gap between IT and security while ensuring fast, scalable and reliable delivery of code. For government agencies, once-siloed teams are enhanced by increased communication and share the responsibility of security tasks during the delivery process.

In DevSecOps, normally opposing goals — speed of delivery and secure code — are merged into a streamlined system. Utilizing lean-agile practices, security testing is done in iterations without slowing down delivery cycles. Security issues are dealt with as they arise; not after a threat of compromise has occurred.

## Pit Stop

# Kessel Run

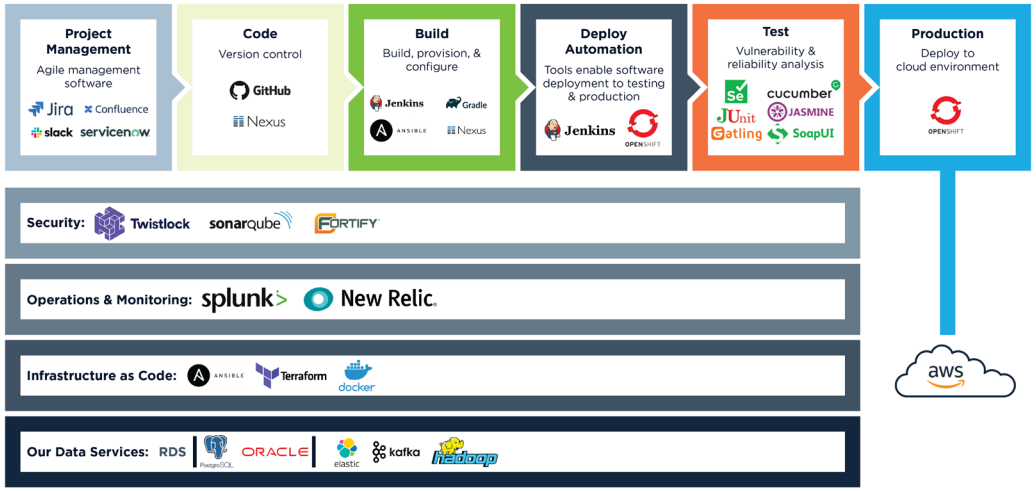
The Software Factory that Fights Wars

The goal of Kessel Run was to change how the Air Force, and by extension the Department of Defense, develops and delivers software run on classified networks that adopts the best practices from the commercial industry.

The program, once thought of as far-fetched, successful demonstrated continuous delivery of software development, or in developer terms, the holy grail. It was a huge win for the Department of Defense.

# Understanding the DevSecOps pipeline

**DevSecOps Pipeline Example**




A typical DevOps pipeline includes different stages. For example, a typical software development lifecycle (SDLC) includes phases like Plan, Code, Build, Test, and Deploy.

In DevSecOps, specific security checks are added in each phase.

**Plan:** Execute security analysis and create a test plan to determince scenarios for where, how, and when testing is done.

**Code:** Deploy linting tools and Git controls to secure passwords and API keys.

**Build:** While building code execution, incorporate static application security testing (SAT) tools to track down flaws in code before deploying to production.

**Test:** Use dynamic application security testing (DAST) tools to test your application while in runtime. These tools actively investigate running applications with penetration tests to detect possible security vulnerabilities.

**Deploy:** After completing the above tests in runtime, send a secure build to production for final deployment.

# How is DevSecOps different from DevOps?

## DevOps: Critical Combination

DevOps is the collaboration between development and operations teams to create a more agile, streamlined deployment framework. It's a critical shift away from the traditionally siloed mentality of many IT teams, which prioritizes areas of specialization over communications.

DevOps has become a driving force in many forward-thinking agencies — born from the need to deliver software and services more reliably and quickly. The ideals of continuous testing and automation are essential to DevOps implementations. New deployments must be tested from the moment code is written to the hour the final product is released. Leveraging automation makes it possible to address form and function issues at speed rather than relying on outdated manual testing frameworks.

## DevSecOps: Logical Next Steps

DevSecOps introduces the concept of security into the existing DevOps paradigm. It's critical to apply the notion of developing a "Security-as-code" culture that prioritizes secure deployment and speed rather than attempting to separate the two concepts.

DevSecOps integrates key security policies such as code analysis, compliance monitoring, threat investigation, and vulnerabilities assessments into typical DevOps workflows. The ideal result? Native security already built into new product deployments, which limits the risk of zero-day flaws and software recalls.

# How to navigate treacherous terrain

## Who will give you the most pushback on your journey to DevSecOps?

To fully embrace DevSecOps, you must break down barriers and foster open collaboration between, development, security, and operations organizations. DevSecOps is a software engineering culture.

The cultivation of share responsibilities is vital. All DevSecOps team members must see the delivery of secure services as their responsibility, rather than something handled by other teams during development or after services are deployed.

As it turns out, not everyone in your organization may be as excited about the culture change as you. Not everyone likes change, and there's one thing for certain, DevSecOps will bring change—how you work, what you do, how you interact with other people in the team, and beyond.

## Not Invented Here

“Been there, done that.” We’ve all heard this. If your management has decided that a move to DevSecOps should be undertaken—even if the existing practices have been working—there’s probably been a realization that things could be more efficient, faster, and more secure.

“To win these folks over, show how the new idea makes the system and their ability to influence it better. There is a bit of a leap of faith, but keeping an eye on the price (the business value) and enabling them to deliver it in a slightly different yet more transparent way is what truly helps move this type along.” — Matt Takane, Agile coach, Red Hat Open Innovation Labs

## Losing Control

People who have gained a level of experience in a particular area or domain and feel threatened by new processes. They often feel they are giving up control or diluting their expertise.

It’s important to stress they’re not diminishing expertise, but rather applying it to a broader set of processes. For example, testing experts need to explain to developers and operations people how testing methodologies can be exposed in their areas.

## Stuck in the Middle

Middle managers are often stuck trying to manage their operations, preventing them from seeing the big picture when it comes to change. They will mostly likely accept something once it has been tested and proven and has reliable people backing it.

To help this type, “The biggest thing is balance. You have to balance your wants and desires with your manager’s and that of the organization as well. Coaches, influencers, and friends help here, and so does time management. Start small; you don’t eat an entire meal in one bite.” — Chris Short, principal product manager, Red Hat Ansible

## Burned by the Past

These people, who have had bad experiences or wrong incentives from the organization, have either been through too many reorganizations or think DevSecOps is another fad.

Perhaps they put a lot of effort on the last agile transformation, but the organization only sent them to Scrum training, so no additional benefits were realized. Other people might be afraid of losing their power or jobs.

“When you can prove things like decreased lead time, more frequent and successful deployments, faster feedback, etc. then the naysaying eventually stops and whoever still doesn’t agree will likely leave. And that’s OK.” — Jared Ladner, Chief Architect, Geocent

## The Careful CIO

You've likely met this person. Their fiefdom is crumbling, they've been burned by previous incidents, or they've been mandated to adopt and are playing catch-up.

The good news: It's not too late. Show them what is likely an easier on-ramp thanks to maturity. Also, remind them that playing catch-up is possible, but there has to be a real concerted effort with headcount and funding. Look across the agency and organization for help and advice. Your problems probably aren't as isolated as you think they are.

# Field guides: Real-world examples

# Department of Homeland Security U.S. Customs & Immigration Services Verification Modernization

**The Problem:**

The Department of Homeland Security Citizen and Immigration Services (USCIS) needed to modernize its systems used for immigration status verification. USCIS processes millions of requests for Immigration Benefits per year; It needed a system that could keep up with a rapidly increasing case volume. The monolithic legacy systems in place could not keep pace with the flood of requests and policy changes.

**The Solution:**

Geocent worked with the USCIS OIT leadership, with little oversight, to design and implement modernize systems used by USCIS Verifications. We provided two Scrum teams consisting of 12 people, along with a Program Manager, DevSecOps Architect, and a Data Specialist. Utilizing a DevSecOps methodology, Geocent helped replace these burdensome legacy systems with agile, efficient, cloud-based applications that help USCIS process claims more quickly and effectively. Geocent also used open-source technologies whenever possible to limit costs and increase flexibility.

**The Results:**

USCIS uses the following applications, provided by Geocent, to improve their processes:

**Status Verification System Modernization (SVS-MOD)**

SVS-MOD replaced the original legacy application, which was slow and inflexible. The new modified application helped USCIS Verifications better organize a backlog in Immigration Benefits claims and increase efficiency in processing cases. The modernized system accomplished this by providing automated case assignment based on USCIS Verifications business rules, better information to Verifications personnel through expanded data availability, and enhanced metrics to leadership so they could better understand the dynamic challenges of this process.

These improvements helped USCIS reduce its case backlog from hundreds of thousands to around 10,000. As a result, case processing time was also reduced tremendously.

**myE-Verify**

This application, available to the public, set up a system to allow individuals to check their eligibility for employment at any time. The new, standalone application provides enhanced security for users and interfaced with external USCIS partners such as Verizon, Equifax, and Akamai to enhance service offerings to public users. There are now more than 30,000 user accounts on myE-Verify.

**E-Verify/Social Security Administration Tentative Non-Confirmation Automated Response (EV-STAR)**

Social Security Administration offices across the country use EV-STAR to help resolve E-Verify cases where applicants receive a Tentative Non-Confirmation (TNC) because of negative data flagged in their SSA file. Geocent modernized this application using DevSecOps principles into a standalone system and thus greatly enhanced stability, scalability, and flexibility. This effort was completed ahead of schedule, , while still ensuring SSA satisfaction and seamless migration for end-users. The application now has more than 11,000 users worldwide.

**Verification Reports as a Service**

USCIS uses internal reports in nearly every aspect of its day-to-day operations. With its legacy system, reporting capabilities were inherently tied to the inflexible, monolithic system. With Geocent's help, a modernized, standalone reporting system was implemented that allows for reports to be automatically generated from a wider set of data and consumed as a service across USCIS Verifications applications.

# Meet MCBOSS: The first DevSecOps multi-platform capability for the Marine Corps developed by OASIS

**What is MCBOSS?**

The Marine Corps Business Operations Support Services (MCBOSS) is a revolutionary multi-platform, cloud-enabled environment that allows users to access and build applications for Marine Corps use. It was the first DevSecOps capability developed by the Naval Information Warfare Center (NIWC) Atlantic's newly accredited Operational Application and Service Innovation Site (OASIS) team, providing DevSecOps to the U.S. Marine Corps for the first time.

**Why was it built?**

***The Problem:*** Legacy Systems

Before MCBOSS, the Marine Corps had little in the way of cloud-enabled computing, and if they did, it took months to vet for security. Their legacy systems were operating relatively well, but software updates were slow — measured in years. They needed a cloud-enabled platform able to rapidly update with an authority to operate (ATO).

***The Approach:***

DevSecOps is the strategy—development, security, and operations. But what is the execution?

As part of the Application, Development, and Test Services (ADTS) team within OASIS, our team constructed the majority of the automated testing and development, which enables the DevSecOps process within the MCBOSS environment. Our team built custom software factories, or integrated sets of tools and data, for the project. These factories can help automate development and deployment of updates to the environment after they are built.

The details:

- We prioritized Infrastructure as Code (IaC), utilizing cloud agnostic tools such as Terraform to provision the environment.

- We automated the security and hardening into the IaC scripts so that hardened, secure environments can be provisioned at any time.

- We provided a secure, accredited hosting environment in AWS GovCloud so that application owners/developers can focus on software delivery vs software deployment, hosting, and OS hardening.

- We also furnished cloud resources and operational services using the "X as a Service" concept.

- Lastly, we utilized the software factories approach to provide a list of hosting platforms, accept the code base from application owners, and then deploy/host the software.

***The Results:***

MCBOSS is now fully operational, with approved applications running on the platform. These include:

Appian – A low-code platform that provides capability for enterprise application development.

Pega – This provides a no-code platform for model-driven, unified enterprise-grade, agile application development.

Pivotal – This platform is a unified, multi-cloud system that runs enterprise applications at scale.

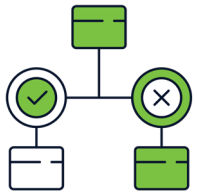
Tactical service-oriented architecture (TSOA) – TSOA is the Marine Corps service aligned with the DoD's net-centric services strategy (NCSS), which is an effort to better enable our warfighters by using the latest—and most secure—technology.

***The Future:***

Going forward, the Marine Corps can utilize and develop applications to better serve its business operations and, in turn, its warfighters. Having that secure, approved environment also saves time and money when completing agency objectives.

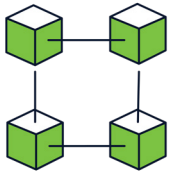
With success of OASIS and MCBOSS, federal agencies are no longer doubting the effectiveness of the DevSecOps approach. We expect to see the practice become the norm rather than the exception. Improvements in integration, automation, security, and remediation have become important influences across the government, in developing and prioritizing secure code.

# Packing the right tools



### Version-control System

Systems like Git are used for tracking changes in source code during software development. They are designed for coordinating work among programmers, but can also be used to track changes in any set of files. Their goals are speed, data integrity, and support for distributed, non-linear workflows.



### Container-orchestration System

Kubernetes is an open-source container-orchestration system for automating application deployment, scaling, and management. It aims to provide a “platform for automating deployment, scaling, and operations of application containers across clusters of hosts.” It works with a range of container tools, including Docker.

### Containers

Containers are a form of operating system virtualization. A single container might be used to run anything from a small microservice or software process to a larger application. Inside a container are all the necessary executables, binary code, libraries, and configuration files. It allows applications to run quickly and reliably from one computing environment to another.

### Open-source Automation Server

Software products like Jenkins are free, open-source automation servers which help to automate the non-human part of the software development process, with continuous integration and facilitating technical aspects of continuous delivery.

### Agile Project Management Tools

Tracking products like Jira or Confluence allow bug tracking and agile project management, both critical components of a software development infrastructure.

### Container Security

A container environment, in general, encompasses your images, containers, hosts, container runtime (Docker, runC, cri-o), registries, and orchestrator. Understanding potential risks and how to protect your environment against them is essential.

### Artifact Repositories

An artifact repository manages your end-to-end artifact lifecycle and supports different software package management systems while providing consistency to CI/CD workflow. An artifact repository is both a source for artifacts needed for a build, and a target to deploy artifacts generated in the build process.

The DevSecOps artifact repository is crucial for the software development process. Without it, things can become extremely muddled, with multiple developers using different artifacts and third-party components that only slow—or completely halt—your project. Using the artifact repository can help you agency avoid testing problems and delayed releases.

### Code Quality Tools

Platforms like SonarQube are used to continuously inspect code quality and perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities.

GEOCENT

DESTINATION POINT

## You Made it!

Congratulations! Now that you know everything that's needed to build a DevSecOps pipeline, check us out! We can help you modernize your application development and achieve your mission goals!

Stay up-to-date with the latest Agile information by subscribing to our newsletter.

**Subscribe!**