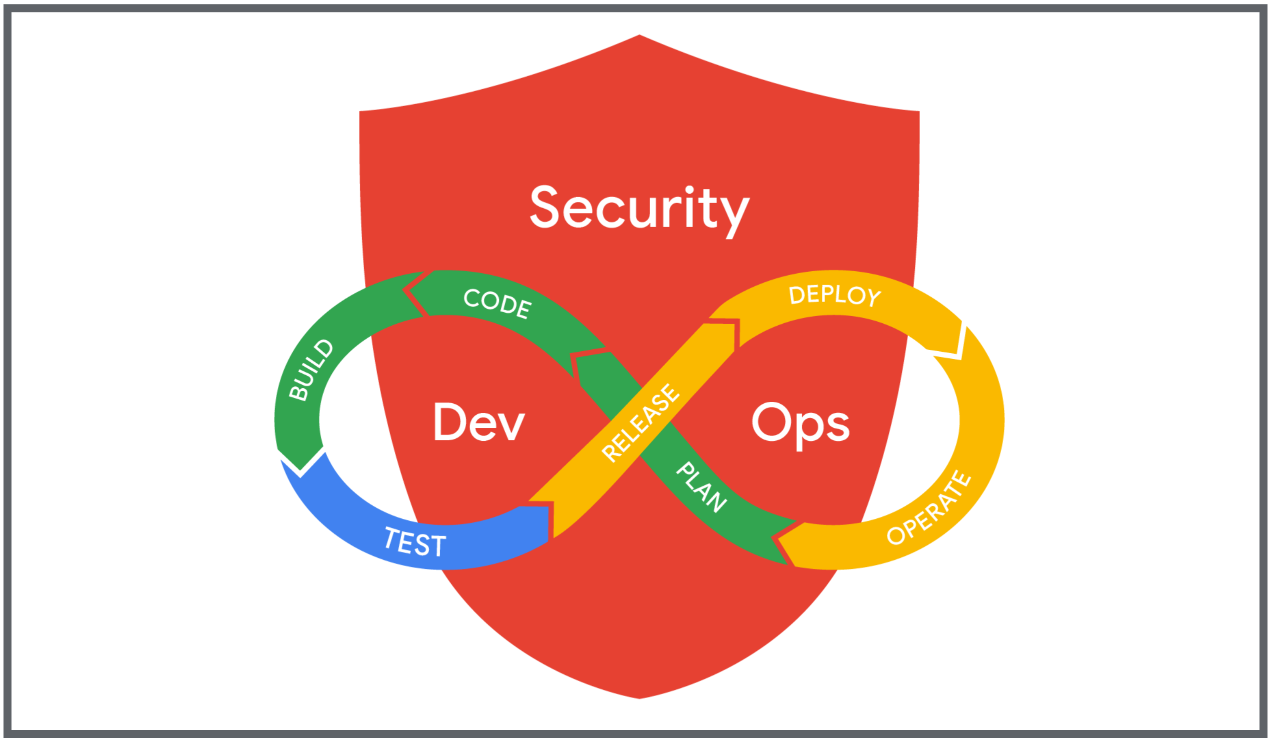
DevSecOps in action

DevSecOps is a collaborative culture, supported by guidelines, best practices, and tools that development, operation, and security teams use to work together. DevSecOps is an important part in the software development lifecycle. Each phase outlines the types of tasks and security measures needed to keep the development process in motion. In this reading, you'll learn more about security's role in the phases of the DevSecOps lifecycle.

The DevSecOps lifecycle

DevSecOps supports a "security first" culture, where security shifts left. Shifting left means that security is incorporated at the beginning of the development process instead of at the end. The DevSecOps infinity loop is a common way to portray the development and operations lifecycle with security intermixed throughout.



The DevSecOps lifecycle typically has seven phases, including:

1. Plan
2. Code
3. Build

4. Test
5. Release
6. Deploy
7. Operate

Phases of the DevSecOps lifecycle

Plan

The first phase is planning. In this phase, the development team plans the initial parts of the software development project. First, the team conducts a threat analysis to determine the types of threats that could create problems for the software. Then, they plan the type of security scans, tools, and tests that should be used throughout the lifecycle. During this phase, team members are assigned identity and access management (IAM) roles and permissions to ensure the correct access is granted to the correct people. Teams also outline the application's asset types, data, and templates, and the scope of the project is determined.

Code

In the coding phase, the software artifacts outlined in the plan phase are translated into source code. In DevSecOps culture, developers incorporate safe coding practices, like implementing code reviews and following secure code design practices. Development tools use automation to compile code and integrate security features. Tools might include plugins that monitor for bugs or policy and compliance requirements. Developers should use automation within their existing tools to streamline receiving security scan reports directly. This will enable them to quickly fix any identified vulnerabilities during the coding process.

Build

In the build phase, the software starts taking a more structured form, while automated tools build artifacts and services from the source code. While the application is being built, the DevSecOps team incorporates automated security elements like vulnerability checks and security scans. With shift left methodology in place, developers are alerted to these vulnerabilities early in the process.

Test

Once the software or application is built, the lifecycle moves into the test phase. During this phase, the software goes through several rounds of both manual and automated testing to ensure its build integrity. Tests assess software features ranging from integration and system capabilities, to functions and performance. For example, one purpose of a manual test is to perform quality assurance and triage any areas of concern for improvement. Automated tests check that security and compliance requirements are met.

Release

During the release phase, the configuration environment is evaluated against automated security checks to determine readiness for the production environment. The source code and software artifacts are reviewed and receive final sign-offs. Once the configuration has met all of these requirements, it's ready to be released in the deploy phase.

Deploy

The deployment stage is a huge moment for the DevSecOps team; the application goes live! In this phase, the team uses automation to push the software into production and to end users.

Operate

After the software is deployed, the operate phase monitors the deployment for any event or alert relating to functionality and vulnerabilities that need patching. A continuous feedback loop of the software's performance keeps operations, development, and security teams up to date and consistent.

Key takeaways

DevSecOps improves the software development process by fostering collaboration and efficiency among the development, security, and operations teams. The DevSecOps lifecycle supports shifting security left, so that vulnerability scans and remediation activities occur earlier in the development process. With DevSecOps becoming a more widely adopted culture and workflow, it's important to understand security's role in the development process.