All You Need to Know About DevSecOps and its Implementation

In today's IT world, few companies have yet to figure out how to secure the cloud effectively. As a result, a shortage of cloud security knowledge and legacy security regulations that do not cover the cloud.

Security must adopt at every stage of the DevOps life cycle, known as **DevSecOps**. A company that runs cloud technology should adopt the DevSecOps approach. It requires policies, procedures, and new security guidelines.

"By 2027, the global DevSecOps industry will have grown ninefold, to more than USD 17 billion, up from just over USD 2 billion in 2019."

Users who have a substantial exposure to the world of app and software development, then they are familiar with DevOps. But do they have any idea about the concept of DevSecOps? The combination of DevOps and security is DevSecOps.

Before going in-depth about the topic. Let's explore the concepts first. What is DevSecOps? Why is DevSecOps important, adopting and measures?

What is DevSecOps?

DevSecOps is a security issue for the development team and operations team. It's an automation, culture, and platform design model that involves security as a shared responsibility through the IT life cycle. It is an application security (AppSec) technique involving security in the software development life cycle (SDLC).

DevSecOps should include security across the SDLC so that DevOps teams can deliver secure apps quickly and with high quality. In addition, CI/CD workflow, such as testing and risk mitigation, reduces the time-consuming and expensive consequences of making patches post-production.

DevSecOps has always been ideal to build security as an integral part of their app life cycle. DevSecOps means built-in security into application development from start to end. It is an important evolution in how development companies approach security.

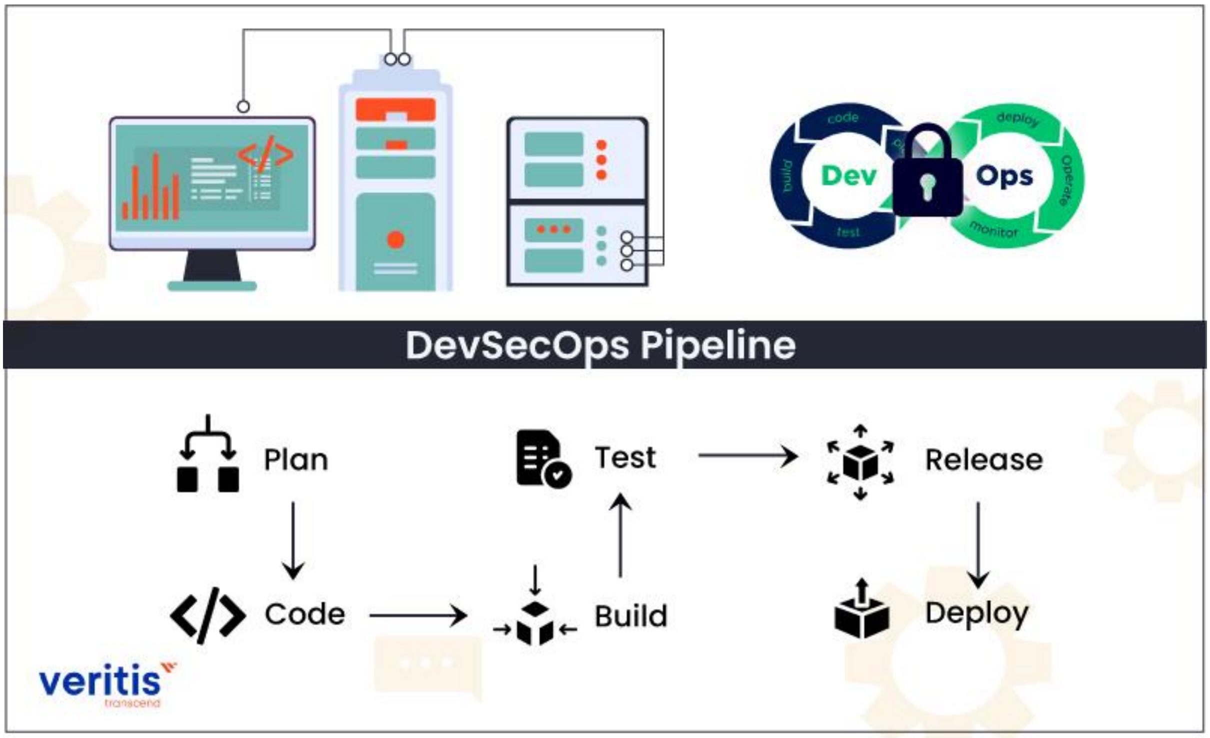
With real-time monitoring of loops and insights, DevSecOps spans the whole SDLC. From planning and design to coding.

Useful link: **DevSecOps Process, Benefits, Tools and Implementation**

To adopt DevSecOps, teams should follow the below steps:

- To reduce software code attacks, include security throughout the SLDC.
- Assign some security practices to the DevOps team.
- Allow automatic security checks at each stage of the SLDC to combine tools, processes, and security controls through the DevOps workflow.

Security should apply at each stage of the DevOps pipeline, including plan, code, build, test, release, and deploy.



Plan

DevSecOps plan stage is the least automated, comprising discussion, review, collaboration, and strategy of security analysis approach. Teams should execute a security analysis and develop a plan outlining where, how, and when security testing will happen.

IriusRisk tool is the most DevSecOps plan tool and is a collaborative design tool for threat models. Jira tool is for issue tracking and management. The slack tool is for software, communication, and chat platforms. All these tools use as add-ons.

Code

Developers can run DevSecOps in the code phase to guide them write better secure programs. Code-phase security practices include pre-commit hooks, static code analysis, and code reviews.

Security technologies involve in developers' existing Git workflow. When a security test automates triggers every commit and merge. It even supports combined development and programming languages. Security code tools such as CheckStyle, PMD, Gerrit, Phabricator, Find Security Bugs, and SpotBugs are the most popular.

Build

The build process starts when developers add code to the source repository. It aims to create tools to automate the security analysis of build output artifacts. Security approaches include unit tests, software component analysis, and static application software testing (SAST). In addition, it can add tools to an existing CI/CD pipeline to automate these tests.

Checkmarx, SourceClear, OWASP Dependency-Check, SonarQube, Snyk, and Retire.js is for the build phase tools.

Useful link: DevSecOps – A DevOps Savior to 'Cybersecurity' Challenge!

Test

The test phase starts after a build artifact is built and deployed to testing conditions or staging. Executing a complete test suite requires a significant amount of time. Therefore, this phase should fail quickly to complete the more costly test jobs later.

During the test phase, DAST tools used to detect actual application flows such as SQL injection, authorization, user authentication, API-related endpoints, and SQL injection. There are multiple paid testing and open-source tools available in the current market, including Gauntlt, Arachi, Boofuzz, JBroFuzz, BDD Automated Security Tests, Owasp Zap, SecApp suite, and IBM AppScan.

Release

Test the application and executable code when the DevSecOps cycle reaches the release phase. The phase explores the configuration variables such as network firewall access, personal data management, and user access control to secure the runtime architecture.

Configuration management technologies are essential for security during the release phase. It allows visibility into the static configuration of a dynamic environment. Terraform, Puppet, Chef, Ansible, and Docker are configuration management tools.

Deploy

It's time to deploy the build artifact to production if the preceding phases were successful. The only security issues to address during the deployment phase are against the live production system. Production TLS and DRM certificates should be checked and assessed for impending renewal.

Useful link: What are the best DevSecOps practices for security and balance agility?

Why is DevSecOps becoming essential?

DevSecOps combines security into the SDLC earlier. As a result, it is easy and less expensive to discover the patch vulnerabilities before they go too far into production.

When development groups code with security from the start. In addition, companies with multiple industries can adopt DevSecOps to break through boundaries between development, security, and operations. It allows them to deploy more secure software faster.

Companies might ignore security measures for speed, but this is an experiment that could backfire disastrously. Do you want to experiment with jeopardizing your latest app launch, especially if the launch's success is critical to your company's survival? Then there's the possibility that a slew of security concerns emerge after the product is released, resulting in an army of irate, disgruntled customers, many of whom will abandon your product and organization.

IT security is a significant concern in today's digital world, and the threats aren't going away anytime soon. Cyber-attacks and fraud are becoming more common. With this harsh reality in mind, it's impossible to see any enterprise today ignoring the security part of the DevOps process.

Useful link: DevSecOps Solution to Cloud Security Challenge

Challenges of DevOps security

- Security observes as a problem by DevOps teams
- IT security teams are unable to keep up with DevOps' speed
- Most inexperienced tools and open sources have security flaws
- More attack opportunities emerge as a result of inadequately managed privileged access controls

Adopting DevSecOps measures

To successfully adopt DevSecOps in a strategy summed up as "moving security focus to the left," the team must ensure that security embeds into the program development from starting to end.

Below mentioned are five crucial components of any DevSecOps model are:

Compliance monitoring

Stay compliant at all times to be ready for an audit. Integrated compliance monitoring provides a framework for accomplishing. It allows teams to work more quickly while maintaining traceability and more reliable controls.

Code analysis

Deliver code in small chunks to make it easy to discover faults quickly. Of course, the business will always need to perform code analysis. First, however, the functionality must be ingrained into the tooling that developers use while pushing, merging, pulling, writing, and integrating lines of code.

Change management

Any team can submit changes, then decide whether the change benefits or troubles the team. Change management is associated with bureaucracy in IT businesses.

Threat investigation

Identify threats as they emerge in each code to change and update quickly. Threat investigation is one of the crucial security practices used to guide and visualize the potential of something awful.

Useful link: Achieving Continuous Application Security with DevSecOps

Final Thoughts on DevSecOps

More development teams modernize their processes and use new tools; they must maintain a high level of security. It is a cyclical process that should regularly be iterated and applied to new code deployments. Hackers and exploits constantly change; thus, modern software teams must keep up.

Moreover, Enterprises will adopt DevSecOps to a greater extent as the process becomes more automated. DevSecOps deployment becomes a no-brainer when automation is combined with high security.

Veritis, the Stevie Awards winner, has been a trusted partner for Fortune 500 companies, including emerging companies; we have enough expertise to offer the best solutions for your business.

So, reach out to **Veritis** and walk away with the best DevSecOps tools and cost-effective solution.

Services