INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

DEPARTMENT OF SOFTWARE ENGINEERING

SOFTWARE ENGINEERINGTOOLS AND PRACTICES

ASSIGNMENT

ZEKARIAS WOREKET 1306501

Contents

Introduction

In this introduction, we will embark on a journey to uncover the origins, workings, benefits, and career opportunities associated with this exciting field.

Firstly, we will delve into the root cause that led to the initiation of DevSecOps. Software engineering encountered a series of challenges that necessitated a paradigm shift. Traditional practices often failed to adequately address security concerns, leaving software vulnerable to malicious attacks. This realization sparked the inception of DevSecOps, aiming to seamlessly integrate security practices into the software development process.

Next we will shed light on what exactly DevSecOps entails. DevSecOps, a fusion of Development, Security, and Operations, represents an approach that emphasizes collaboration and integration of security throughout the entire software development lifecycle. It seeks to embed security as an integral component right from the inception of an idea, rather than an afterthought tacked on at the end.

To gain a better understanding of DevSecOps, let's briefly explore its lifecycle. The DevSecOps lifecycle revolves around four key stages: planning, development, testing, and operations. Through continuous integration, continuous delivery, and continuous monitoring of security measures, DevSecOps ensures that security is ingrained at every step of the software development journey.

But how does DevSecOps actually work? DevSecOps operates by leveraging automation, collaboration, and integration. By utilizing a variety of well-known tools, it enables developers, security teams, and operations professionals to work harmoniously. These tools, such as SonarQube, OWASP ZAP, Jenkins, GitLab, and Docker, facilitate automated security scans, vulnerability assessments, and code analysis, among other critical tasks.

Next, we will explore the myriad benefits that DevSecOps brings to the table. By integrating security from the outset, DevSecOps accelerates software delivery while ensuring robust security measures are in place. It fosters collaboration between teams, minimizes security vulnerabilities, and diminishes the risk of data breaches. Ultimately, DevSecOps empowers organizations to deliver secure software that meets the ever-evolving demands of the digital landscape.

Finally, we will turn our attention to the exciting career opportunities and paths that await those venturing into DevSecOps. With the increasing global demand for secure software

development, both locally and internationally, a plethora of opportunities arise. From DevSecOps engineers to security analysts and automation specialists, a diverse range of roles awaits those passionate about contributing to the secure digital future.

1. CAUSES FOR INTIATION OF DEVSECOPS

DevSecOps is a practice that combines development, security, and operations in order to integrate security into the software development process from the start. It addresses several software engineering problems that can arise, leading to the need for its initiation. Some of these problems include:

1. Security vulnerabilities: Traditional development practices often prioritize functionality over security, leading to the introduction of vulnerabilities in software. DevSecOps aims to identify and mitigate these vulnerabilities early in the development process.
2. Slow response to security issues: In traditional software development, security issues are often addressed as an afterthought. This can result in delayed responses to security threats, leaving systems exposed to potential attacks. DevSecOps promotes a proactive approach to security, enabling faster identification and resolution of security issues.
3. Lack of collaboration: In many organizations, there is a lack of collaboration between development, security, and operations teams. This can lead to miscommunication, misunderstandings, and delays in addressing security concerns. DevSecOps promotes crossfunctional collaboration, ensuring that security requirements are considered throughout the
development lifecycle.
4. Compliance challenges: Compliance with industry regulations and standards is a critical aspect of software development. However, ensuring compliance can be challenging, particularly when security measures are only assessed at the end of the development process. DevSecOps helps organizations incorporate security and compliance requirements early on, making it easier to meet regulatory obligations.
5. Inefficient security testing: Traditional development practices often rely on manual security testing, which can be time-consuming and error-prone. DevSecOps emphasizes the use of automated security testing tools and processes, enabling faster and more effective identification of security vulnerabilities.

2. WHAT IS DEVSECOPS?

Definition

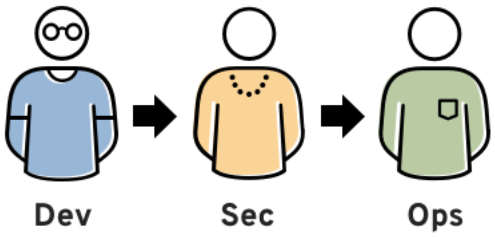
DevSecOps (short for development, security, and operations) is a development practice that integrates security initiatives at every stage of the software development lifecycle to deliver robust and secure applications

DevSecOps infuses security into the continuous integration and continuous delivery (CI/CD) pipeline, allowing development teams to address some of today's most pressing security challenges at DevOps speed.

Historically, security considerations and practices were often introduced late in the development lifecycle. However, with the rise of more sophisticated cybersecurity attacks, and development teams shifting to shorter, more frequent iterations on applications, DevSecOps is now becoming a go-to practice for ensuring applications are secure in this modern development ecosystem.

DevSecOps is a trending practice in application security (AppSec) that involves introducing security earlier in the software development life cycle (SDLC). It also expands the collaboration between development and operations teams to integrate security teams in the software delivery cycle. DevSecOps requires a change in culture, process, and tools across these core functional teams and makes security a shared responsibility. Everyone involved in the SDLC has a role to play in building security into the DevOps continuous integration and continuous delivery (CI/CD) workflow.

- DevSecOps (development plus security plus operations) is an approach that combines application development, security, operations and infrastructure as code (IaC) in an automated continuous integration/continuous delivery (CI/CD) pipeline.



- The main objective of DevSecOps is to automate, monitor and apply security at all phases of the software lifecycle: plan, develop, build, test, release, deliver, deploy, operate and monitor. Applying security at every stage of the software development

process supports CI/CD, reduces the cost of compliance and enables faster software delivery.

- DevSecOps means that every employee and team is responsible for security from the outset, and they must make decisions efficiently and put them into action without forfeiting security.

- DevSecOps is a tactical trifecta that connects three disciplines: development, security, and operations. The goal is to seamlessly integrate security into your continuous integration and continuous delivery (CI/CD) pipeline in both pre-production (dev/test/staging) and production (ops) environments. Let's take a look at each discipline and the role it plays in delivering better, more secure software faster.

Development

Development teams create and iterate on new software applications. This includes:

- Custom, built-in-house apps designed for a single, specific purpose
- API-driven connections that bridge the gap between legacy systems and new services
- Apps that leverage open-source code to accelerate the development process

Modern development practices rely on agile models that prioritize continuous improvement versus sequential, waterfall-type steps. If developers work in isolation without considering operations and security, new applications or features may introduce operational issues or security vulnerabilities that can be expensive and time-consuming to address.

Operations

Operations refers to the processes of managing software functionality throughout its delivery and use life cycle, including:

- Monitoring system performance
- Repairing defects
- Testing after updates and changes
- Tuning the software release system

DevOps has gained ground in recent years as a way to combine key operational principles with development cycles, recognizing that these two processes must coexist. Siloed post-development operations can make it easier to identify and address potential problems, but this approach requires developers to circle back and solve software issues before they can move

forward with new development. This creates a complex road map instead of a streamlined software workflow.

Implementing operations parallel to software development processes allows organizations to reduce deployment time and increase overall efficiency.

Security

Security refers to all the tools and techniques needed to design and build software that resists attack, and to detect and respond to defects (or actual intrusions) as quickly as possible.

Historically, application security has been addressed after development is completed, and by a separate team of people — separate from both the development team and the operations team. This siloed approach slowed down the development process and the reaction time.

Also, security tools themselves have historically been siloed. Each application security test looked only at that application, and often only at the source code of that application. This made it hard for anyone to have an organization-wide view of security issues, or to understand any of the software risks in the context of the production environment.

By making application security part of a unified DevSecOps process, from initial design to eventual implementation, organizations can align the three most important components of software creation and delivery.

3. DEVSECOPS LIFECYCLE

Steps in the Devsecops Lifecycle

DevSecOps is a software development methodology that emphasizes security and collaboration between development, security, and operations teams throughout the software development lifecycle. DevSecOps works best with teams that use CI/CD, or continuous integration and delivery process, meaning code changes are integrated and released as part of an automated process.

The DevSecOps lifecycle can be broken down into the following steps, with the development, testing, and deployment stages often happening in a loop as software updates are made and new features are added:

1. **Plan**-In the planning phase, development teams work with security and operations teams to identify potential security risks and develop a security strategy.

   This includes identifying security requirements, defining security policies, and selecting the appropriate security testing tools.

2. Develop-During the development phase, development teams both build and test the application. This includes integrating automated security testing into the development process, conducting code reviews, and ensuring that security requirements are met.

Since development and testing happen together in the DevSecOps lifecycle, less secure components, such as third-party code, can be tested as they are put into place.

This is where the continuous integration part of the CI/CD process comes in. Code changes are automatically integrated into a shared repository on a regular basis, allowing developers to identify and address conflicts and issues early in the development process.

Optional: Test

Since testing happens during development, a separate testing phase is not necessary in a DevSecOps approach. When it is included, testing takes much less time than it does in a traditional testing process.

During the testing phase, security teams test the application for security weaknesses, vulnerabilities, and threats using penetration testing, vulnerability scanning, and other security testing techniques.

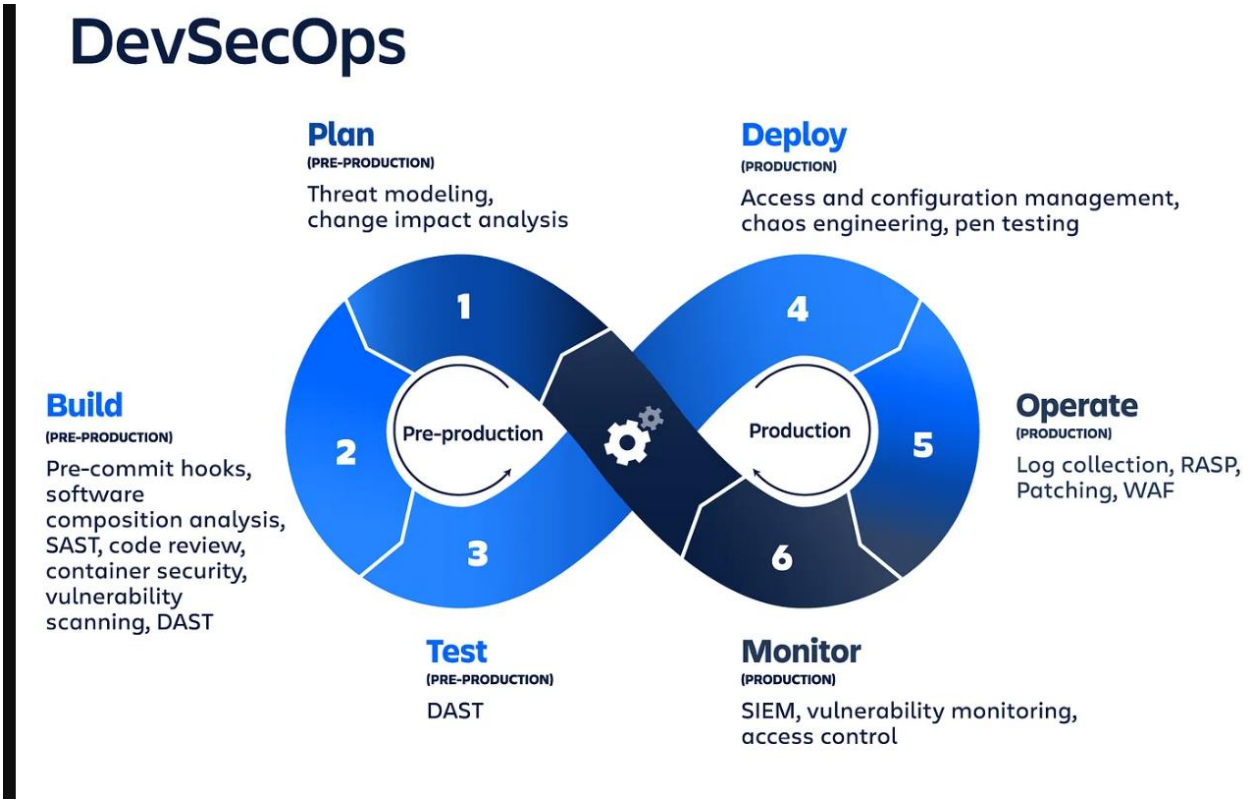
3. Deploy and Monitor

In a traditional process, the operation team would have deployed the application to production. However, the DevSecOps lifecycle follows the DevOps approach, which shifted the responsibility of deploying the application from operations teams to development teams.

The process of deploying to production includes configuring and securing the infrastructure, implementing access controls, and monitoring the environment for security threats.

Today, many development teams trigger deployments using continuous delivery. This involves the use of tools and processes to automatically build, test, and deploy code changes to production environments.

After deployment, teams then monitor the application for security threats and respond to any incidents that occur.



Additional Phases for DevSecOps

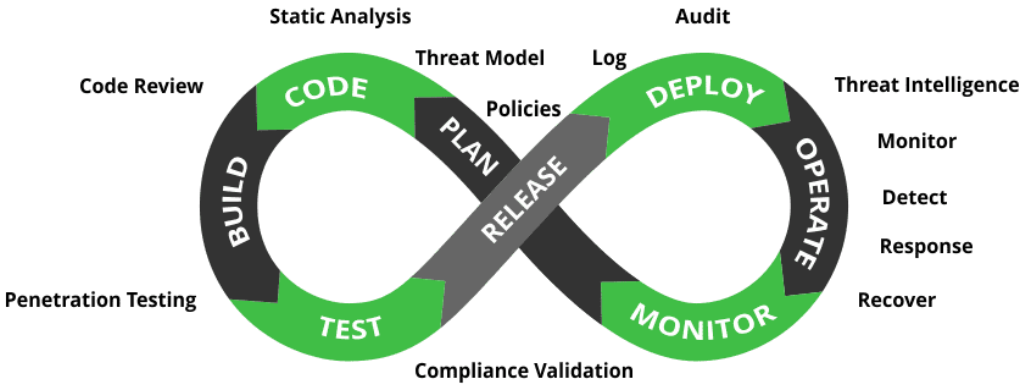
DevSecOps is the integration of security into a pipeline of continuous integration, continuous delivery, and continuous deployment. By infusing DevOps values into software security, security verification becomes an integral, active component of the development process.

Similar to DevOps, DevSecOps is an organisational and technological paradigm that blends automated IT technologies with project management workflows. DevSecOps incorporates active security audits and security testing into agile development and DevOps workflows so that security is incorporated into the product, as opposed to being added after the fact.

For teams to deploy DevSecOps, they must:

- Incorporate security throughout the software development lifecycle to decrease software code vulnerabilities.
- Ensure that the whole DevOps team, including developers and operational teams, is liable for adhering to security best practices.

- Facilitate automatic security checks at each level of software delivery by integrating security controls, tools, and processes into the DevOps workflow.



4. HOW DOES DEVSECOPS WORK?

DevSecOps is designed to provide development teams with the full security stack. This is achieved by establishing ongoing collaboration between development, release management (also known as operations), and the organization's security team and emphasizing this collaboration along each stage of the CI/CD Pipeline.

The CI/DI Pipeline is broken into six stages known as Code, Build, Store, Prep, Deploy and Run.

Each stage of the workflow is explained here to illustrate the benefits of embedding security early in the process.

- **Code**
  The first step to a development approach that aligns with DevSecOps is to code in segments that are both secured and trusted.
- **Build**
  To take code and deliver comprehensive container images that contain a core OS, application dependencies and other run-times services, requires a secure process.
- **Store**
  Any off-the-shelf technology stack needs to be considered a risk in today's ever-evolving cybersecurity landscape. To this point, each off-the-shelf app or back-end service should be continually checked.
- **Prep**
  Before deployment, organizations need to ensure their application complies with

security policies. To achieve this, VMware Tanzu and Carbon Black Cloud Container can validate configurations against the organization's security policies before entering subsequent stages of the development cycle. These configurations define how the workload should run, not only providing key insight into potential vulnerabilities but also setting subsequent stages of the CI/CD pipeline up for a successful deployment.

- **Deploy**
  Scans delivered in previous steps give organizations a comprehensive understanding of the application's security strength. Here, vulnerabilities or misconfigurations in the development process that has been identified are clearly presented allowing organizations to fix issues and define stronger security standards to promote a stronger security posture.
- **Run**
  As deployments run, SecOps teams can leverage active deployment analytics, monitoring and automation to ensure continuous compliance while also mitigating the risk of vulnerabilities that surface following deployment.

A typical DevSecOps workflow is as follows:

- Software is developed using a version control system.
- A different team member analyzes the changes made to the application for security weaknesses, overall code quality and possible bugs.
- The application is deployed within security configurations.
- Automation is used to test the application's back end, user interface, integrations and security.
- If the application passes the tests, it is moved to the production environment.
- In the production environment, various monitoring applications and security software monitor the application.

5. DEVSECOPS TOOLS

There are several families of security and compliance tools to address different aspects of the SDLC. This includes: Static Code Analysis (SAST), Software Composition Analysis (SCA), and different approaches for testing the code for vulnerabilities (DAST and IAST). In addition there are tools that are aimed to monitor and protect your binaries in production environments against attacks that exploit your code or your environment vulnerabilities. Ideally teams should aim to adopt all of these areas for complete SDLC security.

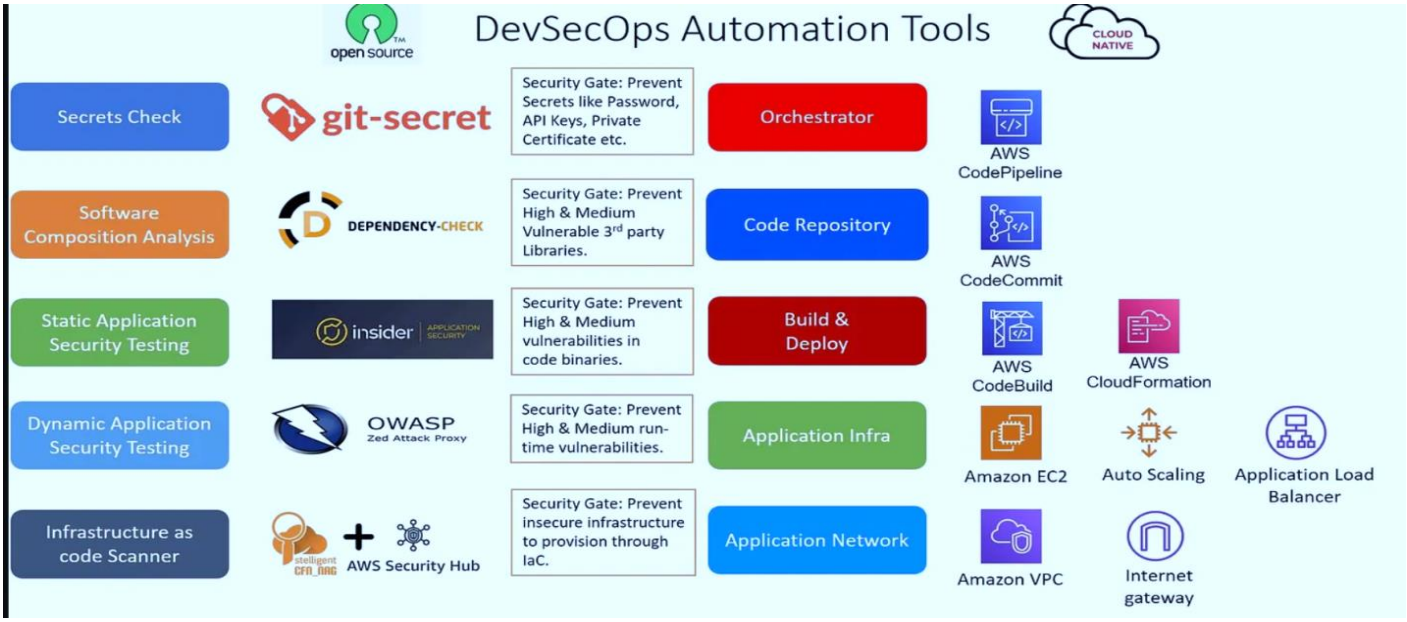
- **Static Application Security Testing (SAST)** tools can help you in identifying vulnerabilities in your own proprietary developed code. Developers should be aware of and use SAST tools as an automated part of their development process. This will help to detect and remediate potential vulnerabilities early on in the DevOps cycle.
- **Software Composition Analysis (SCA)** encompasses managing and monitoring license compliance and security vulnerabilities in the open source components your code depends on. Knowing what OSS components are being used and what their dependencies are of primary concern. After identifying the open source components, SCA tools such as JFrog Xray, will provide information on licenses and whether there are any known security vulnerabilities associated with these components. Advanced SCA tools offer policy enforcement capabilities, preventing the download of binaries, failing builds, and notifying other systems.
- **Dynamic and Interactive Application Security Testing (DAST and IAST)** tools test the running application's exposed interfaces, looking for vulnerabilities and flaws. While DAST looks at the application as a black box, IAST uses instrumentation that combines dynamic application security testing (DAST) and static analysis security testing (SAST) techniques to increase the accuracy of application security testing.
- **Container Runtime Security** tools monitor the containers in their runtime environment. Such tools provide different abilities including – fire walling on different levels, identifying anomalies based on behavioral analytics and more.

DevSecOps tools include the following:

- Security scanning tools to detect vulnerabilities
- Container security solutions to secure containers
- CI/CD tools to streamline the development process
- Software composition analysis tools to identify open source components
- Secure coding tools to ensure code quality
- Security policy management tools to ensure that policies are enforce

Examples of DevSecOps tools

- **ThreatModeler** is an automated threat modeling tool that can be deployed on premises or in a cloud instance. ThreatModeler continuously monitors threat models for cloud computing environments, notifying users of updates and changes. ThreatModeler provides a bidirectional API to integrate with CI/CD tools, enabling teams to build secure cloud infrastructures. ThreatModeler offers reusable templates and built-in threat information and frameworks.
- **Acunetix** is a web security scanner intended to help developers find vulnerabilities as early in the development cycle as possible. Acunetix enables organizations to protect their web assets from hackers by providing specialized technologies that developers can use to detect and fix issues.
- **Checkmarx** offers a static application security testing (SAST) tool that scans for security vulnerabilities in code. This tool helps developers deliver secure, reliable applications by incorporating code security analysis and testing into the development process. Checkmarx integrates with a variety of CI/CD tools and environments.
- **Aqua Platform** from Aqua Security is an application security tool for containers and their infrastructures designed to prevent intrusions and vulnerabilities throughout the DevSecOps pipeline. Aqua implements runtime security processes and controls and focuses on vulnerabilities related to network access and application images. Aqua integrates with a variety of infrastructures, including Kubernetes, to secure clusters at the lowest network level and control container activity in real time using behavior profiles based on machine learning.



6. WHAT ARE THE BENEFITS OF DEVSECOPS?

Security is top of mind for every organization today. Fortunately, DevSecOp's emphasis on incorporating security at every stage is proving to be a more secure approach to development while meeting the velocity of today's rapid release cycle.

The DevSecOps approach brings with it specific benefits:

- **Enhanced Application Security**
  DevSecOps embeds a proactive approach to mitigate cybersecurity threats early in the development lifecycle. This means that development teams will rely on automated security tools to test code on the fly, performing security audits without slowing development cycles.

  DevOps teams will review, audit, test, scan, and debug code at various stages of the development process to ensure the application is passing critical security checkpoints. When security vulnerabilities are exposed, application security and development teams will work collaboratively on solutions at the code level to address the problem.

- **Cross-team ownership**
  DevSecOps brings development teams and application security teams together

early in the development process, building a collaborative cross-team approach. Rather than siloed, disparate operations that stifle innovation and even lead to division among business units, DevSecOps empowers teams to get on the same page early, leading to cross-team buy-in, and more efficient team collaboration.

- **Streamline Application Delivery**
  Embed security earlier and often the development lifecycle, automate as many security processes as possible and streamline reporting all enhance security and enables compliance teams, ensuring that security practices embolden fast development cycles.

  For example, suppose a development team completes all the initial development stages of an application, only to find that there is an array of security vulnerabilities right before bringing the application to production. In that case, this can result in a major delay in delivery.

- **Limit Security Vulnerabilities**
  Leverage automation to identify, manage, and patch common vulnerabilities and exposures (CVE). Use pre-built scanning solutions early and often to scan any prebuilt container images in the build pipeline for CVEs. Introduce security measures that not only mitigate risk but also provide insight to teams so that teams can remediate quickly when vulnerabilities are discovered.

  One of the strongest benefits of DevSecOps is it creates a streamlined agile development process - an approach that if done correctly can greatly limit security vulnerabilities. Many of the cybersecurity testing processes, tasks, and services integrate quite easily with the automated services found in an application development or operations team.

  By emphasizing a security-first approach to the development process, organizations can remove unknown variables that will undoubtedly influence the product release timelines.

The benefits of adopting DevSecOps include the following:

- improved quality and security of software;
- faster software delivery;
- enhanced communication and collaboration between teams;
- faster recovery from security incidents;
- better cloud service deployments with strong security protocols;
- faster response to ever-changing customer needs;
- earlier identification and correction of vulnerabilities in code;
- increased use of automation, especially in quality control testing; and
- more opportunities for automated builds and quality assurance.

DevSecOps can improve the overall security of software with benefits such as:

- Increased security: By integrating security into the DevOps process, DevSecOps can help to prevent security vulnerabilities from being introduced into production systems.
- Reduced risk: Reduce the risk of security and data breaches.

- Improved compliance: Automate processes that will help enforce compliance with security regulations..
- Improved efficiency: Improve the efficiency of the software development process by automating security checks and scans.
- Improved compliance: Help organizations to comply with security regulations.
- Increased collaboration: Improve collaboration between development, operations, and security teams, through a shared sense of responsibility
- Faster time to market: Speed up the software development process by automating security checks and scans.

- Improved quality: Improve software quality by catching security vulnerabilities early in the development process.
- Improved risk management: Help organizations identify and address security risks more effectively.
- Increased customer satisfaction: Increase customer satisfaction by delivering secure and reliable software.
- Reduced costs: Reduce the costs associated with security and data breaches.
- Improved visibility: Help organizations gain visibility into their security posture to quickly identify and address security risks.

7. LOCAL AND INTERNATIONAL DEVSECOPS CAREER OPPORTUNITIES AND CAREER PATH

DevSecOps, which stands for development, security, and operations, is a growing field in the technology industry that focuses on integrating security practices within the DevOps process. If you are considering a career path in DevSecOps, there are opportunities available both locally and internationally.

On a local level, you can explore job opportunities with various companies in your region that are looking to enhance their security measures within their development and operations teams. This can be a great way to gain practical experience and build a network within your community.

Internationally, the demand for DevSecOps professionals is rapidly increasing as organizations prioritize cybersecurity in their software development lifecycle. By positioning yourself as an expert in DevSecOps, you can open doors to exciting career opportunities in different countries and industries.

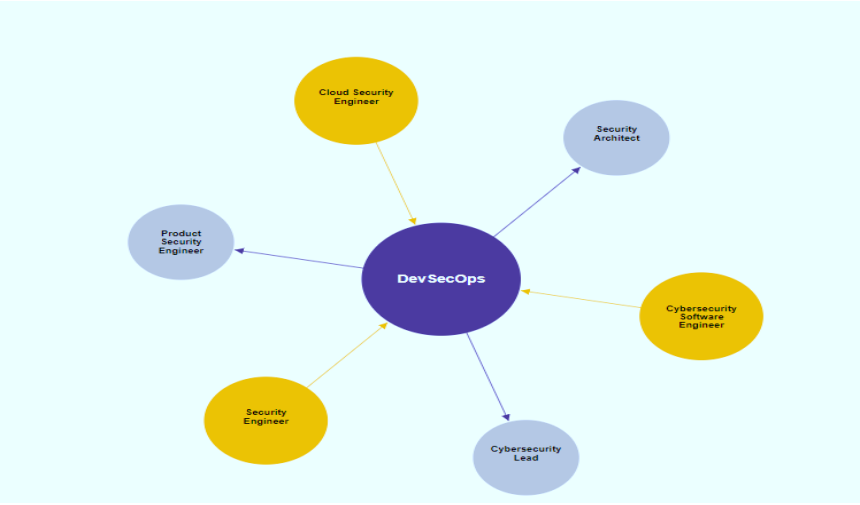
To embark on a successful career path in DevSecOps, consider earning relevant certifications,staying updated on the latest industry trends, and gaining hands-on experience with security

tools and practices. Building a strong foundation in both development and security will make you a valuable asset to any organization, whether local or international.

Types of jobs in DevSecOps

You'll find many types of jobs in which you can build a career in DevSecOps. For example, you could become a developer, a tester, an operations engineer, or a security analyst. Here are some roles advertised in DevSecOps environments and their average annual salaries.

- DevSecOps engineer: $116,2351 [2]
- DevSecOps software engineer: $124,195 [3]
- Cloud security engineer: $102,939 [4]
- Cloud and DevSecOps architect: $133,059 [5]
- Senior DevSecOps engineer: $124,258 [6]
- DevSecOps lead: $126,731 [7]



Summary:

- DevSecOps was initiated to address software engineering problems such as security vulnerabilities, slow release cycles, and lack of collaboration between development, security, and operations teams.
- DevSecOps is a software engineering approach that integrates security practices into the DevOps lifecycle, aiming to ensure the security of software throughout its development, deployment, and operation.
- The DevSecOps lifecycle involves several stages: planning, coding, building, testing, releasing, deploying, operating, and monitoring. Each stage incorporates security practices to identify and address vulnerabilities and risks.
- DevSecOps works by integrating security practices and tools into the DevOps workflow. It involves automating security testing, incorporating security controls into the development process, and fostering collaboration between development, security, and operations teams.
- Some well-known DevSecOps tools include OWASP ZAP, SonarQube, Veracode, Snyk, and Twistlock. These tools help in identifying security vulnerabilities, performing code analysis, and ensuring secure software development practices.
- The benefits of DevSecOps include improved software security, faster release cycles, reduced security risks, increased collaboration between teams, and enhanced overall software quality.
- DevSecOps offers both local and international career opportunities. Professionals with expertise in DevSecOps can pursue careers as DevSecOps engineers, security analysts, security architects, or security consultants. The career path in DevSecOps typically involves gaining experience in software development, security practices, and operations, along with acquiring relevant certifications.

References

https://www.vmware.com/topics/glossary/content/devsecops.html?resource=cat-1696897914

https://www.synopsys.com/glossary/what-is-devsecops.html

https://www.techtarget.com/searchitoperations/definition/DevSecOps

https://jfrog.com/devops-tools/what-is-devsecops/

https://about.gitlab.com/topics/devsecops/

https://www.dynatrace.com/news/blog/what-is-devsecops/

https://blog.gitguardian.com/devsecops-introduction-accelerating-software-development/

https://www.udemy.com/course/devsecops-foundation-course/

https://www.mayhem.security/blog/the-devsecops-lifecycle-how-to-automate-security-in-software-development

https://www.atlassian.com/devops/devops-tools/devsecops-tools

https://ranjaniitian.medium.com/top-devsecops-tools-for-2023-open-source-solutions-for-enterprises-7c146f80b325

https://cybersn.com/role/devsecops/