# Institution of Technology

## Department of software Engineering

## Assignment of software Engineering Tools and Practices

**Name: ETSUBDINK DESALEGN**
**ID: 146018**
**Section: One**

**Submitted to: ESMAIL M**
**May/29/2024G.C**

**Woldia, Ethiopia**

## Contents

# Introduction

DevSecOps is a compound of Development, Security, and Operations, is a methodology that integrates security practices into the DevOps process. This synergistic approach aims to create a culture of shared responsibility for security, where teams collaborate seamlessly to deliver secure and reliable software at a faster pace.

In this document, we delve into the meaning and significance of DevSecOps, exploring how it transforms traditional approaches to software development by emphasizing security from the outset. We will discuss the key benefits of implementing DevSecOps, such as enhanced security posture, faster time- tomarket, and improved collaboration among teams.

Furthermore, we will examine the lifecycle of DevSecOps, outlining the stages involved from planning and development to testing, deployment, and monitoring. Understanding the lifecycle is crucial for organizations looking to adopt DevSecOps practices effectively.

Lastly, we will explore the tools and technologies that support DevSecOps implementation. These tools play a pivotal role in automating security processes, enabling continuous integration and deployment, and facilitating the seamless integration of security into every phase of the software development lifecycle.

Join us on this journey as we unravel the essence, advantages, lifecycle, and tools of DevSecOps, and discover how it can revolutionize your approach to software development and security.

# 1. Software Engineering Problems Cause For Initiation Of DevSecOps

These software engineering problems highlighted the need for a more comprehensive and secure approach to software development and deployment. DevSecOps emerged as a response to these challenges, promoting the integration of security practices throughout the entire software development lifecycle. By addressing these problems, DevSecOps aims to enhance the security, reliability, and resilience of software systems in an increasingly complex and dynamic threat landscape.

**Some of the key issues that led to the emergence of DevSecOps include:**

- **Lack of Security Awareness**

One of the main causes for the initiation of DevSecOps is the lack of security awareness among software engineers. Traditional software development processes often prioritize functionality and speed over security, leading to vulnerabilities in the code. DevSecOps aims to integrate security practices into every stage of the software development lifecycle, ensuring that security is considered from the outset.

- **Increasing Cyber security Threats**

With the rise of cyber-attacks and data breaches, organizations are realizing the importance of prioritizing security in their software development processes. DevSecOps helps to address security vulnerabilities early on in the development process, reducing the risk of security incidents and minimizing the impact of potential breaches.

- **Compliance Requirements**

Many industries have strict regulatory requirements around data privacy and security, such as GDPR or HIPAA. Failure to comply with these regulations can result in hefty fines and damage to an organization's reputation. DevSecOps helps organizations ensure compliance by integrating security controls and monitoring mechanisms into their development processes.

- **Complexity of Modern Software Systems**

Modern software systems are becoming increasingly complex, with multiple interconnected components and dependencies. This complexity makes it difficult to identify and mitigate security vulnerabilities in the code. DevSecOps helps organizations manage this complexity by automating security testing, monitoring, and remediation processes throughout the development lifecycle.

- **Shift Left Approach**

DevSecOps promotes a "shift left" approach to security, which means integrating security practices early on in the development process. By addressing security concerns at the beginning of the development lifecycle, organizations can identify and fix vulnerabilities before they become costly to remediate. This proactive approach to security is essential in today's fast-paced software development environment.

- **Security Vulnerabilities**

Traditional software development methodologies often prioritized speed and functionality over security, leading to vulnerabilities that could be exploited by malicious actors.

- **Lack of Collaboration Between Teams**

Development, security, and operations teams often worked in isolation, resulting in communication breakdowns and delayed responses to security concerns.

- **Slow Response to Security Incidents**

Traditional security practices relied on reactive approaches to address security incidents after they had already occurred, leaving organizations vulnerable to prolonged exploitation.

By addressing these software engineering challenges through the adoption of DevSecOps practices, organizations aim to overcome these issues and build a more secure and resilient software development process. DevSecOps strives to embed security into every phase of development, promoting a culture of security awareness, collaboration, and continuous improvement.

## 2. What is DevSecOps

DevSecOps is a set of practices and principles that aim to integrate security into every stage of the software development lifecycle, from initial design to deployment and maintenance. The term "DevSecOps" is a combination of "Development" (Dev), "Security" (Sec), and "Operations" (Ops), highlighting the need to incorporate security considerations into the DevOps approach to software development and delivery.

DevSecOps is a software development approach that integrates security into the software development lifecycle from the beginning. It involves close collaboration between development, security, and operations teams, as well as the use of automated tools and processes to improve efficiency and security.

DevSecOps emphasizes collaboration and communication between development, security, and operations teams, with the goal of creating a culture of shared responsibility for security. This involves automating security testing, integrating security controls into the development process, and fostering a mindset of continuous security improvement.

### Key aspects of DevSecOps :

**1. Automation:** DevSecOps encourages the automation of security processes, such as vulnerability scanning, code analysis, and compliance checks. By automating these tasks, organizations can identify and address security issues more efficiently and consistently.

**2. Shift Left:** DevSecOps promotes a "shift left" approach to security, meaning that security considerations are addressed early in the development process. This proactive stance helps to identify and mitigate security vulnerabilities at an earlier stage, reducing the risk of security incidents later in the lifecycle.

**3. Continuous Monitoring:** DevSecOps advocates for continuous monitoring of applications and infrastructure to detect and respond to security threats in real time. This includes monitoring for anomalous behavior, potential breaches, and compliance violations.

**4. Collaboration:** DevSecOps emphasizes collaboration between development, security, and operations teams to ensure that security is integrated seamlessly into the development process. This collaborative approach fosters a shared understanding of security risks and responsibilities across teams.

**Key principles of DevSecOps include:**

- Security is everyone's responsibility.
- Security should be integrated into the development process.
- Automation is key.
- Collaboration is essential.

**DevSecOps has a number of benefits, including:**

- Improved security
- Faster and more efficient development
- Reduced costs
- Improved collaboration
- Increased compliance

**Examples of DevSecOps practices:**

- Implementing automated security testing tools
- Conducting regular security audits
- Enforcing code review policies
- Using secure coding practices
- Collaborating with security teams to identify and address security risks

Overall, DevSecOps is an essential approach for organizations that want to improve the security of their software development process and reduce the risk of security breaches.DevSecOps represents a cultural and technical shift in how organizations approach software development, placing a strong emphasis on proactive security measures throughout the entire development lifecycle. By integrating security into DevOps practices, organizations can build more secure, resilient, and compliant software systems.

## 3. Briefly Explain DevSecOps Lifecycle

The DevSecOps lifecycle is a continuous and iterative process that integrates security practices into every stage of the software development lifecycle. It aims to ensure that security is considered and addressed from the initial design phase to deployment and ongoing maintenance.

Here is a detailed explanation of the DevSecOps lifecycle:

**Planning**

In the planning phase, security considerations are incorporated into the overall project plan. This involves identifying security requirements, defining security objectives, and establishing security policies and guidelines.

**Design**

During the design phase, security controls and measures are integrated into the architecture and design of the software. This includes considering secure coding practices, secure configuration management, and data protection mechanisms.

**Development**

In the development phase, security practices are implemented by following secure coding standards pand best practices. Developers use secure coding techniques, such as input validation, output encoding, and proper error handling, to prevent common vulnerabilities like injection attacks or cross-site scripting.

**Testing**

The testing phase focuses on verifying the security of the software through various types of testing. This includes static code analysis, dynamic application security testing (DAST), penetration testing, vulnerability scanning, and security code reviews. Automated tools can be used to identify vulnerabilities and security weaknesses.

### Integration

In the integration phase, security controls and processes are integrated into the continuous integration and continuous delivery (CI/CD) pipeline. Security tests and scans are automated and run as part of the build process to ensure that any new code or changes do not introduce security vulnerabilities.

### Deployment

During deployment, security controls are enforced to ensure secure configuration of the infrastructure and application environments. This includes secure network configurations, access controls, encryption, and monitoring systems for detecting and responding to security incidents.

### Operations

In the operations phase, ongoing monitoring and maintenance of the software are performed to identify and respond to any security threats or vulnerabilities. Continuous monitoring tools help detect anomalies, potential breaches, or compliance violations. Security patches and updates are applied regularly to address any known vulnerabilities.

### Feedback and Improvement

Throughout the entire lifecycle, feedback loops are established to gather insights from security incidents, audits, or user feedback. These insights are used to continuously improve security practices, refine security policies, update secure coding guidelines, and enhance security awareness training for developers and other stakeholders.

By following this DevSecOps lifecycle approach, organizations can proactively address security concerns at every stage of software development, fostering a culture of shared responsibility for security and enabling the delivery of more secure and resilient software systems.

## 4. How DevSecOps Works?

DevSecOps works by integrating security practices and principles into the DevOps process, which focuses on collaboration, automation, and continuous delivery. Here is how DevSecOps works:

- **Collaboration**

DevSecOps promotes collaboration between development, security, and operations teams to ensure that security is considered from the beginning of the software development lifecycle. By breaking down silos and fostering communication and teamwork, organizations can address security concerns more effectively.

- **Automation**

Automation plays a crucial role in DevSecOps by enabling security controls, tests, and scans to be integrated into the CI/CD pipeline. Automated security tools can help identify vulnerabilities, enforce security policies, and ensure compliance with security standards. This reduces manual effort, speeds up delivery, and improves the overall security posture of the software.

- **Continuous Monitoring**

DevSecOps emphasizes continuous monitoring of applications and infrastructure to detect security threats and vulnerabilities in real-time. Monitoring tools provide visibility into the security posture of the system, enabling teams to respond quickly to security incidents and take proactive measures to mitigate risks.

- **Shift Left**

DevSecOps promotes a "shift left" approach, which means addressing security early in the software development process. By incorporating security practices in the planning, design, and development phases, organizations can identify and remediate security issues before they become costly or impact the software's integrity.

- **Security as Code**

DevSecOps encourages treating security as code, where security policies, controls, and configurations are defined as code and managed alongside application code. This allows security measures to be version-controlled, automated, and deployed consistently across different environments.

- **Continuous Improvement**

DevSecOps is a continuous improvement process that involves gathering feedback, analyzing security incidents, and implementing lessons learned to enhance security practices. By continuously refining security processes, updating security tools, and providing security training to team members, organizations can strengthen their security posture over time.

- **Compliance and Governance**

DevSecOps ensures that security practices are aligned with regulatory requirements and industry standards. Compliance and governance are integrated into the development process to ensure that security controls are in place.

- **Feedback Loop**

DevSecOps relies on a feedback loop to continuously improve security practices. By collecting feedback from security incidents, testing results, and user feedback, teams can learn from past experiences and enhance security measures

In summary, DevSecOps works by integrating security practices into the DevOps process, emphasizing automation, collaboration, continuous monitoring, and feedback loops to build secure software applications. By adopting DevSecOps principles, organizations can create a culture of security and deliver secure software at a faster pace.

## 5. Explain well known DevSecOps tools

There are several well-known DevSecOps tools that help teams integrate security practices into their DevOps processes. Here are some of the popular tools used in DevSecOps:

## Static Application Security Testing (SAST) Tools

**1.Veracode**: Veracode is a cloud-based SAST tool that scans code for security vulnerabilities and provides detailed reports on potential issues.

**2. SonarQube**: SonarQube is an open-source platform for continuous code quality inspection that includes static code analysis for identifying security vulnerabilities.

**3. Bandit**: Bandit is a Python-focused SAST tool that analyzes Python code for common security issues and vulnerabilities.

**4. SpotBugs**: Find Bugs is an open-source static analysis tool for Java applications that detects common coding errors, potential vulnerabilities, and performance issues.

**5. RIPS**: RIPS is an open-source PHP security analysis tool that helps identify security vulnerabilities and coding flaws in PHP applications.

**6.PMD**: PMD is an open-source source code analyzer for various programming languages, including Java, JavaScript, and XML, which identifies potential bugs, dead code, and security

**7.Checkmarx**: Checkmarx is another SAST tool that analyzes source code to identify security vulnerabilities and compliance issues.

## Dynamic Application Security Testing (DAST) Tools:

**1.OWASP ZAP:** OWASP Zed Attack Proxy (ZAP) is an open-source DAST tool that helps identify security vulnerabilities in web applications by simulating attacks.

**2. Nikto:** Nikto is an open-source web server scanner that performs comprehensive tests against web servers to identify potential vulnerabilities.

**3. Wapiti:** Wapiti is an open-source web application vulnerability scanner that audits the security of web applications by performing black-box testing.

**4. Arachni:** Arachni is an open-source, modular web application security scanner that checks for a wide range of vulnerabilities and provides comprehensive reports.

**5. Grabber:** Grabber is an open-source web application scanner that detects security vulnerabilities by crawling and scanning web pages.

**6.Netsparker:** Netsparker is a commercial DAST tool that scans web applications for security vulnerabilities and provides detailed reports.

## Container Security Tools:

**1. Docker Bench for Security**: Docker Bench for Security is a script that checks for common best practices when deploying Docker containers to ensure they are secure.

**2. Clair:** Clair is an open-source container vulnerability scanner that analyzes container images and provides reports on known vulnerabilities.

**3. Trivy:** Trivy is an open-source vulnerability scanner for containers and other artifacts, such as operating system packages and application dependencies. It scans container images and provides detailed reports on any vulnerability detected, including their severity and remediation steps.

**4. Clair:** Clair is an open-source vulnerability scanner for containers that helps identify vulnerabilities in container images.

**5. Anchore Engine:** Anchore Engine is an open-source tool for analyzing container images for vulnerabilities, policy violations, and best practices.

**6. Sysdig Falco**: Sysdig Falco is an open-source behavioral activity monitoring tool designed specifically for containers and Kubernetes. It detects and alerts on anomalous behavior and potential security threats in real-time. Falco uses rules and policies to define expected container behavior and raises alerts when deviations occur

## Security Information and Event Management (SIEM) Tools:

**1. Splunk**: Splunk is a SIEM tool that collects, monitors, and analyzes security data from various sources to detect and respond to security incidents.

**2. LogRhythm**: LogRhythm is another SIEM tool that offers real-time monitoring, threat detection, and incident response capabilities.

## Compliance Tools:

**1. OpenSCAP**: OpenSCAP is a Security Content Automation Protocol (SCAP) framework for compliance checking, vulnerability management, and measurement.

**2. OpenVAS**: OpenVAS (Open Vulnerability Assessment System) is a full-featured vulnerability scanner that can detect security vulnerabilities in systems and network

**3. Wazuh**: Wazuh is an open-source host-based intrusion detection system (HIDS) that helps with compliance monitoring, file integrity monitoring, and log analysis.

## Infrastructure as Code (IaC) Security Tools:

**1. Terraform:** Terraform is an IaC tool that allows you to define and provision infrastructure using code. Security best practices can be incorporated into Terraform scripts to ensure secure infrastructure deployment.

**2. OpenSCAP:** OpenSCAP is an open-source framework for compliance checking and vulnerability management, which includes capabilities for assessing and securing infrastructure systems.

**3. Lynis:** Lynis is an open-source security auditing tool that assesses the security configuration of Linux and Unix-based system

**4. AWS Config:** AWS Config is a service that helps monitor and assess the configuration of AWS resources for compliance with security best practices.

**5. Dagda**: Dagda is an open-source container security analysis tool that performs static analysis of container images to detect security issues and vulnerabilities.

**6. ScoutSuite**: ScoutSuite is an open-source multi-cloud security auditing tool that assesses the security posture of containerized infrastructure in public cloud environments.

## Software Composition Analysis (SCA) Tools:

**1. OWASP Dependency-Check**: OWASP Dependency-Check is a software composition analysis tool that identifies known vulnerabilities in project dependencies.

**2. Retire.js:** Retire.js is a scanner that detects vulnerable JavaScript libraries in your web application.

**3. WhiteSource Bolt**: WhiteSource Bolt is an open-source SCA tool that scans your project dependencies for known vulnerabilities and provides actionable remediation steps.

**4. Dependency-Track**: Dependency-Track is an open-source platform that tracks and monitors your project's dependencies, providing insights into their known vulnerabilities.

**5. OSSIndex**: OSSIndex is an open-source vulnerability database and analysis platform that integrates with various development tools to provide real-time security intelligence on project dependencies

## Secrets Management Tools:

**1. HashiCorp Vault:** HashiCorp Vault is a secrets management tool that helps securely store and manage sensitive information such as passwords, API keys, and certificates.

**2. AWS Secrets Manager:** AWS Secrets Manager is a service provided by AWS for managing secrets and credentials securely.

## Dashboard Tools:

**1. Grafana**: Grafana is an open-source analytics and monitoring platform that allows you to create customizable dashboards for visualizing various metrics and data sources.

**2. Kibana:** Kibana is an open-source data visualization dashboard for Elasticsearch, used for exploring, analyzing, and visualizing data stored in Elasticsearch indices.

**3. Metabase:** Metabase is an easy-to-use open-source business intelligence and analytics tool that allows you to create dashboards and visualize data from various source.

### Vulnerability Tracking Tools:

**1. OWASP DefectDojo**: DefectDojo is an open-source vulnerability management tool that helps you track and manage vulnerabilities in your applications and infrastructure.

**2. TheHive: TheHive** is an open-source incident response and case management platform that includes features for tracking and managing vulnerabilities.

In conclusion, open-source tools play a crucial role in the field of cybersecurity, offering a wide range of solutions for different categories such as Software Composition Analysis (SCA), Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Container Security, and Infrastructure Security. These tools provide valuable support in identifying vulnerabilities, assessing security risks, and ensuring compliance.

## 6. What Are The Benefits Of DevSecOps?

- **Early Detection of Security Vulnerabilities:**

By integrating security practices into the development process, DevSecOps allows teams to identify and address security issues early in the software development lifecycle. This helps prevent vulnerabilities from being introduced into the codebase and reduces the cost and effort of fixing security flaws later in the development process.

- **Improved Collaboration and Communication:**

DevSecOps promotes collaboration between development, security, and operations teams. By breaking down silos and encouraging cross-functional communication, teams can work together to address security concerns throughout the development process, leading to more secure applications.

- **Faster Time to Market:**

Implementing security practices as part of the continuous integration/continuous deployment (CI/CD) pipeline allows teams to automate security checks and testing, reducing the time it takes to deliver secure software to market. This results in faster release cycles and improved agility.

- **Enhanced Compliance and Governance:**

DevSecOps helps organizations meet regulatory requirements and industry standards by incorporating security and compliance checks into the development process. This ensures that applications are developed in accordance with security best practices and regulatory guidelines.

- **Reduced Security Risks and Breaches:**

By integrating security into every phase of the software development lifecycle, DevSecOps helps reduce the risk of security breaches and data leaks. Proactive security measures, such as automated vulnerability scanning and secure coding practices, help mitigate potential risks.

- **Cost Savings:**

Addressing security issues early in the development process is more cost-effective than fixing them after deployment. DevSecOps helps minimize the impact of security incidents, reduces the likelihood of costly security breaches, and lowers the overall cost of managing security risks.

- **Continuous Monitoring and Improvement:**

DevSecOps emphasizes continuous monitoring of applications in production, enabling teams to quickly detect and respond to security threats. This approach fosters a culture of continuous improvement, allowing teams to adapt to evolving security challenges.

- **Customer Trust and Satisfaction:**

Building secure applications instills confidence in customers and users, leading to increased trust and satisfaction. DevSecOps practices demonstrate a commitment to delivering secure, reliable, and resilient software, which can positively impact customer relationships.

Overall, implementing DevSecOps practices leads to more secure, resilient, and reliable software while promoting collaboration, agility, and efficiency across development, security, and operations teams.

# 7. About Local and international DevSecOps career opportunities, career path.

## 7.1 Local DevSecOps Career Opportunities:

**Local DevSecOps career opportunities** are abundant in various industries, including finance, healthcare, retail, and technology. DevSecOps professionals play a critical role in integrating security practices into the software development lifecycle, ensuring that applications are secure, compliant, and resilient to cyber threats.

DevSecOps career paths often involve obtaining relevant certifications such as Certified DevSecOps Professional (CDSP) or Certified Information Systems Security Professional (CISSP), and gaining hands-on experience with security tools and automation frameworks. With experience and expertise, professionals can transition into leadership roles such as DevSecOps Manager or Chief Information Security Officer (CISO).

**Local DevSecOps Career Opportunities and Career Path:**

1. **Security Engineer:** Security engineers focus on implementing security measures in software development processes, including code reviews, vulnerability assessments, and security testing.
2. **DevSecOps Engineer:** DevSecOps engineers specialize in integrating security practices into the DevOps pipeline, automating security testing, and ensuring compliance with security standards.

3. **Security Analyst:** Security analysts monitor and analyze security threats, conduct risk assessments, and provide recommendations for improving security practices within an organization.
4. **Application Security Specialist**: Application security specialists focus on securing applications by implementing secure coding practices, conducting security assessments, and addressing vulnerabilities.

## 7.2 International DevSecOps Career Opportunities:

Internationally, DevSecOps career opportunities are prevalent in global enterprises, cyber security firms, and cloud service providers. Organizations worldwide are increasingly prioritizing security within their DevOps practices, creating a high demand for skilled professionals who can bridge the gap between development, operations, and security.

To excel in the international DevSecOps domain, professionals may pursue certifications such as Certified DevSecOps Engineer (CDSE) or Certified Cloud Security Professional (CCSP), and stay updated with the latest security trends and technologies. With experience and leadership capabilities, individuals can progress to executive roles such as Director of Security Operations or Global Head of DevSecOps.

**International DevSecOps Career Opportunities and Career Path:**

1. **Security Architect:** Security architects design and implement secure systems and applications, develop security policies and procedures, and provide guidance on security best practices.
2. **Cybersecurity Consultant:** Cybersecurity consultants offer expertise in evaluating and enhancing an organization's cyber security posture, conducting security assessments, and developing security strategies.
3. **Chief Information Security Officer (CISO):** CISOs are responsible for overseeing an organization's information security program, managing security initiatives, and ensuring compliance with security regulations.
4. **Security Operations Center (SOC) Analyst:** SOC analysts monitor and investigate security incidents, analyze security logs, and respond to cyber security threats in real-time.

In conclusion, local and international DevSecOps career paths offer diverse opportunities for professionals to contribute to the convergence of security and DevOps practices, safeguarding organizations against evolving cyber threats while enabling efficient software delivery.

### 7.3 DevSecOps Career Path:

1. **Entry-Level:** Start as a Security Analyst, Junior DevSecOps Engineer, or Security Intern to gain foundational knowledge in security practices and tools.
2. **Mid-Level:** Progress to roles such as DevSecOps Engineer, Security Engineer, or Application Security Specialist, focusing on integrating security into the development process and automating security testing.
3. **Senior-Level:** Advance to positions like Security Architect, CISO, or Cybersecurity Consultant, where you lead strategic security initiatives, design secure systems, and provide guidance on security governance.

To advance in a DevSecOps career path, professionals can pursue certifications such as Certified DevSecOps Professional (CDP), Certified Information Systems Security Professional (CISSP), or Certified Cloud Security Professional (CCSP) to demonstrate expertise in security practices and technologies. Continuous learning, hands-on experience, and staying updated on industry trends are essential for success in the dynamic field of DevSecOps.

## Conclusion

DevSecOps represents a paradigm shift in the world of software development, where security is not an afterthought but an integral part of the entire development process. By weaving security practices into the fabric of DevOps, organizations can reap a myriad of benefits, ranging from improved security posture to accelerated delivery cycles and enhanced collaboration among disparate teams.

Throughout this document, we have explored the meaning and essence of DevSecOps, underlining its pivotal role in fostering a culture of shared responsibility and proactive security measures. We have also delved into the array of benefits that come with embracing DevSecOps, including heightened resilience against cyber threats, enhanced customer trust, and increased efficiency in software delivery.

Understanding the lifecycle of DevSecOps is paramount for organizations embarking on this transformative journey. From the initial planning stages through development, testing, deployment, and monitoring, each phase of the lifecycle presents unique opportunities to embed security into the software delivery pipeline.

Lastly, the tools and technologies that support DevSecOps implementation serve as enablers for seamless integration of security into every step of the development lifecycle. Automation, continuous monitoring, and actionable insights provided by these tools empower organizations to proactively address security vulnerabilities and deliver robust and secure software solutions.

As we conclude our exploration of DevSecOps, it is evident that this methodology presents a holistic approach to software development that prioritizes security without compromising speed and agility. Embracing DevSecOps is not just a choice; it is a strategic imperative for organizations looking to stay ahead in today's rapidly evolving digital landscape. By adopting DevSecOps principles, organizations can build a strong foundation for secure, reliable, and resilient software solutions, safeguarding their assets, reputation, and stakeholders in an increasingly interconnected world.

## References

1. www.microsoftoffice.com
2. www.springboadrd.com/blog/software-engineering/what-is-devsecops
3. www.atlassian.com/devops-tools/devsecops-tools

4. www.practical-devsecops.com/devsecops-life-cycle/
5. https://about.gitlab.com/topics/devsecops
6. https://www.mavhem.securitv/blog/the-devsecops-lifecycle-how-to-automate-securitv-in-softwaredevelopment. "lifecycle of devsecops"