# WOLDIA UNIVERSITY

## INSTITUTE OF TECHNOLOGY SCHOOL OF COMPUTING

## DEPARTMENT OF SOFTWARE ENGINEERING

Course Title : Software Engineering Tools And Practice

Course Code:SEng3051

Assignment title:- DevSecOps

Third(3rd) Year First(1rst) Semester Software Engineering Individual Assignment

| Name | Id No |
|---|---|
| 1. YOHANNES ALEMAYEHU | 1303096 |

Submitted to:-**Mr. Esmael M.**

Deadline date:- **15/03/2024**

## Table content

Content page

# introduction

development security operations(DevSecOps) is a new word that has emerged in the software industry as a result of rising cybercrime and cyber security risks in recent years. It is essential for developers and businesses to implement DevSecOps in order to stay up with the demands of contemporary application and software development. it is a trending practice application security that involves introducing security earlier in the software development life cycle. It also expands the collaboration between development and operations teams to integrate security teams in software delivery cycle. It requires a change in culture, process, and tools across these core functional teams and makes security a shared responsibility.

devSecOps have phase planning, code(build), **Build and Continuous Integration, Test, Deploy and Operate, monitoring, scaling, Respond and Recover, Securely End of Life**

well known tools of DevSecOps are: codacy,aunetix,sonarQube, gitlab, aqua security.

DevSecOps works by integrating security practices into every phase of the software development lifecycle (SDLC), ensuring that security is prioritized, automated, and embedded early in the development process.

DevSecOps, the integration of security practices into DevOps processes, has become essential in modern software development to ensure secure and resilient applications. Both local and international job markets offer various career opportunities and career paths in DevSecOps.

## 1. Software engineering problems which was cause for in initiation of DevSecOps

The initiation of DevSecOps was driven by addressing various software engineering problems and shortcomings in traditional software development practices. Some of the key software engineering problems that led to the emergence of DevSecOps include:

1. Silos Between Development, Operations, and Security Teams: -Traditional software development often had distinct silos between development, operations, and security teams. This lack of collaboration and communication hindered the integration of security practices early in the software development lifecycle (SDLC), leading to security vulnerabilities and delays in addressing security issues.

2. Late Identification of Security Issues: - Security vulnerabilities and issues were often identified late in the software development process, such as during testing or in production. This late discovery increased security risks and the cost of remediating these issues.

3. Manual Security Reviews and Testing: - Traditional software development relied heavily on manual security reviews and testing, which were time-consuming, error-prone, and inadequate for addressing the scale and complexity of modern software applications.

4. Lack of Security Awareness Among Developers: - Developers often lacked the necessary security knowledge and skills to proactively address security concerns during development. This led to the introduction of vulnerabilities in code and the need for reactive security measures.

5. Security as an Afterthought: - Security was commonly treated as an afterthought in the software development process, with a primary focus on functionality and speed of delivery. This approach resulted in vulnerable software that was susceptible to various cyber threats.

6. Inefficient Security Processes: - The disconnect between security and development slowed down the security review process, leading to delays in addressing security issues and releasing secure software. Manual security processes were often cumbersome, hindering agile and continuous delivery practices.

7. Increasing Security Threat Landscape: - The growing number and sophistication of cyber threats, along with stringent regulatory requirements, necessitated a shift towards integrating security into every phase of the software development lifecycle.

DevSecOps emerged as a response to these software engineering challenges by promoting a culture of collaboration, automation, and shared responsibility among development, operations, and security teams. By embedding security practices early in the SDLC, automating security processes, and fostering a secure-by-design approach, DevSecOps aims to address these longstanding problems and ensure the delivery of secure and resilient software applications.

## 2. What is DevSecOps?

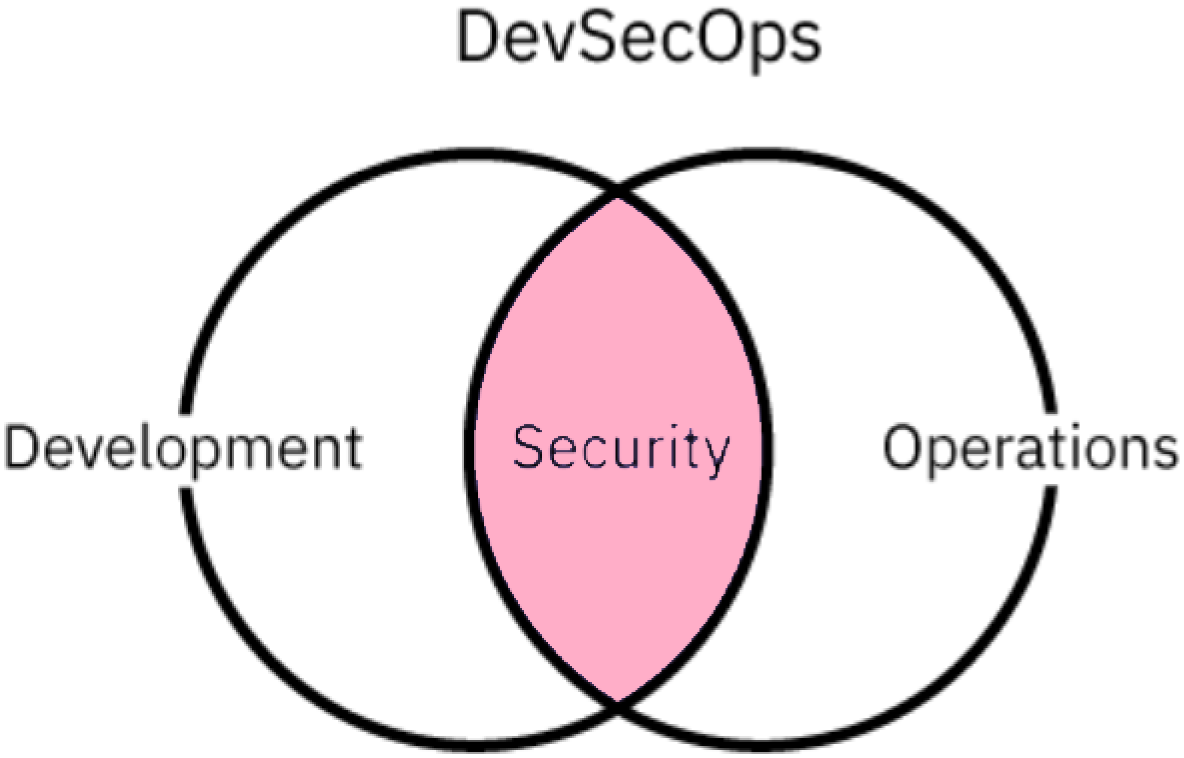
DevSecOps is a strategy for incorporating safety protocols into the DevOps procedure. It fosters and encourages collaboration among security staff and launch technicians based on the 'Security as Code' ideology. Considering the ever-increasing vulnerabilities to software programs, DevSecOps has increased in popularity and significance.

It is also identified as **"Development Security Operation."** DevSecOps is a recursive system that integrates protection into your product pipeline. It extensively integrates safety into the majority of the Development Operation (DevOps) methodology.

It is crucial for software development teams to evaluate for potential threats and weaknesses. Until the alternative can be implemented, security professionals must tackle problems. This incremental methodology guarantees that security flaws are highlighted.

As a current and innovative restraint, DevSecOps may take some chance to accumulate universal attention and assimilation. Late in the manufacturing process, a substantial set of security procedures are conducted. This postponement can have severe consequences for businesses and their product lines. Safety is typically among the last characteristics to be considered during the development phase. When safety problems occur near release, if protection is kept after the development pipeline, you will discover yourself back at the beginning of lengthy development processes.

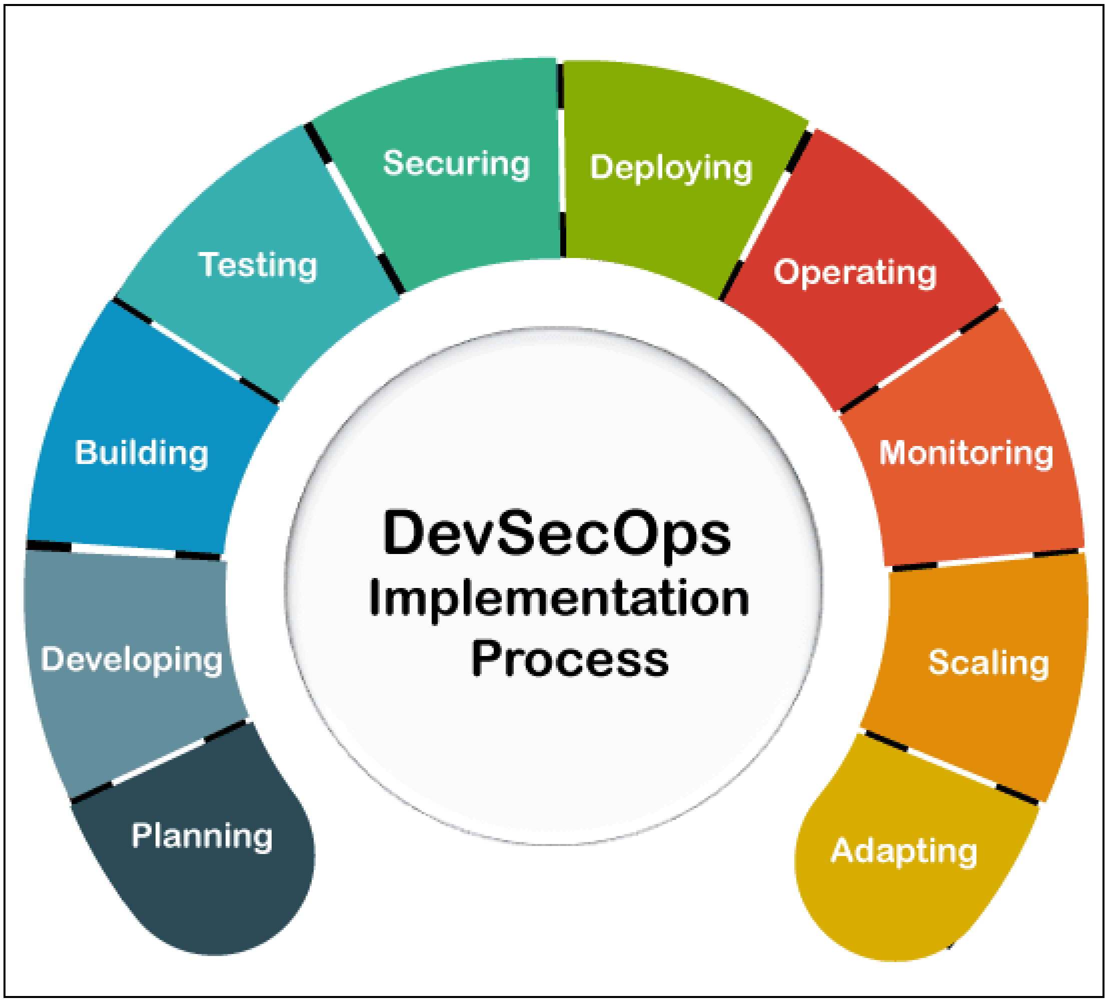
Development, security, and operations all make up the acronym "devSecOps." It is the inclusion of security right away in the process of developing software or an application.



Security was previously added to an application late in the lifecycle, following the development phase. The traditional development approach was hampered by the development of cloud platforms, microservices, and containers. As developers adopted agile and DevOps practices for modern application development and deployment, security was unable to keep up with the rapid releases.

By addressing issues as they arise within the Continuous Integration (CI) and Continuous Delivery (CD) pipelines, DevSecOps integrates security with DevOps.

## 3. DevSecOps Lifecycle



The DevSecOps lifecycle integrates security practices into every phase of the software development lifecycle (SDLC) to ensure that security is not an afterthought but a core part of the development process. Here is an overview of the DevSecOps lifecycle stages:

1. Plan and Design:

- Security Requirements: Identify security requirements based on compliance, industry standards, and the application's sensitivity.

- Threat Modeling: Conduct threat modeling to anticipate potential threats and vulnerabilities.

- Security Architecture: Define secure architecture and design principles, considering security best practices.

2. Develop:

- Secure Coding: Implement secure coding practices, such as input validation, output encoding, and proper session management.

- Static Code Analysis: Perform static code analysis to identify security vulnerabilities early in the development phase.

- Code Reviews: Conduct regular code reviews focusing on security aspects to identify and remediate security issues.

3. **Build and Continuous Integration**:- Automated Build Process: Implement an automated build process that includes security testing tools.

Continuous Integration: Integrate security scans, code analysis tools, and security tests into the CI/CD pipeline.

4. **Test**:- Dynamic Application Security Testing (DAST): Conduct DAST to identify vulnerabilities in running applications.

Security Testing Automation: Automate security testing processes, including vulnerability scanning and penetration testing.

Security Testing Tools: Utilize security testing tools like SAST, DAST, SCA, and fuzz testing tools.

5. **Deploy and Operate**:- Container Security: Ensure container security by scanning container images for vulnerabilities before deployment.

Infrastructure Security: Implement infrastructure as code (IaC) security practices for secure cloud and on-premise environments.Continuous Monitoring: Implement security monitoring and logging to detect and respond to security incidents.

- **Operating and monitoring**
  The operations team must do routine updates and monitoring, paying close attention to find zero-day vulnerabilities (software flaws that are publicly disclosed but may be discovered by attackers before the necessary patches are released). DevSecOps' ongoing security helps to avert such problems.

- **Scaling**
  Large data centers are no longer necessary for enterprises to maintain because of cloud computing and virtualization technologies. They may simply replace it in the case of a specific danger or grow their IT infrastructure as needed. Security of corporate communications is particularly relevant to this.
- **Adapting**
  DevSecOps requires ongoing improvement, just like any other business process, to make sure it is operating as it should. This entails assessing procedures and adapting to accommodate shifting trends and boost development.

6. **Respond and Recover:**- Incident Response: Have incident response plans in place to address security breaches promptly.

Forensics and Analysis: Conduct post-incident forensics and analysis to understand the root cause and prevent future incidents.

- Continuous Improvement: Apply lessons learned from incidents to improve security processes continually.

**7. Securely End of Life:**- Data Sanitization: Ensure secure data removal or sanitization when retiring applications.

Environment Cleanup: Remove access rights, credentials, and sensitive data from production and testing environments.

Key Aspects of the DevSecOps Lifecycle:- Automation: Implement automation and security testing tools to enforce consistent security practices.

Continuous Improvement: Embrace a culture of continuous learning and improvement to stay ahead of evolving security threats.

Shared Responsibility: Foster collaboration and shared responsibility among development, operations, and security teams.

Shift Left: Embed security as early as possible in the SDLC to identify and remediate security issues sooner.

The DevSecOps lifecycle is iterative, emphasizing the need for ongoing security measures and continuous integration of security practices to build and maintain secure software applications throughout their lifecycle.

## 4. DevSecOps works

DevSecOps works by integrating security practices into every phase of the software development lifecycle (SDLC), ensuring that security is prioritized, automated, and embedded early in the development process. Here is how DevSecOps works in practice:

1. Collaboration and Culture:- Cross-Functional Teams: DevSecOps promotes collaboration between development, operations, and security teams, breaking down silos for shared responsibility.

Culture of Security: Fosters a culture where security is everyone's responsibility, emphasizing security awareness and knowledge across the organization.

2. Shift Left Approach:- Early Security Integration: Shifts security practices and testing to the left, meaning security is integrated from the beginning of the SDLC, enabling early detection and remediation of security vulnerabilities. Automated Security Testing: Introduces automation for security testing in the early stages of development, such as static code analysis, dynamic application security testing (DAST), and software composition analysis (SCA).

3. Automation and Tooling:- Continuous Security Testing: Automates security testing in the CI/CD pipeline, enabling fast feedback on security issues and vulnerabilities.

Infrastructure as Code (IaC) Security: Implements security checks in IaC templates to ensure secure cloud infrastructure provisioning.

4. Security Controls and Compliance:- Security Monitoring and Logging: Implements continuous security monitoring and logging to detect and respond to security incidents in real time.

Compliance Automation: Integrates compliance checks into the deployment process to ensure regulatory standards are met automatically.

5. Secure Development Practices:- Secure Coding Guidelines: Enforces secure coding practices and guidelines to prevent common security vulnerabilities.

Threat Modeling: Conducts threat modeling exercises to identify potential security threats and design security controls.

6. Continuous Security Improvement:- Incident Response Readiness: Establishes incident response plans and runbooks for swift response to security incidents.

Feedback Loops: Collects feedback from security incidents and security testing to continuously improve security practices.

7. Security Automation:- Security Orchestration: Orchestrates security tooling and processes to automate security tasks and incident response.Security Policy as Code: Embraces the concept of treating security policies and controls as code, enabling automated security enforcement.

8. Education and Reskilling:- Security Awareness Programs: Provides security training and awareness programs for developers, operators, and other team members to enhance security competence.

Continuous Learning: Encourages ongoing learning and upskilling in security practices to adapt to evolving security challenges.

DevSecOps works by fostering a security-first mindset, continuous collaboration, and automation to ensure that security is an integral part of the software development and deployment lifecycle, promoting the delivery of secure and reliable software applications.

## 5. Tools of Development Security Operations (DevSecOps)

We've compiled an index of some of the best DevSecOps toolkits that businesses can assimilate into their DevOps piping system to ensure safety is controlled constantly throughout the development process. These tools are-

### 1. **Codacy**

Codacy provides development groups with a high-quality mechanization and optimization remedy, allowing them to transfer as far left as possible in the design process, introducing potential problems as soon as possible. Their static code analysis platform helps designers instantly identify and resolve security problems, redundancy, difficulty, classic infringements, and connectivity gaps with each dedicated and drag proposal, explicitly from their Git workspace.

### Features of Codacy

- High-security standards
- Standardization of code
- customized your requirements
- Automation of review process
- Analyses of Code performance
- Analytics in Engineering
- Examination of the Security Code
- Configuration in a cluster / various examples

### 2. **Acunetix**

Acunetix provides an All-in-One internet security scanning to assist programmers in finding vulnerabilities as early as possible.

Acunetix's mission is to assist corporations with a significant online presence in protecting their web assets susceptible to malware by providing exceptional techniques that help developers detect more difficulties and rapidly resolve them. The alternative is simple to implement and allows for centralized control, computerization, and assimilation.

Acunetix is a better solution and one of the industry's most established solutions because it concentrates on internet security and is associated with highly scanning, negligible false positives, easiness of use, distinctive techniques, and SDLC implementation.

### Features of Acunetix

- Threats should be prioritized and controlled.
- Vulnerability analysis
- Management of risks
- Scanning the internet
- Scan the network
- In-depth crawling and assessment
- WordPress monitoring program

- Network safety
- Constant checking
- Users should be assigned aim management.

3. **SonarQube**

SonarSource's open software project also aims to assist programmers via computerization. SonarQube is a code coverage tool that automatically detects errors, security flaws, and code stinks in your source code. It incorporates the native frameworks of design teams to offer continuous code evaluation across several project divisions and pull requests.

SonarQube endorses approximately 27 programming languages and enables reliable code checking, allowing small development teams and businesses relatable to identify problem and defects weaknesses in their applications, preventing unspecified actions from affecting end users.

**Features of SonarQube**

- Static analysis of code
- Bug detection
- Unit tests
- Code coverage
- Improve the workflow with consistent code quality and code safety
- Application Security
- Supports 27 programming languages

4. **Gitlab**

GitLab is a web-based Development Operations model that supports an entire CI/CD toolkit in a separate application. It promotes collaboration among Growth, Safety, and Operations teams, allowing them to achieve faster and recognize security issues without slowing or stopping the CI/CD pipeline by reducing toolchain intricacy.

GitLab, in addition to being titled a CI member, provides the full kit to assist organizations in reducing their Development Operations processing time by merging silos, phases and facilitating a cohesive workload that decreases and simplifies events that were previously separate, such as application security and CI/CD.

### Features of Gitlab

- Audit events
- Compliance management
- Compliance dashboard
- Authentication and authorization
- Multiple LDAP/AD server support
- Kerberos user authentication
- LDAP group sync
- Multiple integrations
- Value stream management

## 5. **Aqua Security**

Aqua security saves the day by securing jars throughout the Development Security Operations pipeline. Aqua's cloud-native threat protection gives users total control over container-based conditions at level, with strict debugging security measures and intrusion detection abilities.

The framework offers customers an API that allows for simple integration and computerization. The Aqua package Security Platform provides entire SDLC (software development life cycle) controls for safeguarding intermodal programs running on-premises or in the data center and Linux or Windows. The framework is compatible with a wide range of improvisation environments.

### Features of Aqua Security

- The most comprehensive CNAPP (cloud-native application protection platform) from design to production.
- Container safety.
- Kubernetes Protection for the Organization as a Whole
- Security without a server.
- Virtual machine Protection.
- CSPM is an abbreviation for Computer Science Project Management.
- Searching for Vulnerabilities.
- Immersive Threat Assessment.

## 6. The Benefits of DevSecOps

By integrating security all throughout the software development process, DevSecOps can provide the following benefits:

1.More secure software development. Teams that incorporate frequent security evaluations into their development workflows reduce the risk of seeing vulnerabilities make it to production.

2.Continuous security testing and monitoring. In the DevSecOps methodology, application security is constantly tested and monitored as the software is created, operated, and updated. This ongoing evaluation allows security issues to be identified and resolved more quickly not only when software is being developed, but also after it is deployed.

3.Improved collaboration and communication. DevSecOps brings together development, security, and operations teams, improving collaboration and communication among these teams and promoting better problem-solving and overall effectiveness.

4. Better visibility and control. Through its built-in security evaluations, DevSecOps gives teams better visibility into an application's security strengths and weaknesses, which in turn helps them better calculate security risks.

5. Compliance. By integrating security into the development process, DevSecOps can help organizations meet regulatory compliance requirements and reduce the risk of non-compliance penalties.

## 7. About Local And International DevSecOps Career Opportunities Career Path.

DevSecOps, the integration of security practices into DevOps processes, has become essential in modern software development to ensure secure and resilient applications. Both local and international job markets offer various career opportunities and career paths in DevSecOps:

- **Local DevSecOps Career Opportunities:**

1. Security Engineer: Specializing in securing software development processes, implementing security controls, and ensuring compliance with security standards.

2. DevOps Engineer with Security Focus: Combining DevOps practices with a strong emphasis on security, automating security tasks, and integrating security tools into CI/CD pipelines.

3. Security Analyst: Analyzing security threats, conducting risk assessments, and implementing security measures in DevOps workflows.

4. Security Architect: Designing secure software architectures, ensuring security best practices are incorporated into the design and development processes.

5. Penetration Tester: Evaluating the security of applications and systems, identifying vulnerabilities, and providing recommendations for improvement.

- International DevSecOps Career Opportunities:

1. Cloud Security Specialist: Focusing on securing cloud environments, such as AWS, Azure, and Google Cloud, and integrating security into cloud-native DevOps practices.

2. Application Security Engineer: Specializing in securing web applications, mobile applications, and APIs through code analysis, vulnerability assessments, and secure coding practices.

3. Compliance and Risk Management Specialist: Ensuring compliance with industry regulations and standards, managing security risks, and implementing security governance frameworks.

4. Security Automation Engineer: Developing security automation solutions, integrating security tools with DevOps processes, and optimizing security operations.

5. Cybersecurity Consultant: Providing strategic security guidance, conducting security assessments, and assisting organizations in strengthening their security posture.

- DevSecOps Career Path:

1. Entry Level: Start as a Junior DevSecOps Engineer, focusing on learning security principles, tools, and working closely with experienced team members.

2. Mid-Level: Progress to a DevSecOps Engineer or Security Analyst role, gaining expertise in integrating security measures into CI/CD pipelines, automating security tasks, and conducting security assessments.

3. Senior Level: Advance to a Senior DevSecOps Engineer or Security Architect role, leading security initiatives, developing security strategies, and guiding teams in implementing secure DevOps practices.

4. Management: Transition into a Security Manager, Director, or Chief Information Security Officer (CISO) role, overseeing security operations, setting security policies, and ensuring organization-wide security compliance.

In both local and international markets, career growth in DevSecOps often involves continuous learning, acquiring certifications (e.g., CISSP, CISM, AWS Security), staying updated with security trends, and gaining hands-on experience in securing software development lifecycles. Embracing a culture of continuous improvement and being adaptable to emerging security challenges are key to building a successful DevSecOps career, both locally and globally.

## Summary

Generally, Development, security, and operations all make up the acronym "devSecOps." It is the inclusion of security right away in the process of developing software or an application. It is a concept where app security is a shared responsibility across all of IT.

- DevSecOps is is management lifecycle approach that combines application planning, delivery and monitoring approaches under a single frame work.Parts of the allure of devsecops is it can speed up many steps in the software development lifecycle and ensure continuous code integrations and updates are handled at the ever-increasing speed of business. An improved return on investment (ROI) for the organization's current security system

- Due to automation, there are fewer opportunities for error or administrative failure situations, two factors that could normally lead to cyberattacks and downtime.
- Automation eliminates the need for cybersecurity architects to set up security consoles, allowing security teams to focus on other urgent challenges while increasing their agility and speed.
- Improved team collaboration and communication
- More adaptability in handling unforeseen changes throughout the development lifecycle
- greater potential for automated builds and quality assurance tests

# References

- www.synopsys.com/glossary/what – is-DevSecOps.html
- www.softwaresolutions.com/resources/benefits-of-devsecops.html
- www.techtarget.com/searchsecurity/tip/benefits of devsecops
- www.practical-devsecops.com/devsecops-life-cycle
- www.fortnet.com/lat/resources/cyberglossary/devsecops