DevSecOps

Institute of Technology

School of Computing

Department of Software Engineering

Software Engineering Tools and Practices

COURSE TITLE: SOFTWARE ENGINEERING TOOLS AND PRACTICE

COURSE CODE: SEng3051

INDIVIDUAL ASSIGNMENT

STUDENT NAME ID

1. ENDALAMAW GETIE ----------145987

SUBMITTED DATE: MAY /21/ 2016

SUBMMITED TO: ISMAEL M.

Table of content

Content page

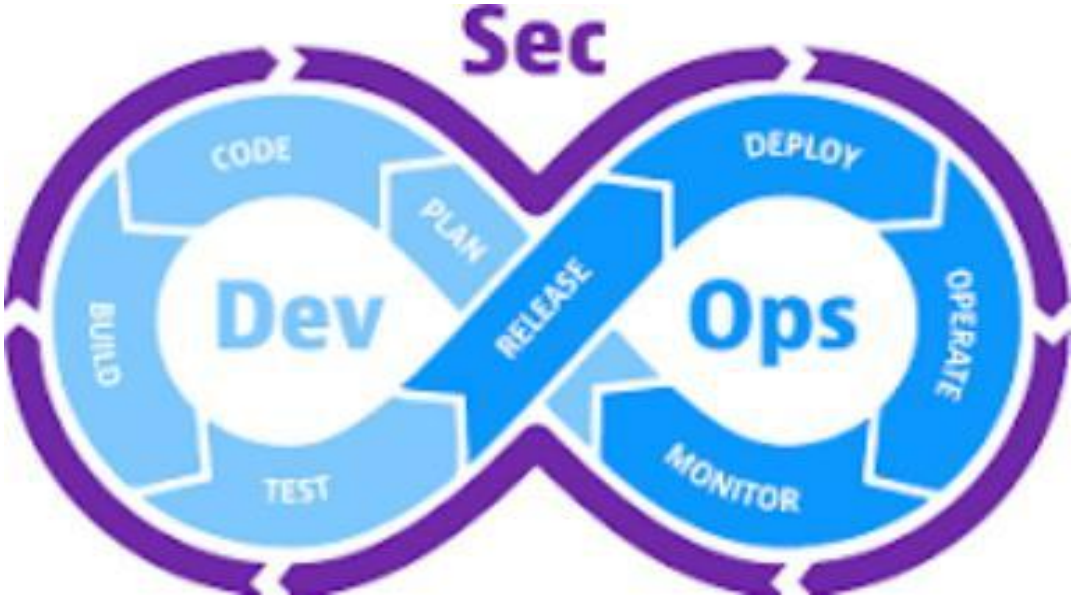
DevSecOps

DevSecOps

Introdauction

DevSecOps, which stands for Development, Security, and Operations, is a method of integrating security principles into the software development lifecycle. It emphasizes the need of addressing security risks early in the development process, rather than as an afterthought. Because of the increased requirement for enterprises to emphasize security in an era of regular cyber-attacks and data breaches, DevSecOps has gained prominence in the software development scene. DevSecOps strives to achieve a balance between agility and security by embedding security into the DevOps approach, allowing teams to release software quickly while ensuring it is strong and resilient against potential security vulnerabilities. DevSecOps arose from the desire to foster a culture of shared responsibility and collaboration among developers.

DevSecOps


Sec
CODE
PLAN
BUILD
TEST
RELEASE
DEPLOY
OPERATE
MONITOR
Dev
Ops

1. What areSoftware Engineering Problems which was cause for initiation of DevSecOps.

The DevOps movement is a method to software development that places an emphasis on teamwork, communication, and automation between IT operations teams (Ops) and software development teams (Dev). It attempts to eliminate the conventional silos that have existed between these two functions and advance a culture of shared accountability, continuous improvement, and quicker software delivery.

Businesses frequently struggle to strike a balance between security and speed when implementing software development techniques. In order to meet customer requests and maintain competitiveness, it is important to deploy software rapidly, yet security precautions must not be compromised. Here are a few obstacles that businesses frequently confront when trying to strike the correct balance.

The need for a security-focused approach within the DevOps framework.

For several reasons, a security-focused strategy within the DevOps framework is essential

1. Growing challenges to security
2. Security that shifts left
3. Quick release iterations
4. Requirements for conformity
5. Maintaining client confidence
6. Cooperation and shared accountability
7. Constant security development

2.What is DevSecOps ?

DevSecOps, which is short for *development*, *security* and *operations*, is an application development practice that automates the integration of security and security practices at every phase of the software development lifecycle, from initial design through integration, testing, delivery and deployment.

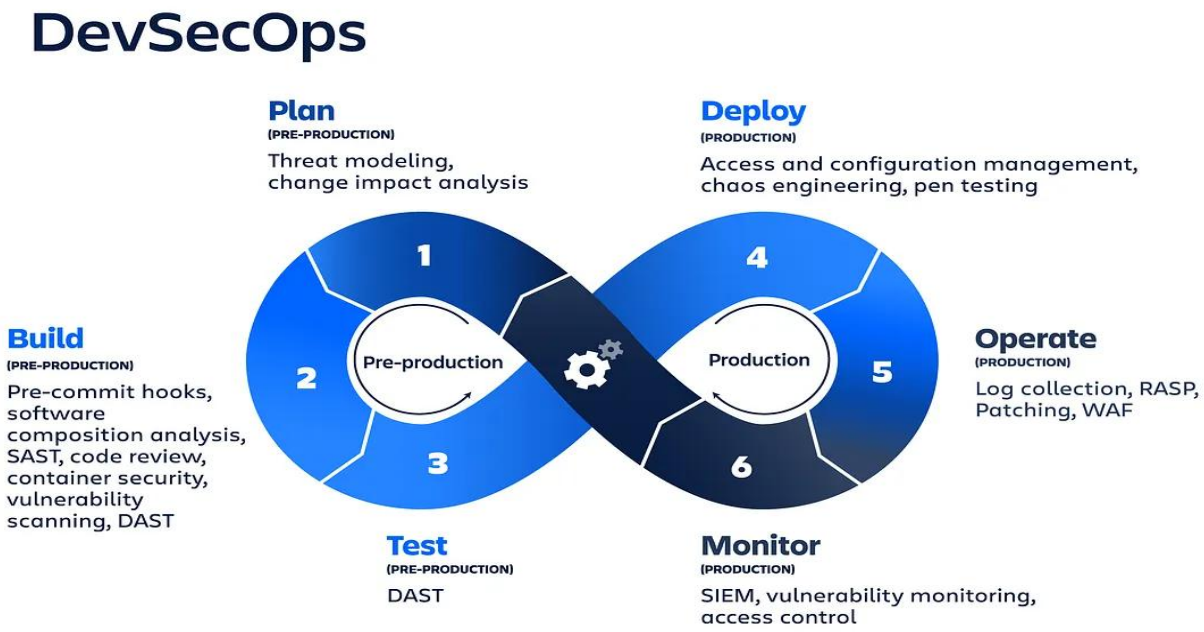
DevSecOps represents a natural and necessary evolution in the way development organizations approach security. In the past, security was 'tacked on' to software at the end of the development cycle, almost as an afterthought. A separate security team applied these security measures and then a separate quality assurance (QA) team tested these measures.

This ability to handle security issues was manageable when software updates were released just once or twice a year. But as software developers adopted Agile and DevOps practices, aiming to reduce software development cycles to weeks or even days, the traditional 'tacked-on' approach to security created an unacceptable bottleneck.

DevSecOps integrates application and infrastructure security seamlessly into Agile and DevOps processes and tools. It addresses security issues as they emerge, when they're easier, faster, and less expensive to fix, and before deployment into production.

Additionally, DevSecOps makes application and infrastructure security a shared responsibility of development, security and IT operations teams, rather than the soleresponsibility of a security silo. It enables “software, safer, sooner”—the DevSecOps motto–by automating the delivery of secure software without slowing the software development cycle.

3.What are Life Cycle of DevSecOps



1. **Plan**

   In the planning phase, the development team collaborates with security and operations teams to define the security requirements and goals for the software project. This involves identifying potential security risks, establishing security controls, and defining compliance requirements.

2. **Develop**

   During the development phase, developers write code while incorporating security best practices. Secure coding guidelines and standards are followed to minimize vulnerabilities. Static code analysis tools can be used to identify potential security issues early on.

3. **Build**

   In the build phase, the code is compiled, packaged, and built into executable software. Continuous integration tools automate the build process and run automated security tests, such as static application security testing (SAST), to identify security weaknesses in the code.

4. **Test**

   the testing phase focuses on comprehensive security testing. It includes various types of security assessments, such as dynamic application security testing (DAST), penetration testing, vulnerability scanning, and security-focused unit testing. These tests help identify and address security vulnerabilities and weaknesses.

5. **Deploy**

   In the deployment phase, the software is deployed to the target environment. Automation tools like continuous deployment (CD) pipelines and infrastructure-as-code (IaC) facilitate consistent and secure deployments. Security checks, such as environment hardening and secure configuration, are performed during this phase.

6. **Operate**

   the operations phase involves monitoring the deployed software for security incidents, performance issues, and compliance. Security monitoring tools and techniques, like log analysis, intrusion detection systems, and security information and event management (SIEM) systems, are employed to detect and respond to security threats.

7. **Maintain**

   the maintenance phase focuses on ongoing security and maintenance tasks. It includes patch management, vulnerability management, and regular security updates. Secure coding practices should continue to be followed for any new features or changes introduced.

8. **Respond**

   the response phase involves promptly addressing security incidents and vulnerabilities that are discovered in production. Incident response plans and procedures are in place to handle security breaches effectively. Lessons learned from security incidents are used to improve security practices for future development cycles.

These phases are iterative and continuous, with feedback loops and automation integrated throughout the lifecycle. By following this DevSecOps lifecycle, organizations

can ensure that security is integrated at every stage of the software development process, leading to more secure and reliable software deployments.

Real-World scenario: DevSecOps lifecycle in action

Imagine a scenario where a tech-savvy software development team embraces the DevSecOps lifecycle for a new application launch. By weaving security controls into the planning phase, rigorously testing for vulnerabilities during development, and orchestrating robust monitoring post-deployment, the team successfully fortifies the application against potential threats, ensuring a robust security posture throughout the lifecycle.

Conclusion: Embracing the essence of DevSecOps lifecycle

In embracing the DevSecOps lifecycle, organizations open the gateway to enhanced security resilience and optimized software development practices. By championing collaboration, automation, and persistent improvement imbibed in the DevSecOps lifecycle, organizations cultivate a security-first mindset that shields their digital assets against evolving threats. Witness the transformative power of DevSecOps as it reshapes security paradigms and propels organizations toward a secure digital future.

4 .Explain well known DevSecOps tools.

The approach to DevSecOps is designed to equip development teams with a complete security framework. This is achieved through continuous collaboration among development, release management (operations), and the security team, emphasizing teamwork throughout each CI/CD Pipeline stage.

The CI/CD Pipeline comprises six phases: Code, Build, Store, Prep, Deploy, and Run. Each phase is outlined below to demonstrate the benefits of incorporating security early in the process:

1. Code

The first step in a DevSecOps-aligned development approach is to code in secure and trustworthy segments. Tools are provided that regularly update these fast building blocks, enhancing the protection of data and applications from the beginning.

2. Build

Transforming code into comprehensive container images, which include a core OS, application dependencies, and runtime services, requires a secure process. This process is managed securely, with runtime dependency scans to improve security, allowing DevSecOps teams to develop with both security and agility.

3. Store

Every pre-packaged technology stack is a potential risk in the current cybersecurity context. Developers can securely obtain specific dependencies and conduct vulnerability scans on container images to mitigate these risks.

4. Prep

Before deployment, it's essential to ensure applications comply with security policies. This involves validating configurations against the organization's security policies before moving to the following stages of the development cycle. These configurations, which determine how the workload should run, not only identify potential vulnerabilities but also set the stage for successful deployment in subsequent CI/CD pipeline phases.

5. Deploy

Scans performed in earlier stages give a comprehensive view of the application's security status. At this stage, any identified vulnerabilities or misconfigurations in the development process are presented, allowing organizations to address issues and establish more robust security standards, thus enhancing their security posture.

6. Run

As deployments are executed, teams can utilize active deployment analytics, monitoring, and automation to ensure continuous compliance and address vulnerabilities that arise after deployment.

5.Explain well known DevSecOps tools.

There are several well-known DevSecOps tools that organizations can use to integrate security practices into their DevOps workflows. Here are some popular DevSecOps tools:

1. **Static Application Security Testing (SAST) Tools:**
2. **Checkmarx:** Checkmarx is a leading SAST tool that helps developers identify and fix security vulnerabilities in their code.
3. **Veracode:** Veracode offers a cloud-based SAST solution for secure software development.

4. **Dynamic Application Security Testing (DAST) Tools:**
5. **OWASP ZAP (Zed Attack Proxy):** ZAP is an open-source DAST tool that helps developers find security vulnerabilities in web applications.
6. **Burp Suite:** Burp Suite is a popular DAST tool for web application security testing.

7. **Container Security Tools:**
8. ***Docker Bench for Security:*** Docker Bench for Security is a tool that checks for common best practices in Docker deployments.
9. ***Clair:*** Clair is an open-source container vulnerability scanning tool that helps identify security issues in container images.

10. **Infrastructure as Code (IaC) Security Tools:**
11. ***Terraform:*** Terraform is a popular IaC tool that allows infrastructure to be defined as code. Security best practices can be integrated into Terraform configurations.
12. ***AWS Config:*** AWS Config provides continuous monitoring and assessment of AWS resource configurations for security compliance.

13. **Security Orchestration, Automation, and Response (SOAR) Platforms:**

14. ***Demisto:*** Demisto is a SOAR platform that helps automate incident response processes and integrates with various security tools for orchestration.

15. Security Information and Event Management (SIEM) Tools:

16. ***Splunk:*** Splunk is a widely-used SIEM tool that collects and analyzes security event data to provide insights into security incidents.

17. Open Source Security Tools:

18. ***OpenSCAP:*** OpenSCAP is an open-source Security Content Automation Protocol (SCAP) toolkit for compliance checking and vulnerability scanning.
19. ***OSSEC:*** OSSEC is an open-source host-based intrusion detection system that provides log analysis, file integrity checking, and rootkit detection.

These are just a few examples of the many DevSecOps tools available to help organizations enhance their security practices within the DevOps workflow. It's important to evaluate the specific needs and requirements of your organization to choose the most suitable tools for your DevSecOps initiatives.

6. What are the benefits of DevSecOps?

The two main benefits of DevSecOps are speed and security. Therefore, development teams deliver better, more-secure code faster and cheaper.
The purpose and intent of DevSecOps is to build on the mindset that everyone is responsible for security with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required

1. Rapid, cost-effective software delivery

When software is developed in a non-DevSecOps environment, security problems can lead to huge time delays. Fixing the code and security issues can be time-consuming and expensive. The rapid, secure delivery of DevSecOps saves time and reduces costs by minimizing the need to repeat a process to address security issues after the fact. This process becomes more efficient and cost-effective since integrated security cuts out duplicative reviews and unnecessary rebuilds, resulting in more secure code.

2 .Improved, proactive security

DevSecOps introduces cybersecurity processes from the beginning of the development cycle. Throughout the development cycle, the code is reviewed, audited, scanned and tested for security issues. These issues are addressed as soon as they are identified. Security problems are fixed before additional dependencies are introduced. Security issues become less expensive to fix when protective technology is identified and implemented early in the cycle.
Additionally, better collaboration between development, security and operations teams improves an organization’s response to incidences and problems when they occur. DevSecOps practices reduce the time to patch vulnerabilities and free up security teams to focus on higher value work. These practices also ensure and

simplify compliance, saving application development projects from having to be retrofitted for security.

3. Accelerated security vulnerability patching

A key benefit of DevSecOps is how quickly it manages newly identified security vulnerabilities. As DevSecOps integrates vulnerability scanning and patching into the release cycle, the ability to identify and patch common vulnerabilities and exposures (CVE) is diminished. This capability limits the window that a threat actor has to take advantage of vulnerabilities in public-facing production systems.

4. Automation compatible with modern development

Cybersecurity testing can be integrated into an automated test suite for operations teams if an organization uses a continuous integration/continuous delivery pipeline to ship their software.

Automation of security checks depends strongly on the project and organizational goals. Automated testing can ensure that incorporated software dependencies are at appropriate patch levels, and confirm that software passes security unit testing. Plus, it can test and secure code with static and dynamic analysis before the final update is promoted to production.

5. A repeatable and adaptive process

As organizations mature, their security postures mature. DevSecOps lends itself to repeatable and adaptive processes. DevSecOps ensures that security is applied consistently across the environment, as the environment changes and adapts to new requirements. A mature implementation of DevSecOps will have a solid automation, configuration management, orchestration, containers, immutable infrastructure and even server less compute environments

6. Quick Resolution of Security Flaws

A key benefit of DevSecOps is its quick response to security weaknesses. Dealing with common vulnerabilities during the development phase reduces the risks linked to flaws in development frameworks.

7. Automated Security Monitoring and Testing

DevSecOps enhances security monitoring and testing through automation. This method uses automated testing to check and compare actual results with expected ones, either through automated test scripts or testing tools. It also ensures thorough

code testing and validation with static and dynamic assessments before integration into the development cycle.

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

In the recent years DevSecOPs is highly needed field in the software development process, due to this need demand for professionals who can integrate security practices into the DevOps is highly increasing

1. Local Opportunities: In many regions, including the US, Europe, Asia, and Australia,
there is a growing demand for DevSecOps professionals. Companies of all sizes, from startups to large enterprises, are looking for individuals who can help them secure their software development processes. Local job boards, recruitment agencies, and networking events can be good places to find DevSecOps job opportunities in your area.

2. International Opportunities: DevSecOps skills are in demand globally, and many
companies are open to hiring remote workers or sponsoring work visas for talented professionals. International job boards, networking platforms like LinkedIn, and specialized websites for tech jobs can help you explore opportunities in other countries.

3. Career Path: A typical career path in DevSecOps may start with roles such as Security Analyst, DevOps Engineer, or Software Developer with a focus on security. As you gain experience and expertise in integrating security practices into the development lifecycle, you may progress to roles like DevSecOps Engineer, Security DevSecOps 8Software Tools and Practice
Engineer, Security Architect, or even Chief Information Security Officer (CISO). Continuous learning, obtaining relevant certifications (such as Certified DevSecOps Professional), and staying updated on industry trends are key to advancing your career in DevSecOps.

Reference

https://dodcio.defense.gov/Portals/0/Documents/Library/DevSecOpsTools-ActivitiesGuidebook.pdf
https://www.everand.com/book/488471049/DevSecOps-A-Complete-Guide-2021-Edition
https://devsecopsguides.github.io/
https://www.scribd.com/document/583671435/DoD-Enterprise-DevSecOps-2-0-Fundamentals