Institute of Technology

School of Computing

Department of Software Engineering

Software Engineering Tools and Practices

Individual Assignment DevSecOps

Name id

Elias Sirak----------145961

Submitted to: Instructor Esmael (MSc)

Table of Contents

1. What are Software engineering problems which was cause for initiation of DevSecOps ?

Software engineering problems that caused the initiation of DevSecOps:

• Lack of collaboration between development and security teams: This led to security vulnerabilities being introduced into software products due to a lack of communication and coordination between the two teams.

• Manual and time-consuming security testing: Traditional security testing methods were often manual and time-consuming, which slowed down the software development process and made it difficult to keep up with the pace of development.

• Lack of automation in security processes: Security processes were often not automated, which made it difficult to ensure consistency and quality in security testing and remediation.

• Lack of visibility into security risks: Development teams often lacked visibility into the security risks associated with their code, which made it difficult to prioritize and address security vulnerabilities.

• Lack of accountability for security: There was often a lack of clear accountability for security within software development teams, which made it difficult to identify and address security issues.

DevSecOps was initiated to address these problems by bringing together development and security teams into a collaborative environment, automating security processes, and providing development teams with visibility into security risks. By doing this, DevSecOps helps to improve the security of software products and reduce the time and effort required to develop and deploy secure software

2. What is DevSecOps?

DevSecOps Definition

DevSecOps stands for development, security, and operations. At its core, it is a concept where app security is a shared responsibility across all of IT. The DevSecOps definition revolves around

automatically making security a top priority as part of any software development lifecycle, with that continuing after development ends.

DevSecOps process is a method for handling IT security with the mindset that "everyone is accountable for security." It combines injecting security into a company's DevOps pipeline. The aim is to involve security in all software development life cycle (SDLC) stages. DevSecOps framework indicates you shouldn't save security for the last stage of the SDLC, contrary to its predecessor development methods.

If your business already uses DevOps, consider upgrading it to DevSecOps integration. DevSecOps phases are primarily built on the DevOps services, which will guide your case for switching. By doing this, you'll be able to assemble talented specialists from several technical disciplines to improve your security procedures as they are now.

DevSecOps is a way of thinking or a culture that IT operations and developers' teams follow when creating and deploying software applications. Agile application development incorporates security audits and penetration testing that are both active and automated.

To implement the DevSecOps process, you need to:

- Reducing vulnerabilities in software programming incorporates the concept of security from the beginning of the SDLC.
- Please make sure everyone, including IT operations teams and developers, shares responsibility for adhering to security procedures in their tasks.
- Ensure DevOps workflow begins with the involvement of security controls, processes, and tools. This will allow for automatic security checks throughout the software delivery process.

DevOps managed services have always been about integrating security into the development and release process, quality assurance (QA), database management, and everyone else. DevSecOps process, on the other hand, is an extension of that process where security is always the crucial component of the procedure.

DevSecOps is a term that is becoming increasingly popular in the world of software development, and it is quickly becoming the preferred methodology for many organizations.

DevSecOps is an approach to software development that combines the principles of DevOps with security and compliance. It is a way of automating the process of development, testing, deployment and maintenance of applications, while ensuring that security and compliance requirements are met. In this blog, we will discuss the definition of DevSecOps, best practices for implementing it and the tools available for it.

We've built a platform to automate incident response and forensics in AWS, Azure and GCP — you can grab a free trial here. You can also download a free playbook we've written on how to respond to security incidents in the cloud.

First, let's define DevSecOps. DevSecOps is a combination of DevOps and security practices that focus on the security of applications and infrastructure from the initial stages of development through deployment and maintenance. It is an approach to software development that automates the process of development, testing, deployment and maintenance of applications, while ensuring that security and compliance requirements are met. DevSecOps seeks to ensure that security is built in at every stage of the software development lifecycle, and that security is automated and integrated into the process.

When it comes to best practices for implementing DevSecOps, it is important to ensure that security is baked into the entire process. This means that security must be considered from the design phase all the way through to the deployment and maintenance of the application. Additionally, organizations should ensure that they have the right tools and processes in place to ensure that the security of their applications is maintained.

Examples of DevSecOps best practices include:

• Automating security processes and integrating them into the DevOps

pipeline

• Implementing security scanning tools to detect vulnerabilities

• Using container security solutions to secure containers

• Implementing Continuous Integration/Continuous Delivery (CI/CD)

• Setting up security policies and enforcing them throughout the organization

• Using software composition analysis to identify open source components

• Adopting secure coding practices

What are the principles of DevSecOps?

1. Automate All the Things: Automation is the cornerstone of successful DevSecOps. Automating routine tasks and processes helps to speed up the delivery process while enabling teams to focus on the more complex aspects of the development lifecycle.

2. Shift Security Left: Shifting security left is the process of integrating security practices earlier in the development process. This helps to reduce the risks associated with the production environment.

3. Continuous Integration and Continuous Delivery: Continuous integration (CI) and continuous delivery (CD) are important for DevSecOps. CI and CD ensure that code is tested, reviewed, and deployed quickly and consistently.

4. Collaboration and Communication: Collaboration and communication are key to successful DevSecOps. It's important for teams to be able to share ideas, discuss challenges, and work together in order to ensure that security is properly implemented.

5. Measure and Monitor: Measuring and monitoring are essential for DevSecOps. Teams should measure and monitor their systems, processes, and performance in order to identify potential security issues and address them quickly.

What is the culture of DevSecOps?

The culture of DevSecOps is one that emphasizes collaboration and integration among development, security, and operations teams. It is rooted in the idea that security should not be an afterthought, and instead should be considered from the earliest stages of a project. It is a culture of shared responsibility, where all teams work together to ensure that security is properly addressed across all stages of the software development life cycle. At the core of DevSecOps is a focus on automation and continuous integration, which will ensure that security measures are implemented quickly and consistently.

What problems does DevSecOps solve?

1. Increased Security: DevSecOps provides organizations with a more secure software development process by incorporating security into every stage of the development process. This
2. ensures that any security issues are identified and addressed much earlier in the development cycle, reducing the potential for costly security incidents.

3. Improved Efficiency: DevSecOps enables organizations to streamline their development processes by automating tasks and eliminating manual steps, resulting in faster and more reliable delivery of software.

4. Improved Collaboration: DevSecOps encourages collaboration between development and security teams, allowing developers and security professionals to work together to identify and address security issues. This leads to improved communication and collaboration, resulting in a more secure development process.

5. Increased Visibility: DevSecOps makes it easier to track and monitor security issues throughout the software development process. This provides organizations with greater visibility into the security of their applications and helps them identify potential risks much earlier in the development cycle.

Finally, there are a number of DevSecOps tools available to organizations looking to implement DevSecOps. These tools can help organizations automate the process of development, testing, deployment and maintenance of applications, while ensuring that security and compliance requirements are met**.**

Examples of DevSecOps tools include:

• 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 enforced

3. Briefly explain DevSecOps lifecycle?

In the dynamic realm of cybersecurity, the DevSecOps lifecycle stands tall as a beacon of security integration in software development. Envision yourself as a cybersecurity aficionado, delving into the intricacies of this holistic approach that intertwines development, security, and operations seamlessly. Let's embark on an enlightening journey through the phases of the DevSecOps lifecycle to unravel its significance in fortifying digital fortresses.

- Understanding the Imperative of DevSecOps
- Key Phases in the DevSecOps Lifecycle

  1. Planning and Security Integration

  2. Continuous Integration and Security Testing

  3. Deployment and Configuration Security

  4. Monitoring and Incident Response
- Real-World Scenario: DevSecOps Lifecycle in Action
- Conclusion: Embracing the Essence of DevSecOps Lifecycle

Understanding the Imperative of DevSecOps

The DevSecOps lifecycle serves as the backbone of security enhancement within the software development continuum. It embodies a structured flow of stages that enables organizations to embed security practices from inception to deployment, fostering a security-centric culture across teams.

- **Security by Design**: Embedding security principles at the core of software development processes.

- **Collaborative Synergy**: Nurturing collaboration among diverse teams for enhanced security posture.

- **Risk Mitigation Strategies**: Proactively identifying and mitigating security risks during the lifecycle.

- **Iterative Security Enhancements**: Instilling a cycle of continuous improvement for bolstering security resilience.

Key Phases in the DevSecOps Lifecycle

Embark on a journey through the pivotal phases that define the DevSecOps lifecycle:

1. Planning and Security Integration

- **Define Security Requirements:** Lay down the foundational security requirements and objectives.
-
- **Integrate Security Controls:** Incorporate security controls early in the planning stage to align with overarching security goals.
-

2. Continuous Integration and Security Testing

- **Automated Security Testing:** Integrate robust security testing tools into the development pipeline for automated assessments.

- **Vulnerability Identification:** Conduct frequent security assessments to pinpoint vulnerabilities and address them promptly.

Also Read, DevSecOps Automation Guide

3. Deployment and Configuration Security

- **Secure Deployment Practices:** Implement secure deployment protocols and robust configuration management practices.

- **Infrastructure as Code (IaC):** Employ IaC principles for consistent, secure deployments across environments.

Also Read, Best DevSecOps Tools

4. Monitoring and Incident Response

- **Real-time Monitoring:** Establish vigilant monitoring mechanisms to detect security events and anomalies promptly.

- **Incident Response Protocols:** Define clear incident response procedures and conduct post-incident analyses for continual enhancement.

Also read, Why DevSecOps is a Good Career Option?

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. How dose DevSecOps works?

DevSecOps works by integrating security into every phase of the software development lifecycle (SDLC), from planning and design to deployment and maintenance. This is achieved through a combination of cultural changes, process changes, and technology.

Cultural changes:

• Collaboration: Development, security, and operations teams work together throughout the SDLC.

• Communication: Teams communicate openly and frequently about security risks and mitigation strategies.

• Trust: Teams trust each other to do their part to secure the software product.

Process changes:

• Security requirements are defined and integrated into the software design.

• Security testing is performed throughout the SDLC.

• Code is reviewed for security vulnerabilities.

• Security configurations are applied to the deployment environment.

• Security monitoring is implemented to detect and respond to security threats.

Technology:

• Automated security testing tools: These tools help to identify security vulnerabilities in software products.

• Code review tools: These tools help to identify potential security vulnerabilities in code.

• Threat modeling tools: These tools help to identify and assess potential threats to software products.

• Security monitoring tools: These tools help to detect and respond to security vulnerabilities and threats.

By combining these cultural, process, and technology changes, DevSecOps helps organizations to improve the security of their software products while also increasing the speed and efficiency of the software development process.

Here is a simplified example of how DevSecOps works in practice:

1. A development team is working on a new software product.

2. The security team reviews the software design and identifies potential security risks.

3. The development team implements security controls to mitigate the risks identified by the security team.

4. Automated security testing tools are used to identify security vulnerabilities in the code.

5. Code is reviewed for security vulnerabilities by both the development team and the security team.

6. The software is deployed in a secure environment and security configurations are applied.

7. The software is continuously monitored for security vulnerabilities and threats.

8. In the event of a security breach, the incident response team is notified and takes steps to mitigate the breach and prevent further damage.

By following the DevSecOps lifecycle and using the appropriate cultural, process, and technology changes, organizations can improve the security of their software products and reduce the risk of security breaches.

5. Exline well known DevSecOps tools.

Well-known DevSecOps tools:

Automated security testing tools:

• OWASP ZAP: A free and open-source web application security scanner.

• Nessus: A commercial vulnerability scanner.

• Burp Suite: A commercial web application security testing suite.

• Fortify: A commercial static application security testing (SAST) tool.

• Checkmarx: A commercial SAST tool.

Code review tools:

• SonarQube: A free and open-source code quality and security analysis tool.

• CodeClimate: A commercial code review tool.

• Review Board: A free and open-source code review tool.

• Veracode: A commercial SAST tool that includes code review capabilities.

• Coverity: A commercial SAST tool that includes code review capabilities.

Threat modeling tools:

• Microsoft Threat Modeling Tool: A free and open-source threat modeling tool.

• OWASP Threat Dragon: A free and open-source threat modeling tool.

• ThreatModeler: A commercial threat modeling tool.

• iThreat: A commercial threat modeling tool.

• SecurITree: A commercial threat modeling tool.

Security monitoring tools:

• Splunk: A commercial security information and event management (SIEM) tool.

• Elasticsearch: A free and open-source SIEM tool.

• Logstash: A free and open-source log aggregation tool.

• Kibana: A free and open-source data visualization tool that can be used with Elasticsearch and Logstash.

• Grafana: A free and open-source data visualization tool that can be used with a variety of data sources, including security monitoring tools.

Other DevSecOps tools:

• Jenkins: A free and open-source continuous integration (CI) tool.

• Kubernetes: A free and open-source container orchestration tool.

• Docker: A free and open-source containerization platform.

• Ansible: A free and open-source IT automation tool.

• Terraform: A commercial infrastructure-as-code (IaC) tool.

These are just a few of the many DevSecOps tools available. The specific tools that an organization chooses will depend on its specific needs and requirements.

6. What are the benefits of DevSecOps?

Ultimately, DevSecOps is benefits because it places security in the SDLC earlier and on purpose. When development organizations code with security in mind from the outset, it's easier and less costly to catch and fix vulnerabilities before they go too far into production or after release. Organizations in a variety of industries can implement DevSecOps to break down silos between development, security, and operations so they can release more secure software faster.

- **Automotive:** DevSecOps reduces lengthy cycle times while still ensuring that software compliance standards such as MISRA and AUTOSAR are met

- **Healthcare:** DevSecOps enables digital transformation efforts while maintaining the privacy and security of sensitive patient data per regulations such as HIPAA

- **Financial, retail, and ecommerce:** DevSecOps helps ensure that the OWASP Top 10 web application security risks are addressed and maintains PCI DSS data privacy and security compliance for transactions among consumers, retailers, financial services, and so on

- **Embedded, networked, dedicated, consumer, and IoT devices:** DevSecOps enables developers to write secure code that minimizes the occurrence of the CWE Top 25 most dangerous software errors

DevOps has transformed the field of the software industry, and integrating security into this paradigm, known as DevSecOps, is elevating software development practices. Embracing DevSecOps offers various advantages, such as:

1) Rapidly Addressing Security Vulnerabilities

A significant advantage of DevSecOps lies in its prompt handling of newly discovered vulnerabilities. By seamlessly incorporating vulnerability scanning and patching into the release cycle, DevSecOps significantly improves the capability to detect and address common vulnerabilities and exposures swiftly. This, in turn, reduces the timeframe during which threat actors can exploit vulnerabilities in public-facing production systems.

2) Shared Responsibility Across Teams

DevSecOps aligns development and security teams from the outset of the development cycle, fostering a collaborative cross-team approach. Rather than adhering to a siloed and disjointed

operational approach that stifles innovation and triggers conflicts, DevSecOps encourages teams to synchronize early, promoting effective cross-team collaboration.

3) Improved Application Security

DevSecOps adopts a proactive strategy for addressing security vulnerabilities in the early stages of developing the DevSecOps lifecycle. Development teams in the DevSecOps framework leverage automated

security tools to test code and conduct security audits seamlessly, avoiding any hindrance to the development process or the software delivery pipeline.

Throughout different phases of the development process, the DevSecOps lifecycle reviews, audits, tests, scans, and debugging to ensure that the application successfully clears crucial security checkpoints. In the event of security vulnerabilities emerging, collaboration between application security and development teams ensues, involving a joint effort in conducting security analysis and devising solutions at the code level.

4) Swift and Economical Software Delivery

DevSecOps' quick and secure delivery approach not only saves time but also reduces costs by minimizing the necessity of revisiting processes to address security issues after the fact. Integrating security in this process is efficient and cost-effective, eliminating redundant tasks and unnecessary reworks and reviews, thereby enhancing overall security measures.

5) Suitable for Automation in a Contemporary Development Team

DevSecOps framework empowers software teams to integrate security and observability seamlessly into DevSecOps automation, accelerating the SDLC and ensuring a more secure software release process.

Automated testing plays a crucial role in verifying that integrated software dependencies, such as libraries, frameworks, and application containers, meet the required security standards, especially

in the case of unknown vulnerabilities. DevSecOps automation testing confirms that the software has successfully undergone security unit testing across all levels. This comprehensive approach includes testing and securing code through static, dynamic, and dependency analyses before the final

software is deployed to production. Automated tools can scan containers and scrutinize their dependencies to identify and report vulnerable components.

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

Introduction to DevSecOps Career Path

DevOps is a practical approach for delivering reliable software quickly, but security is often left as an afterthought. DevSecOps integrates security as an essential component of the SDLC, distributing security responsibilities amongst team members and encouraging a “Security as Code” culture.

The DevSecOps Career Path

DevSecOps is a professional career that starts from software development. Most of the engineers in the DevSecOps field started in software development (or) system administration, then later these professionals would be transitioned into DevSecOps. Another related certification is Certified DevSecOps Professional (CDP) & Certified DevSecOps Expert (CDE). If you’re someone who is looking for a high leadership level certification in DevSecOps then you can take Certified DevSecOps Leader certification (CDL).

Here is a brief overview of the DevSecOps Career Path

What is a DevSecOps Engineer?

A DevSecOps Engineer is a security professional responsible for ensuring comprehensive and effective integration by the security team within the Software Development Life Cycle (SDLC). The task is, therefore, to find the possible vulnerabilities that security might have: either a technique or a strategy to mitigate the possible risks, in order for the risks of those vulnerabilities not to materialize.

The DevSecOps engineers are among those who look for, apply security controls to, and assure that the work done conforms to the appropriate high security standards and regulations. He is, in other words, the most important professional who makes sure the security is upfront of the SDLC.

DevSecOps Engineer Skills

To become a pro-DevSecOps engineer in 2024, aspiring individuals must have different technical and soft skills in

combination. Right below, we have listed out the best DevSecOps Engineer skills that are required:

- Strong understanding of security concepts, including threat

  modeling, risk assessment, and vulnerability management.

- Knowledge of the SDLC and experience integrating security best practices at every process stage.

- Familiarity with automation tools and scripting languages like Python and PowerShell.

- Understanding cloud security principles, including secure architecture design and configuration management.

- Knowledge of container security principles, such as Docker and Kubernetes.

- Experience with DevOps practices, such as continuous integration and delivery (CI/CD) and infrastructure as code (IaC).

- Experience with various compliance frameworks and regulations: PCI-DSS, HIPAA, and GDPR.

- Good analytical problem-solving skills to scrutinize and solve very intricate security problems with effective solutions.

- Ability to work cohesively with cross-functional teams and possess good communication skills.

- Passionate about continued learning and being aware of current security trends and technologies.

DevSecOps Engineer Roles and Responsibilities

DevSecOps engineer roles and responsibilities are various tasks, including:

- Integrating security features in the software development life cycle.

- Identification and probable security risks, with their mitigating strategies.

- Implementation of security controls.

- Monitoring of the threat to security.

- Ensuring regulatory compliances for standards of security.

- Proficient in uniting cross-functional teams and communicating clearly, while fervently pursuing knowledge of the latest trends and technologies in security.

DevSecOps Engineer Requirements

DevSecOps engineer requirements are several, and some of them are as follows:

- Early detection of security vulnerabilities

- Faster deployment of secure software

- Enhanced collaboration among development, security, security, and operations teams.

- By following better compliance with security standards and regulations

- Greater visibility into security risks and threats

How to Become a DevSecOps Engineer?

To be a DevSecOps Engineer, one should have a strong basis in software development and principles of security. For example, some would be

computer science, information technology, or any other degree-related stream from a relevant field. The same would stand you in good stead, for example, certifications like Certified DevSecOps Professional (CDP) in the area of shows off your security knowledge.

Also read, Best DevSecOps Books

Learning Resources for DevSecOps

Several resources are available for anyone interested in learning more about DevSecOps.The right DevSecOps Career Path to Becoming a Skilled DevSecOps engineer includes the aspiring individual equipping himself with essential tools.

Here are the resources you can use to pave your way to becoming a DevSecOps engineer, Namely:

- Git (Version Control System)
- CI/CD (Continuous Integration and Delivery)
- Artifact management
- Infrastructure as Code(Configuration management tools)
- Cloud Platforms (AWS, GCP, or Azure)

Do not feel overwhelmed! Initially, you only have to build a basic understanding of these tools.

Here is the link to the List of Videos, Tutorials, Blogs, Hands-on labs, or Online playgrounds you can use to pave your way to becoming a DevSecOps Engineer.

DevSecOps Tools and Technologies

DevSecOps engineers will be armed with a large variety of tools and technologies that they will apply to their work. They typically work within an environment that is supported by automated testing tools in the case of potential security vulnerability areas. Below is a list of Top 6 best tools and technologies used by DevSecOps professionals are:

- Jenkins
- GitLab
- Docker
- Kubernetes
- Ansible
- Terraform

Also Read, Best DevSecOps Tools in 2023

What Does a DevSecOps Engineer Do?

- DevSecOps engineers are required to be capable of efficiently implementing a range of DevSecOps best practices, including:

- Build in security early and often within the SDLC, so each of the phases identifies and mitigates the risks in the process.

- Cultivate a security culture within the organization: every stakeholder should know their responsibilities.

- The idea is that you should automate everything in the security testing and deployment process that you can possibly automate, as more likely to be driven by human error.

- Take a security risk-based approach, focusing on all important but most critical assets and vulnerabilities.

- To leverage IaC (infrastructure as a code) in a more consistent and efficient way to put up secure environments.

- Security is to be regularly assessed, and penetration testing should help in identifying any exposure for the improvement of security posture.

- Help in sharing knowledge and best practices between the security, development, and operation teams to achieve true collaboration. Monitor the environment from any security threat and respond promptly to incidents or breaches.

- Utilize a security-centric DevOps toolchain to integrate security testing, deployment, and processes smoothly.

- Integration of security into the SDLC will ensure developed software complies with some security standards and regulations, for example, PCI-DSS, HIPAA, GDPR, etc.

Also Read, **Must Know** DevSecOps Engineer Interview Questions

Challenges Faced by DevSecOps Engineers

DevSecOps engineers face several challenges, including

- Keep up with new security threats and vulnerabilities.

- Balancing security against development pace while ensuring compliance with the standard and regulation.

- Work harmoniously with developers and other stakeholders to

manage complexity in cloud environments**.**

DevSecOps Engineer- Frequently Asked Questions

What is the difference between a DevOps engineer and a DevSecOps engineer?

A DevOps engineer focuses on integrating development, operations, and quality assurance processes, while a DevSecOps engineer incorporates security practices into the DevOps workflow.

What is the difference between a cybersecurity engineer and a DevSecOps engineer?

A cybersecurity engineer protects systems and responds to threats, whereas a DevSecOps engineer integrates security into the software development process, ensuring secure application delivery.

Is DevSecOps a good career, and is it in demand?

Yes, DevSecOps does make a promising career. But, hand in hand, with the increasing demand for secure software, the number of jobs is also increasing, which focuses on professionals who can deliver a balance between development, operations, and security.

Also Read, Why DevSecOps Engineer is a Promising Career.

What is the goal of a DevSecOps engineer?

The aim of a DevSecOps engineer is to inculcate security in the process: safe coding practices and the most important cultural change to the culture of security awareness and working with collaboration**.**

What is a Certified DevSecOps Engineer?

Certified DevSecOps Engineer is an experienced person who will be responsible for integrating the industry best security practice into DevOps pipelines. They are able to perform secure coding practices, security testing, and risk assessment to enable the betterment of enterprise security posture.

Conclusion

DevSecOps is one of the fundamental practices for organizations running software applications. However, being a good DevSecOps engineer would mean understanding the basic principles of

software development and security. As you knew, in the DevSecOps world, we need to keep continuous learning about emerging technologies and getting to know the latest security threats

in the current market.

The DevSecOps field is projected to experience rapid growth, with revenues exceeding $17.24 Billion by 2028!

Practical DevSecOps offers an excellent Certified DevSecOps Professional (CDP) course with hands-on training through browser-based labs, 24/7 instructor support, and the best learning resources to up skill in DevSecOps.

Start your journey towards becoming a skilled DevSecOps engineer *with Practical DevSecOps!*