

Important DevSecOps Interview Questions






How can security be prioritized within the DevOps workflow?

Answer: "Prioritize security by integrating it into every stage of the DevOps workflow. This includes incorporating security requirements into user stories, performing security testing during development, and conducting regular security audits to ensure our systems are secure. It also requires collaboration and communication across teams, automation tools wherever possible, and an ongoing focus on measuring and improving security metrics."






Why is it important to have security tool output in a machine-readable format?

Answer: "Having security tool output in a machine-readable format is essential as it enables automation, streamlines processes, and allows for greater consistency and standardization across different systems."






What are the main challenges faced while implementing SCA and how can they be addressed in a DevSecOps environment?

Answer: "Challenges in implementing SCA in DevSecOps include:

- Developers' lack of awareness of SCA's importance and risks.
- Outdated dependencies in legacy code.
- Difficulty in identifying vulnerabilities in transitive dependencies."





"These challenges can be addressed through training, awareness, tools to manage dependencies, and improved vulnerability identification."





What are some of the benefits of SAST in the DevSecOps Process?

Answer: "SAST plays an important role in the DevSecOps process. By performing SAST early in the development process, potential vulnerabilities can be identified and addressed before the code is compiled or executed. This can save time and resources, as vulnerabilities discovered later in the development process may require significant rework or the code to be rewritten from scratch."






"In addition to this, SAST is easy to get started and does both data flow and control flow analysis."



How do you approach threat modeling?

Answer: "To approach threat modeling, start by identifying the protected assets, such as data or functionality, and potential attackers who might target them. Next, identify potential threats and attack vectors, such as injection or denial-of-service attacks. Analyze the risks associated with each threat and prioritize them based on their likelihood and impact."





"Once risks have been prioritized, identify and implement controls to mitigate risks. Controls can range from architectural changes to code-level fixes to security awareness training for developers."






How does compliance of code help in the DevSecOps process?

Answer: "Compliance as Code is a methodology that utilizes code and automation to enforce compliance with security policies and industry regulations. This approach can help improve the security of the DevSecOps process in various ways, including automation, integration, and scalability."





"Overall, compliance as code helps implement a proactive and continuous security approach in DevSecOps, allowing for standardization in security practices, improving security through automation, managing costs, and maintaining security compliance across diverse infrastructure and platforms."






How would you assess the effectiveness of DevSecOps implementation across the organization?

Answer: "Assessing the effectiveness of DevSecOps implementation across an organization can be challenging, but there are several key factors to consider like security metrics, code quality, collaboration and communication, automation, and time to market."






"Overall, assessing the effectiveness of DevSecOps implementation is an ongoing process that requires tracking multiple factors and metrics over time. By monitoring and measuring these factors, organizations can identify areas for improvement and continue to refine their DevSecOps implementation to meet their specific security goals and objectives better."



What are some weaknesses of DAST compared to other security methods

Answer: “DAST is performed later in the development process, meaning vulnerabilities may not be identified until after the code has been deployed to a test or production environment. This can increase the costs and time required to remediate vulnerabilities, as well as negatively impact the overall security of the application.





Dynamic analysis is prone to lack of coverage because of its inability to crawl heavy Javascript frameworks. This can result in vulnerabilities going undetected, as attackers may exploit untested areas of the application.

DAST can generate false positives and negatives, leading to wasted resources and missed vulnerabilities that attackers could exploit. Unlike SAST, DAST cannot analyze code directly, which makes identifying the root cause of vulnerabilities and addressing them more challenging.



Differentiate between DevOps and DevSecOps?

Answer: "DevOps and DevSecOps are two related but distinct methodologies related to software development and delivery."

"DevOps is a methodology that emphasizes collaboration and communication between development and operations teams, whereas DevSecOps is an extension of DevOps that integrates security practices throughout the software development lifecycle."





"It aims to shift security "left" in the development process, meaning addressing security throughout the software development lifecycle, from design to deployment."

"While DevOps is focused on delivering software continuously and efficiently, DevSecOps is focused on delivering secure software continuously and efficiently."



What do you think are the key cultural aspects of DevSecOps?


Answer: “The Key principles of DevSecOps are culture, automation, measurement, and sharing (CAMS). Culture is the most important principle. If we do not have the right culture, then everything else falls apart. If each of these principles are not followed, then it will have adverse effects.”



How do you promote collaboration and communication in a DevSecOps culture?

Answer: "Collaboration and communication are essential for successful DevSecOps practices. To promote collaboration, cross-functional teams can be established that include members from development, security, and operations. Regular team meetings and stand-ups can facilitate communication and ensure that everyone is aware of the project's status and any security concerns."





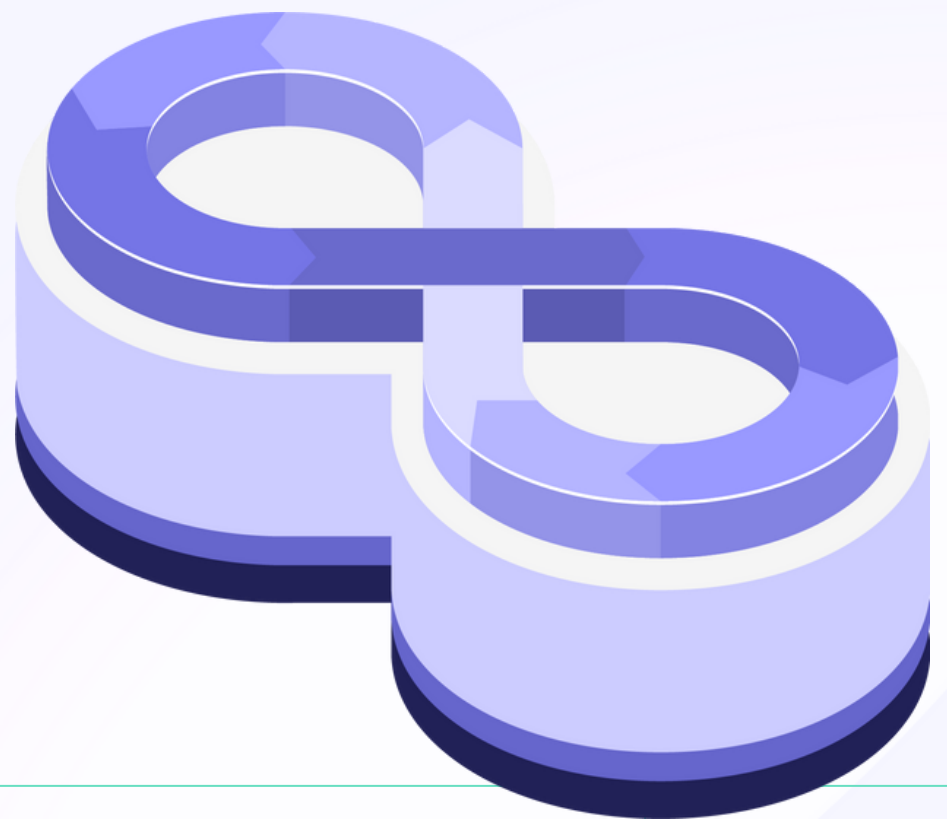
“Additionally, using collaboration tools such as chat applications and project management software can help facilitate communication and collaboration between team members.”



Level up your DevSecOps skills with us!

Certified DevSecOps Professional Course

Link in the description





**Practical
DevSecOps**

Making Product Security Accessible to Everyone