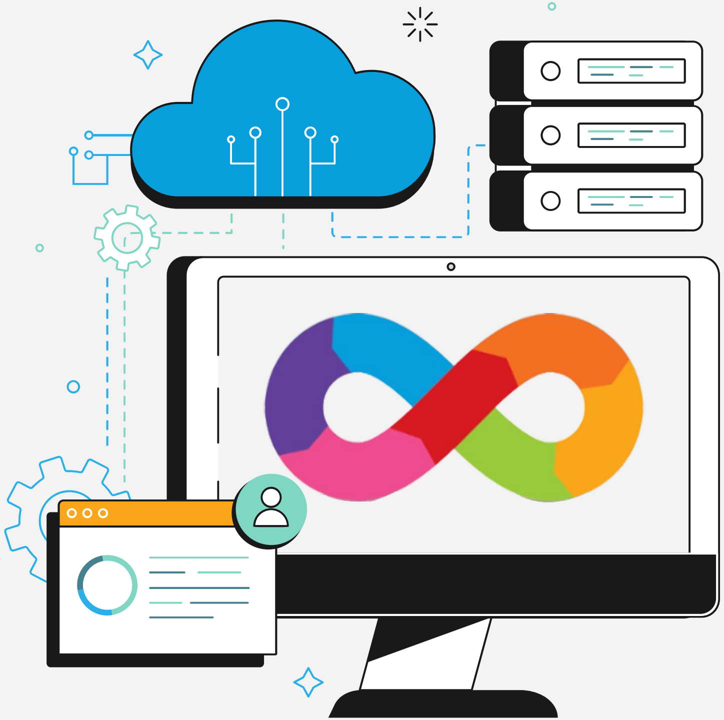
Introduction to DevOps

@ IBA - SMCS

Week 14 - 1

DevSecOps



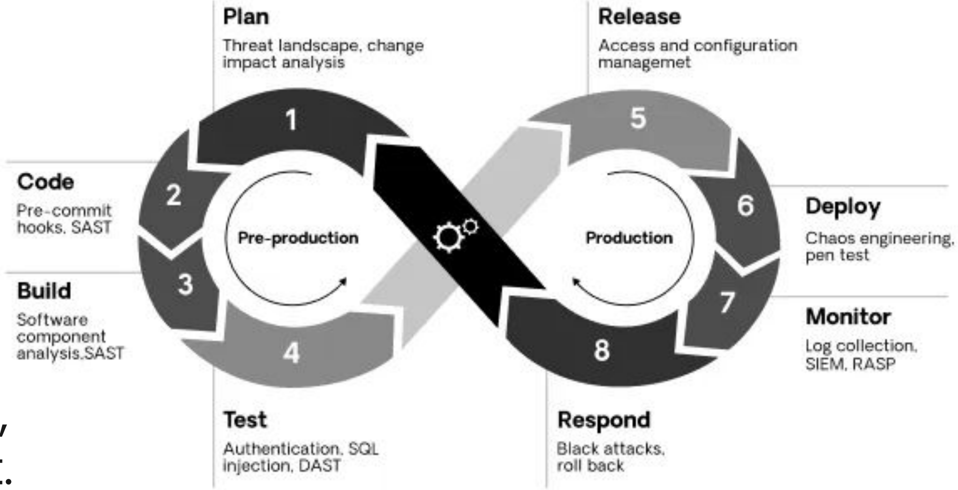
Obaid ur Rehman
Software Architect / Engineering Manager @ Folio3

DevSecOps

- What is DevSecOps
- Security?
- DevOps V/s DevSecOps
- DevSecOps Practices
- DAST
- SAST
- SBOM & SCA
- DevSecOps Pipeline
- Hands-on

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.



Plan
Threat landscape, change impact analysis
Release
Access and configuration managemet
Code
Pre-commit hooks, SAST
Build
Software component analysis,SAST
Deploy
Chaos engineering, pen test
Monitor
Log collection, SIEM, RASP
Test
Authentication, SQL injection, DAST
Respond
Black attacks, roll back
1
2
3
4
5
6
7
8
Pre-production
Production

What is DevSecOps

“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…”

Shannon Lietz, co-author of the “DevSecOps Manifesto.”

https://www.devsecops.org/blog/2015/2/15/what-is-devsecops

Why Security? - Case in Point

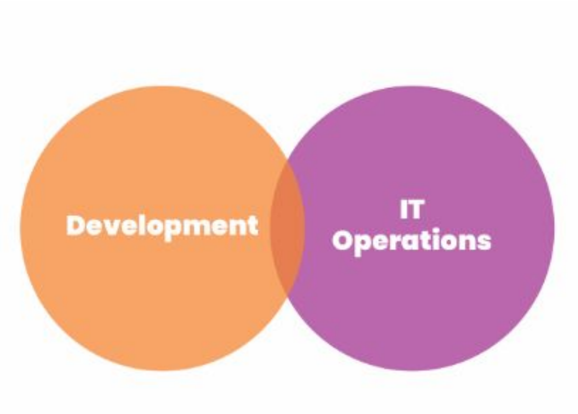
🪲CVE-2024-3094: backdoor in upstream xz/liblzma leading to ssh server compromise (March 2024)

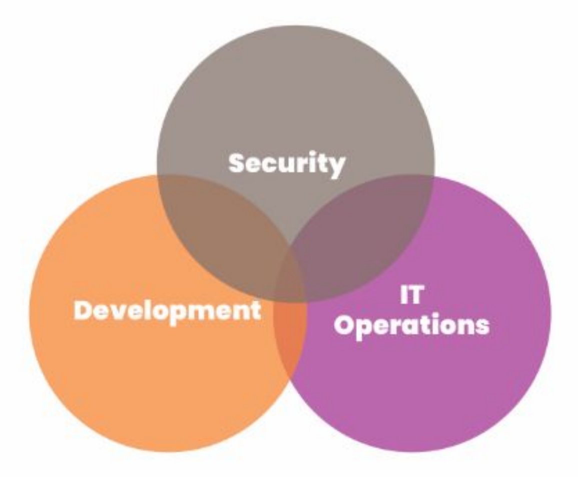
[https://seclists.org/oss-sec/2024/q1/268]

A **supply chain attack** is a type of cyberattack that targets a trusted third-party vendor who offers services or software vital to the supply chain.

Software supply chain attacks inject malicious code into an application in order to infect all users of an app

DevOps V/s DevSecOps





DevOps V/s DevSecOps

Both DevOps and DevSecOps. Both of these approaches emphasize collaboration and communication between developers and operations staff.

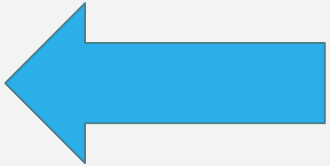
There is a key difference between the two: DevOps focuses on speed and efficiency, while DevSecOps puts a greater emphasis on security.

DevOps V/s DevSecOps

Criteria	DevOps	DevSecOps
Focus 👁️	Streamlines the collaboration between software development (Dev) and IT operations (Ops).	Adds a security (Sec) dimension to the DevOps approach, integrating security aspects at all software development and operation stages.
Security 🔐	Security checks often implemented **towards the end** of the development process or as a separate process.	Security is embedded from the project's inception and integrated throughout all phases of the development process ('shift-left').
Benefits 🌟	Faster and more reliable software delivery due to efficient collaboration and automation.	All the benefits of DevOps, + early and continuous identification and mitigation of security issues, leading to more secure and reliable products.
Tools 🔨⛏️	Tools primarily facilitate the CI/CD process.	In addition to DevOps tools, it uses tools to automate and integrate security checks, such as code analysis tools and continuous security monitoring.

Best practices for DevSecOps

<u>Shift left</u> is a DevSecOps mantra: It encourages software engineers to move security from the right (end) to the left (beginning) of the DevOps (delivery) process.



In a DevSecOps environment, security is an integral part of the development process from the beginning.

Shift Security to the Left

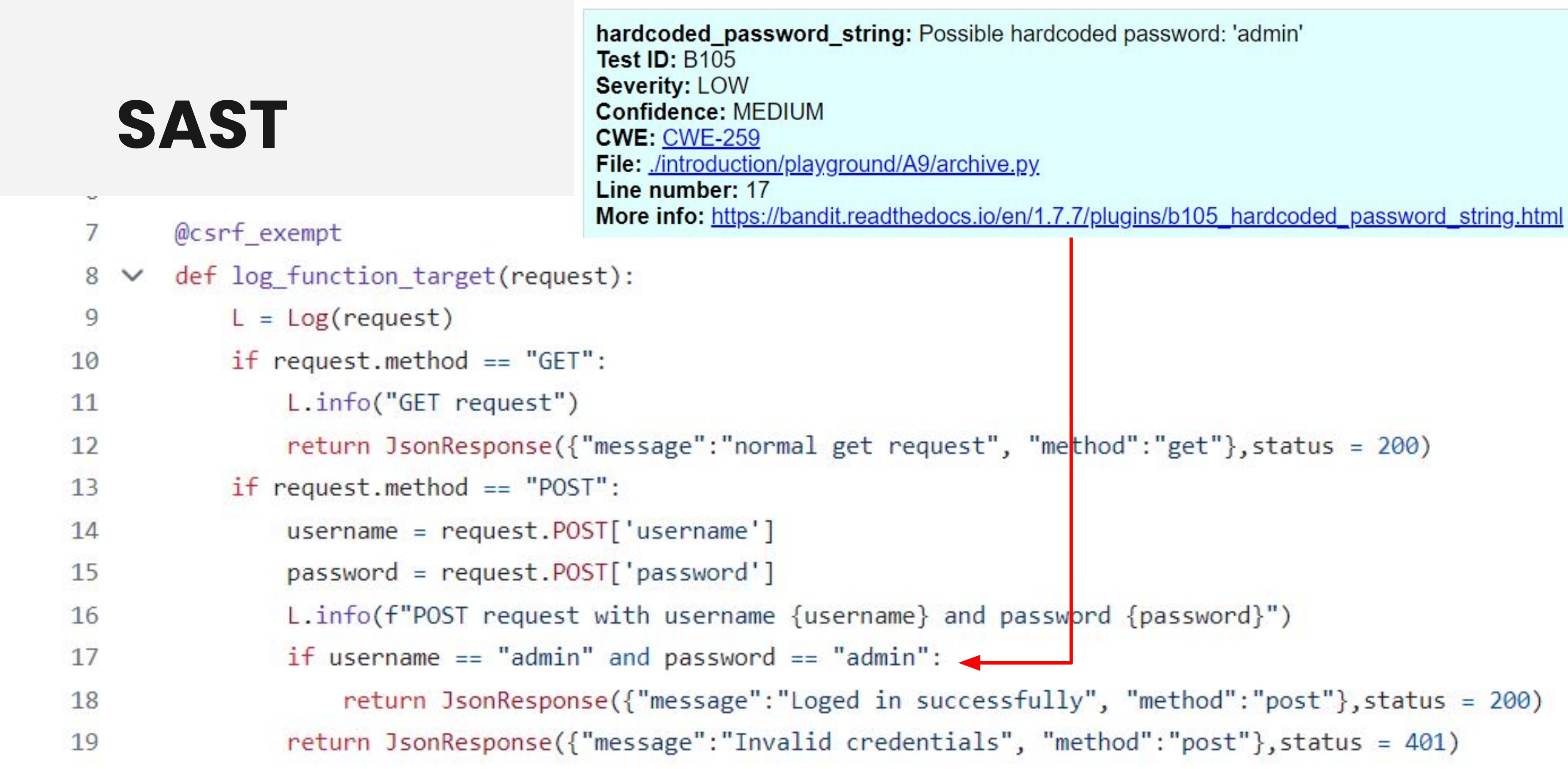
Shift left security will ensures that:

- Vulnerabilities are not discovered late in the software development cycle.
- **Early Remediation:** Notifications are sent whenever potential vulnerabilities are committed, enabling to quickly detect and correct security issues as part of the development phase.
- **Cost of remediation** is the lowest possible as real-time is far less costly than fixing days laters at deployment or even worse when a penetration test report outlines the vulnerability

Shift left security tools: SAST

SAST: Static application security testing, or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities in the code. SAST scans an application before the code is compiled. It's also known as white box testing.

SAST



hardcoded_password_string: Possible hardcoded password: 'admin'
Test ID: B105
Severity: LOW
Confidence: MEDIUM
CWE: CWE-259
File: ./introduction/playground/A9/archive.py
Line number: 17
More info: https://bandit.readthedocs.io/en/1.7.7/plugins/b105_hardcoded_password_string.html

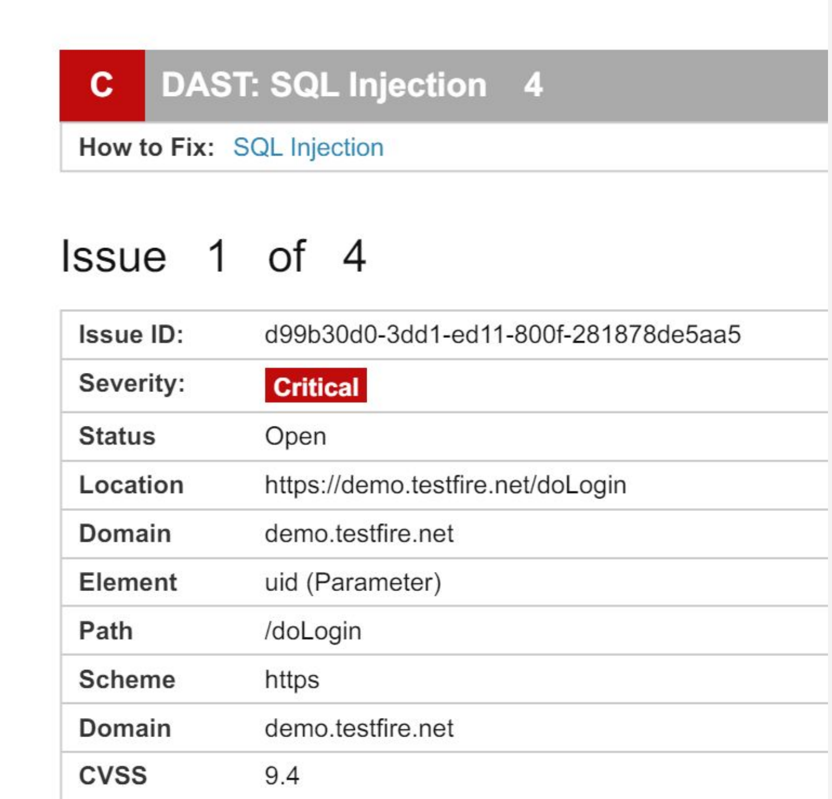
```
 7   @csrf_exempt
 8   def log_function_target(request):
 9       L = Log(request)
10       if request.method == "GET":
11           L.info("GET request")
12           return JsonResponse({"message":"normal get request", "method":"get"},status = 200)
13       if request.method == "POST":
14           username = request.POST['username']
15           password = request.POST['password']
16           L.info(f"POST request with username {username} and password {password}")
17           if username == "admin" and password == "admin":
18               return JsonResponse({"message":"Loged in successfully", "method":"post"},status = 200)
19           return JsonResponse({"message":"Invalid credentials", "method":"post"},status = 401)
```

Shift left security tools: DAST

Dynamic application security testing (DAST) is a type of black-box testing that checks your application from the outside.

Software systems rely on inputs and outputs to operate. A DAST tool uses these to check for security problems while the software is actually running.

DAST



C DAST: SQL Injection 4

How to Fix: SQL Injection

Issue 1 of 4

Issue ID:	d99b30d0-3dd1-ed11-800f-281878de5aa5
Severity:	Critical
Status	Open
Location	https://demo.testfire.net/doLogin
Domain	demo.testfire.net
Element	uid (Parameter)
Path	/doLogin
Scheme	https
Domain	demo.testfire.net
CVSS	9.4

Opensource != Secure

Case in point: log4j CVE

SBOM

A software Bill of Materials (SBOM) is a list of all the open source and third-party components present in a codebase. An SBOM also lists the licenses that govern those components, the versions of the components used in the codebase, and their patch status, which allows security teams to quickly identify any associated security or license risks.

Software Component Analysis (SCA)

Software Composition Analysis (SCA) is an application security methodology for managing open source components.

Using SCA, teams can quickly track and analyze any open-source component brought into a project.

SCA tools can discover all related components, their supporting libraries, and their direct and indirect dependencies. SCA tools can also detect software licenses, deprecated dependencies, as well as vulnerabilities and potential exploits. The scanning process generates a bill of materials (BOM), providing a complete inventory of a project’s software assets.

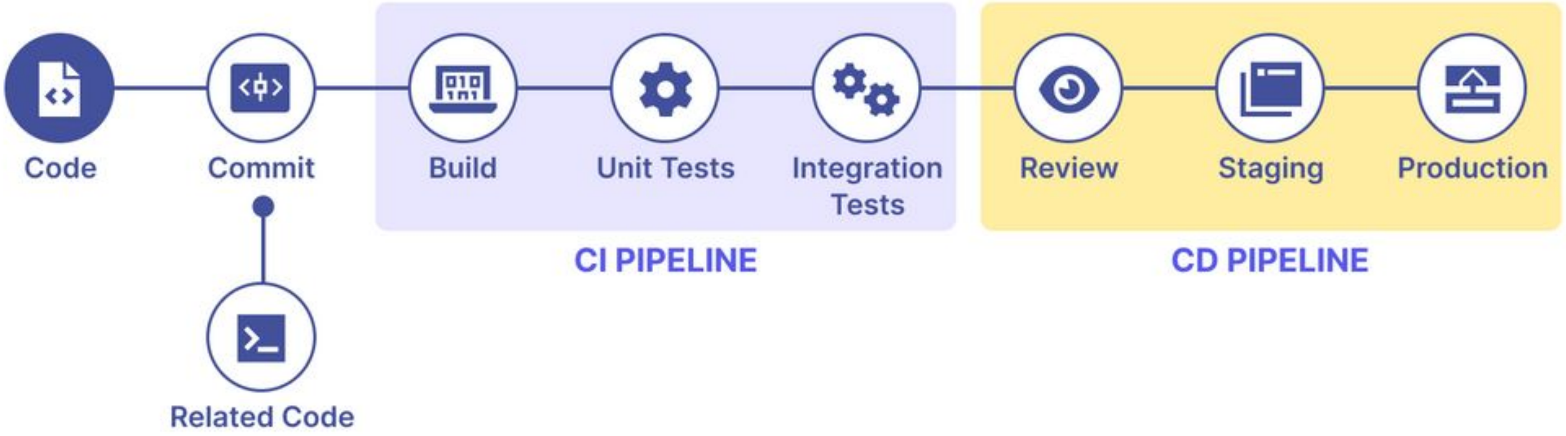
Infrastructure as Code (IaC) Security

Implement security checks and validations within the infrastructure code to ensure secure provisioning and configuration of cloud resources. Tools like Terraform, AWS CloudFormation, and Azure Resource Manager Templates can be used to define and manage infrastructure securely.

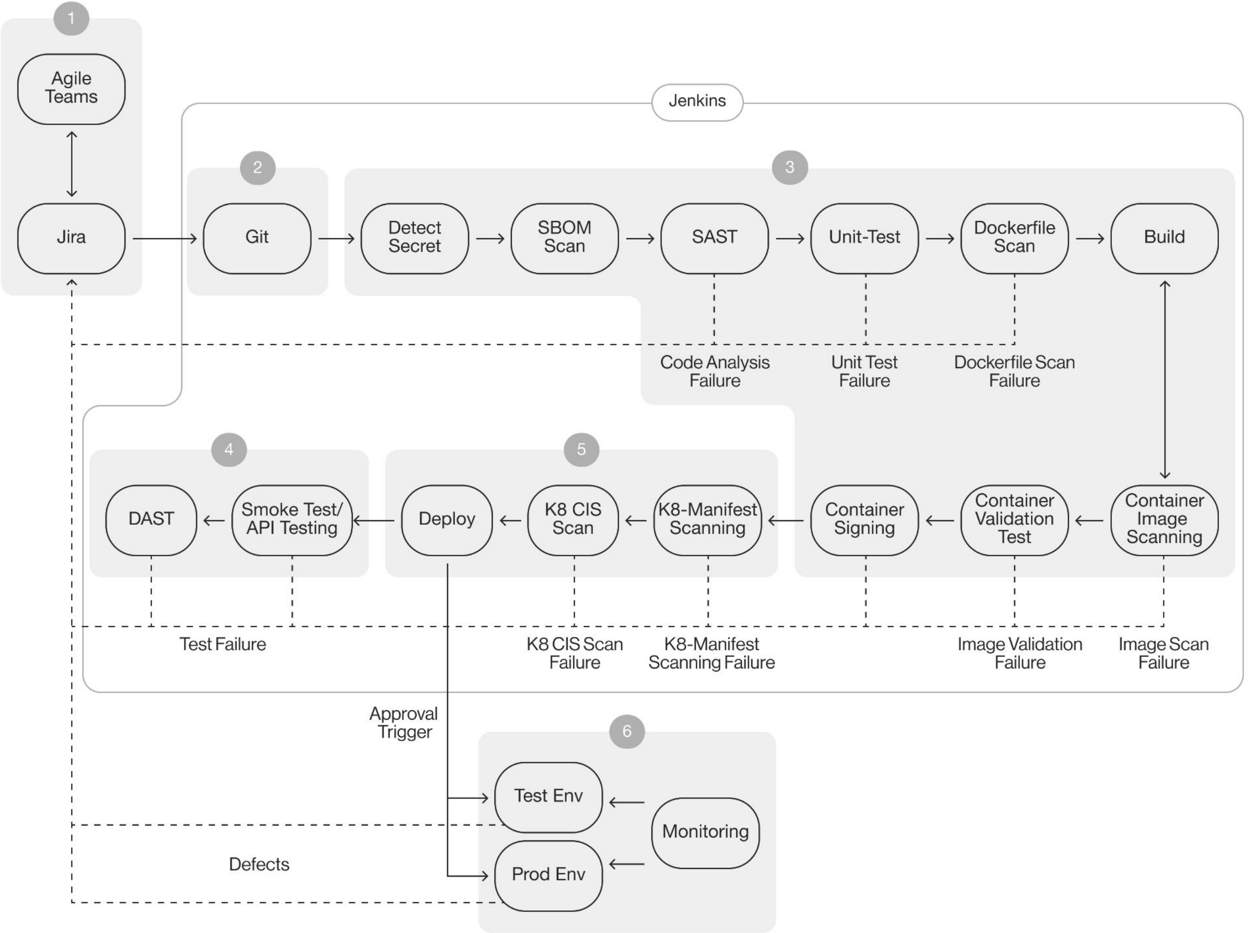
Container Security

Enhance container security by incorporating techniques such as image vulnerability scanning, container runtime security, and secure container orchestration. Tools like Docker Security Scanning (Scout), be used to strengthen container security.

A typical CI/CD Pipeline



DevSecOps Pipeline



DevSecOps Tool chain

SAST: Bandit nodejsscan snyk

SBOM: aqua trivy

Container Scan: Docker Scout

K8 Manifest Scanners: aqua kube-bench

DAST: Pentest Tools

DevSecOps Pipeline

https://github.com/ObaidUrRehman/pygoat

Reference

- https://github.com/devsecops/awesome-devsecops
- https://github.com/OWASP/DevSecOpsGuideline

End

Q&A

https://github.com/marcel-dempers/docker-development-youtube-series/tree/master/kubernetes