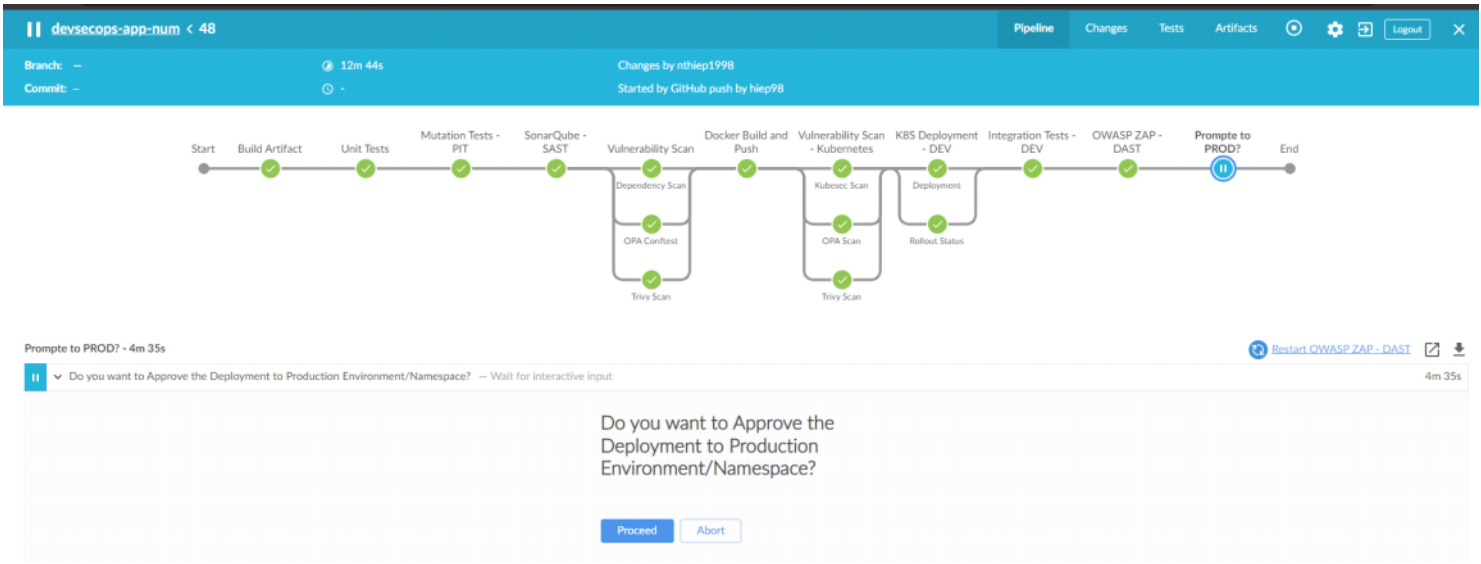
Overview DevSecOps pipeline



In this project, we will implement stages DevSecOps as follows:

1. Build Artifact
The first step is to build the artifact, which is the compiled Java Springboot application. This can be done using a variety of tools, such as Maven or Gradle.

2. Unit Tests
Once the artifact has been built, you can run unit tests to ensure that individual units of code are functioning correctly. Unit tests typically focus on testing small units of code in isolation, such as individual methods or classes. Unit tests are typically written by developers and are executed as part of the build process.

3. Mutation Tests
This stage performs mutation testing on the Java SpringBoot application artifact using the pitest-maven plugin. Mutation testing involves injecting artificial faults or mutations into the code to assess the effectiveness of the unit tests in detecting those faults. Mutation tests are designed to detect faults in the application code that may have been missed by traditional unit tests.

4. SonarQube SAST

This stage performs Static Application Security Testing (SAST) using SonarQube. SonarQube scans the Java SpringBoot application artifact for security vulnerabilities, code smells, and other code quality issues. It provides feedback on coding best practices, security vulnerabilities, and helps to ensure the codebase is secure and maintainable.

5. Vulnerability Scan
A vulnerability scan is a security scan that identifies known vulnerabilities in the software components, configurations or
code. Vulnerability scans can be performed manually or using automated tools. In this project, this stage includes parallel dependency scan, OPA conftest docker scan, and trivy scan docker image.

- Dependency scan is a technique that checks for vulnerabilities in the libraries or packages that your application depends on and ensuring that the latest security patches are applied to them. Using a tool like OWASP Dependency Check.

- OPA Conftest Docker scan: Open Policy Agent (OPA) is a general-purpose policy engine that can be used to enforce policies across various systems, including Kubernetes. Conftest is a utility that helps you write tests against structured configuration data using OPA. In this task, we are using Conftest to scan your Docker configuration files (e.g., Dockerfile, docker-compose.yml) to ensure they adhere to best practices and your organization's security policies. This helps you maintain a secure and compliant container environment.

- Trivy scan docker image: Trivy is a comprehensive vulnerability scanner for container images. It scans your Docker images for known vulnerabilities in the operating system packages and application dependencies. Trivy also checks for misconfigurations in your Docker images that could lead to security risks. By scanning docker images with Trivy, we can identify and fix vulnerabilities before deploying your containers to Kubernetes.

OPA conftest Docker scan and Trivy Docker scan serve different purposes in the vulnerability scanning process. Here's a comparison of the two:

OPA conftest Docker scan:
- OPA (Open Policy Agent) is a general-purpose policy engine that can be used to enforce policies across various domains, including Kubernetes, Docker, and more.
- Conftest is a utility that helps you write tests against structured configuration data using OPA.
- OPA conftest Docker scan focuses on evaluating Docker configuration files (e.g., Dockerfile, docker-compose.yml) against custom or predefined policies. These policies can cover best practices, security standards, and organization-specific requirements.
- The primary goal is to ensure that your Docker configurations are compliant with the desired policies and do not introduce potential security risks or misconfigurations.

Trivy Docker scan:
- Trivy is a vulnerability scanner specifically designed for containers and application dependencies.
- Trivy Docker scan focuses on scanning the Docker image for known vulnerabilities in the operating system packages and application dependencies.
- The primary goal is to identify and report any known security vulnerabilities in the packages and libraries used within the Docker image. It compares the image against known vulnerability databases like CVE/CWE and provides a CVE ID when vulnerabilities are detected.

Why use both tools?
Using both OPA conftest Docker scan and Trivy Docker scan provides a more comprehensive assessment of your Docker images' security:
- OPA Conftest can prevent security issues by enforcing security policies, such as ensuring no or the correct user is used in the Dockerfile, and Trivy scan docker can detect and report issues that could be exploited.
- OPA conftest ensures that your Docker configurations are compliant with best practices and security standards, reducing the risk of misconfigurations.
- Trivy identifies known vulnerabilities in the operating system packages and application dependencies, helping you to address potential security risks before deploying the image.
By combining the strengths of both tools, you can achieve a higher level of confidence in the security and compliance of your Docker images.

6. Docker Build and Push

This stage builds Docker images of the Java SpringBoot application artifact and pushes them to a container registry, such as Docker Hub or a private registry. Docker images are used for packaging and deployment of the application in containers, which provide consistency and reproducibility in deployment across different environments.

7. K8s Scan
This stage involves scanning the Kubernetes cluster for security risks and vulnerabilities. This stage may have multiple parallel phases, such as:

- Kubesec focuses on analyzing Kubernetes resource configurations for security risks. It performs static analysis Kubernetes YAML, JSON, or Helm charts and provides a risk score based on the configuration settings. Kubesec checks for best practices and potential security risks, such as running containers as a non-root user, enforcing resource limits, and using the latest image versions. By including Kubesec in your pipeline, you can identify and mitigate potential security risks in your Kubernetes configurations before they are deployed to production.

- OPA conftest k8s scan: In this parallel, OPA will scans the Kubernetes configuration files and policies with Open Policy Agent (OPA) conftest tool, to ensure that they follow security best practices.
A conftest scan can identify issues such as security configuration violations, namespace isolation violations, and Kubernetes deployment issues.

- Trivy K8s Scan: In the context of Kubernetes, Trivy can be used to checks for vulnerabilities in the operating system packages and application dependencies that are part of the image. By scanning the images used in Kubernetes environment and configuration issues, such as overly permissive Pod Security Policies, Kubernetes API state issues, and exposed secrets. we can identify and address potential security vulnerabilities before deploying to a production environment.

In summary, each parallel scan in your Kubernetes scanning stage serves a different purpose:
Kubesec focuses on analyzing Kubernetes resource configurations for security risks.
OPA Conftest enforces custom policies and compliance requirements on your Kubernetes configurations.
Trivy scans container images for known vulnerabilities in the OS packages and application dependencies.
By including all three parallel scans in your pipeline, you can ensure a comprehensive security and compliance assessment of your Kubernetes environment, reducing the risk of security breaches and maintaining consistency across your infrastructure.

8. K8s Deployment - DEV

Once the Docker image has been created, you can deploy it to a Kubernetes cluster in development. This will allow you to test the application in a production-like environment.

9. Integration Tests - DEV

Once the application has been deployed to development, you can run integration tests to verify that it is working as expected. Integration tests typically involve testing the application's interaction with other systems.

10. OWASP ZAP - DAST

OWASP ZAP is a dynamic analysis tool that can be used to scan web applications for potential security vulnerabilities.
It can be integrated with your development environment to automatically scan the application for vulnerabilities as part of the development process.

11. Prompte to Production

This stage promotes the Java SpringBoot application artifact to production-ready status based on the results of previous stages. This may involve additional checks, approvals, and validations before deploying the application to the production environment.

12. K8s CIS Benchmark

This stage running a Kubernetes CIS Benchmark assessment using the kube-bench tool to evaluate the security posture of the Kubernetes cluster.

The assessment involves verifying the Kubernetes configuration settings against the industry-

standard CIS Benchmark guidelines to ensure that the cluster is configured securely. This stage typically involves multiple phases, such as:

Configuration Audit: This phase involves reviewing the Kubernetes cluster configuration settings against the CIS Benchmark guidelines and identifying any deviations or potential security risks.

Vulnerability Assessment: This phase involves scanning the Kubernetes cluster for any known vulnerabilities and ensuring that the latest security patches are applied to the Kubernetes components.

Compliance Checks: This phase involves verifying that the Kubernetes cluster is compliant with any relevant regulatory requirements or security standards.

13. K8s Deployment - PROD

This stage deploys the tested and secured application to the production environment running on Kubernetes.

14. Integration Tests - Production

Once the application has been deployed to production, you can run integration tests to verify that it is working as expected. Integration tests typically involve testing the application's interaction with other systems.

Benefits of devsecops pipeline:

Each stage of the DevSecOps pipeline has its own benefits. For example, unit tests can help to identify potential bugs in the code, while vulnerability scans can help to identify known vulnerabilities in the code. By implementing all of the stages in the DevSecOps pipeline, you can help to ensure that your application is secure.