Secure DevOps – Project 3 (ISEC6000)

Student: Vrushtiben Patel
Student ID: 22167521
Course: Master of Computing (Computer Science)
Unit: ISEC6000 – Secure DevOps
Submission Date: 1 November 2025
GitHub Repository: Vrushti54/SecureDevOps-Assignment3-22167521

Overview

This project implements a complete **DevSecOps pipeline** demonstrating secure containerization, continuous integration, and threat modelling across six key tasks:

1. **Docker Setup & Validation** – Secure installation and privilege restriction.
2. **Portainer Deployment** – GUI-based container management and log auditing.
3. **Nextcloud + PostgreSQL Stack** – Multi-service deployment using Docker Compose.
4. **Clair Vulnerability Scanning** – Automated container image scanning and CVE mitigation.
5. **AWS CI/CD Simulation** – Source → Build → Test → Deploy pipeline with IAM role.
6. **STRIDE Threat Modelling** – Risk assessment using Microsoft Threat Modelling Tool.

Each task progressively builds a secure, automated environment following DevSecOps best practices.

Project Structure

Path / File	Description
`/docs/P3_22167521.pdf`	Final project report (PDF submission)
`/clair.sh`	Bash script automating Clair scans
`/docker-compose.yml`	Multi-service stack definition for Nextcloud and Clair
`/ubuntu-22_04-vulns.csv`	Sample vulnerability report generated by Clair
`/img/`	All evidence screenshots (Figures 1–27)

Tools & Technologies

Category	Tools / Services
Containers & Orchestration	Docker, Docker Compose
Monitoring & Management	Portainer
Security & Scanning	Clair, STRIDE Framework
CI/CD Simulation	AWS CodePipeline, IAM

Category	Tools / Services
Languages & Scripts	Bash, Node.js
Database	PostgreSQL
Reporting & Visualization	Portainer Dashboard, CSV Outputs

Key Outcomes

- Verified secure Docker configuration using least-privilege enforcement.
- Deployed and monitored containers through Portainer with restart policies.
- Integrated Nextcloud with PostgreSQL for persistent storage.
- Automated vulnerability detection using Clair with CSV export comparison (Ubuntu vs Alpine).
- Simulated AWS CI/CD pipeline stages and cross-account IAM integration.
- Applied STRIDE threat modelling to identify and mitigate potential security risks.

Highlights

Feature	Screenshot Reference
Docker installation verification	`img/T1-1_docker-version.png`
Portainer log inspection	`img/T2-3_portainer-filtered.png`
Nextcloud dashboard	`img/T3-2_nextcloud-dashboard.png`
Clair API and vulnerability scan	`img/T4-2_clair_openapi.png`
AWS CI/CD pipeline flow	`img/T5-7_pipeline-summary.png`
STRIDE Threat Model	`img/T6-STRIDE-diagram.png`

Learning Reflection

Through this project, I developed practical competence in embedding security within every DevOps stage. Key learning points include:

- Applying **least-privilege** and **configuration hardening** in Docker environments.
- Automating vulnerability detection with Clair and interpreting CVE data.
- Simulating enterprise-grade CI/CD pipelines using AWS principles.
- Designing secure architectures using **STRIDE threat modelling**.

This project reinforced the importance of **continuous assurance** and **security-by-design** in maintaining DevSecOps maturity across modern cloud-native environments.

Submission Artifacts

Artifact	Path

Artifact	Path
Final Report (PDF)	`/docs/P3_22167521.pdf`
Screenshots	`/img/`
STRIDE Diagram	`/img/T6-STRIDE-diagram.png`
CSV Vulnerability Report	`/ubuntu-22_04-vulns.csv`
Scripts	`/clair.sh`, `/docker-compose.yml`

References

Docker, Inc. (2024). *Docker documentation.* https://docs.docker.com
Portainer Ltd. (2024). *Portainer documentation.* https://docs.portainer.io
Nextcloud GmbH. (2024). *Nextcloud admin guide.* https://docs.nextcloud.com
Quay.io. (2024). *Clair vulnerability scanner.* https://github.com/quay/clair
Amazon Web Services. (2024). *AWS CodePipeline user guide.* https://docs.aws.amazon.com/codepipeline
Microsoft Corporation. (2024). *STRIDE threat modeling tool.* https://learn.microsoft.com/en-us/security/threat-modeling-tool