AUTOMATING WEB APP DEPLOYMENT & ELK MONITORING USING DEVSECOPS

PRESENTED BY :

- Samrajya Pujari (240844223037)
- Akshat Sisodiya (240844223003)
- Jaspreet Sindhu (240844223018)
- Dhanashree Dhondge (240844223011)
- Divyansh Baghel (240844223013)

INTRODUCTION

The project focuses on deploying a web application through a secure and automated CI/CD pipeline. By integrating **GitHub** for version control, **Jenkins** for automation, and **Docker** for containerized deployments, the pipeline ensures efficient software delivery. Security is enforced using **SonarQube** for code quality analysis, **OWASP ZAP** for dynamic application security testing (DAST), and **Trivy** for container vulnerability scanning.

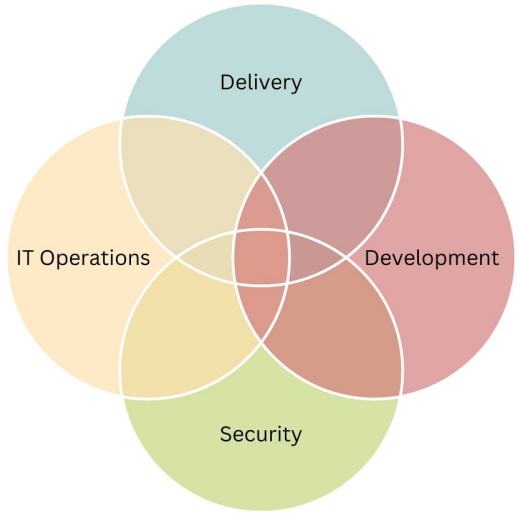
Minikube provides a lightweight Kubernetes environment for deployment testing, while the **ELK stack** (Elasticsearch, Logstash, and Kibana) enables time monitoring and logging.

A key feature is **e-mail notifications** configured through **Jenkins**, which notify the admin whenever a pipeline stage is successfully completed.

This enhances monitoring and transparency, allowing prompt action in case of failures.

The system ensures reliable, scalable, and secure software delivery with real-time security checks and monitoring.

DevSecOps

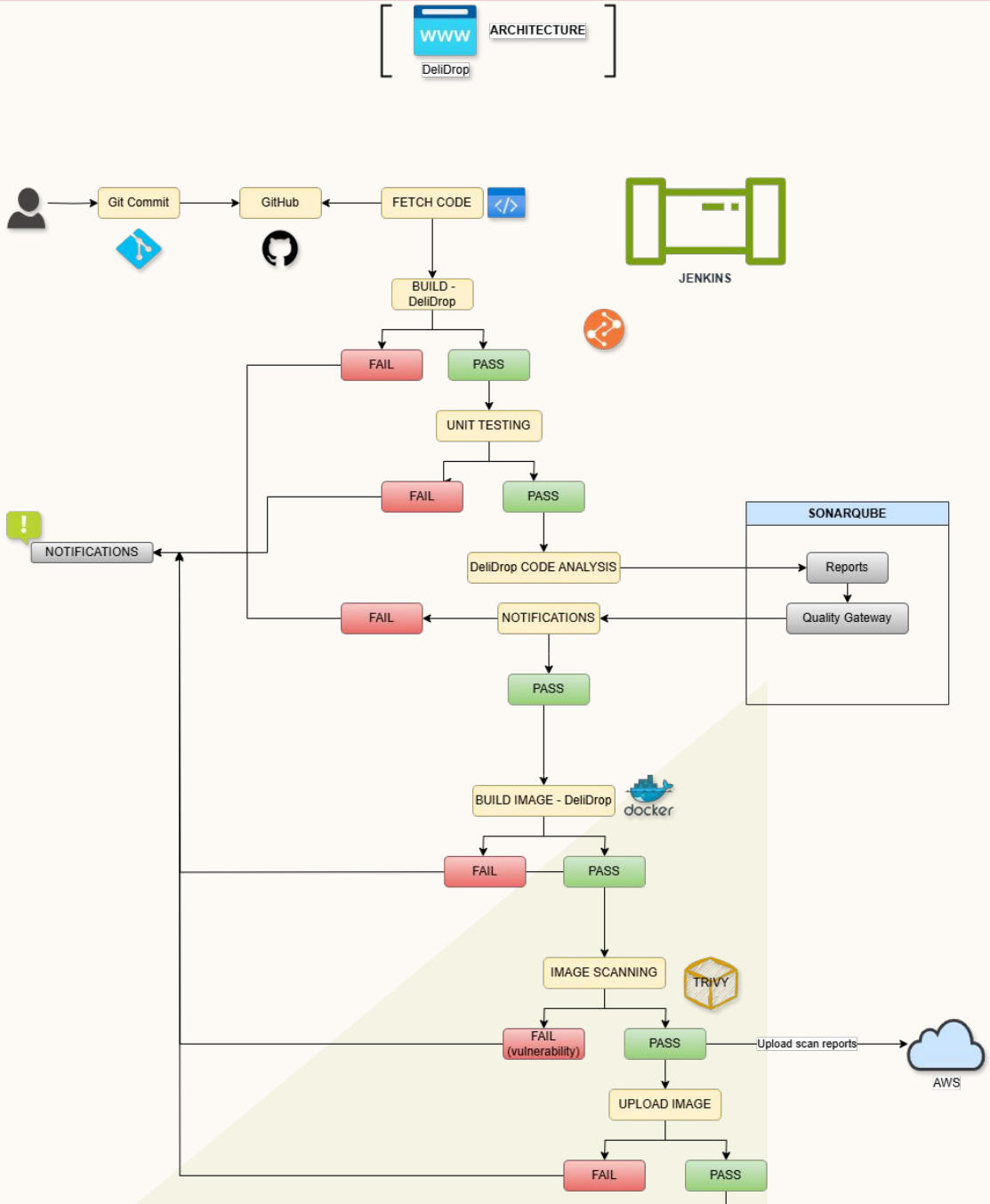


APPLICATIONS

A secure CI/CD pipeline for web app deployment helps by:
Saving Time & Effort – Automates testing, security checks, and deployments, reducing manual work.
Enhancing Security – Detects vulnerabilities early, preventing security breaches.
Ensuring Stability – Continuous monitoring (ELK Stack) helps quickly identify and fix issues.
Improving Efficiency – Faster updates mean users get new features and bug fixes quickly.
Reducing Risks – Automated compliance checks ensure the app meets security standards.

Overall, it streamlines development, enhances security, and ensures a smooth user experience.

WORKFLOW


ARCHITECTURE
DeliDrop
Git Commit
GitHub
FETCH CODE
JENKINS
BUILD - DeliDrop
FAIL
PASS
UNIT TESTING
FAIL
PASS
SONARQUBE
NOTIFICATIONS
DeliDrop CODE ANALYSIS
Reports
FAIL
NOTIFICATIONS
Quality Gateway
PASS
BUILD IMAGE - DeliDrop
docker
FAIL
PASS
IMAGE SCANNING
TRIVY
FAIL (vulnerability)
PASS
Upload scan reports
AWS
UPLOAD IMAGE
FAIL
PASS

TECHNOLOGIES USED

- **GitHub** – A web-based platform for version control and collaboration, allowing developers to store, manage, and track code using Git. It supports issue tracking, CI/CD workflows, and open-source contributions.

- **Docker** – An open-source platform that enables developers to build, package, and deploy applications in lightweight, portable containers. It ensures consistency across different environments.

- **Jenkins** – A self-contained, open-source automation server used for continuous integration and continuous delivery (CI/CD). It automates building, testing, and deploying applications.

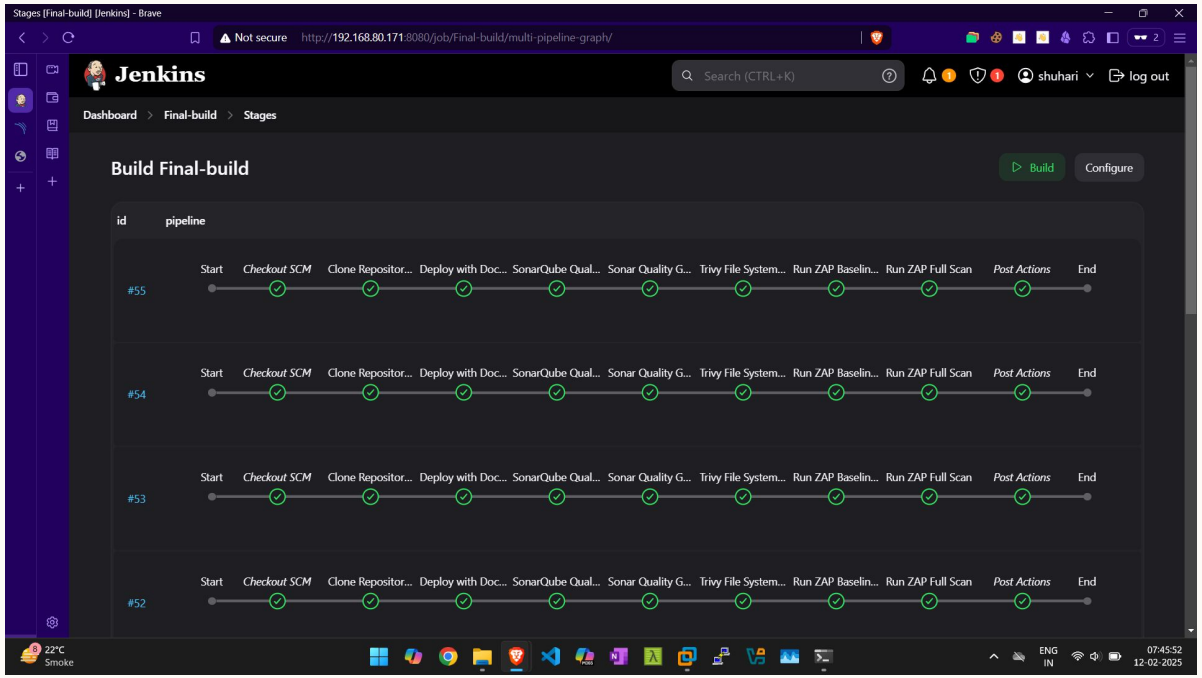
- **SonarQube** – A code quality and security analysis platform that inspects source code for bugs, vulnerabilities, and maintainability issues. It supports multiple programming languages and integrates with CI/CD pipelines.

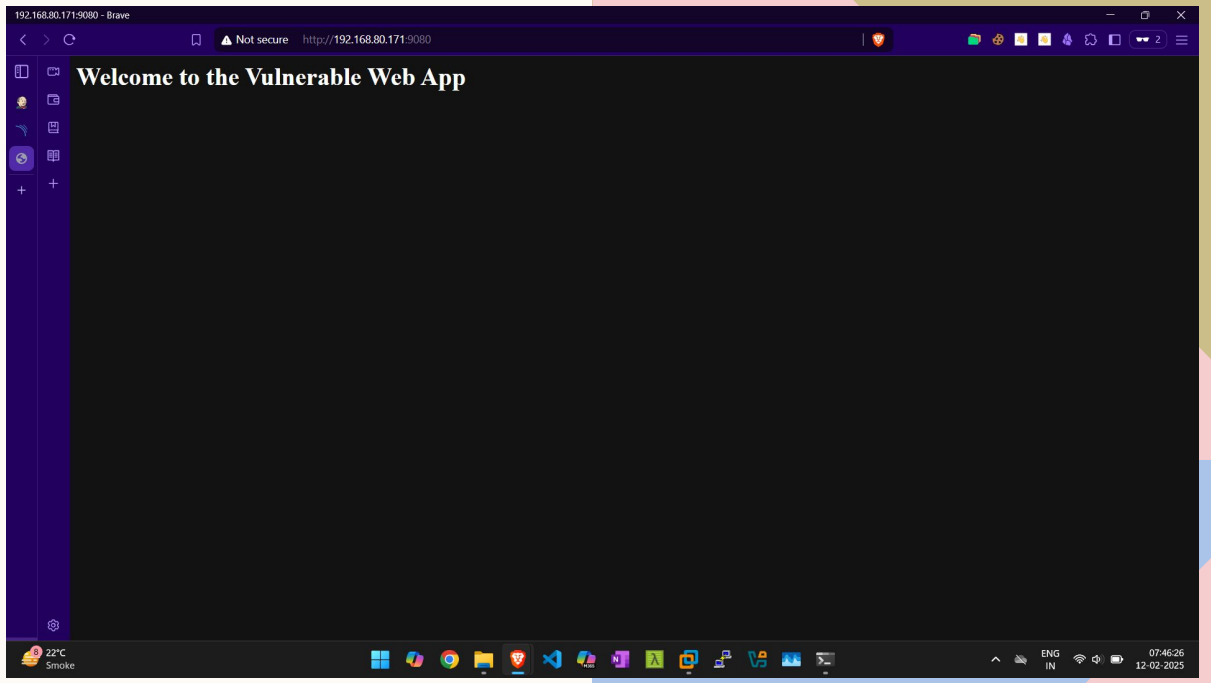
- **OWASP ZAP** – A dynamic application security testing (DAST) tool developed by OWASP to identify vulnerabilities in web applications. It helps detect security flaws like SQL injection and cross-site scripting (XSS).

- **Trivy** – A comprehensive security scanner for containers, Kubernetes, and infrastructure-as-code, detecting vulnerabilities, misconfigurations, and exposed secrets. It supports scanning Docker images, filesystems, and repositories.

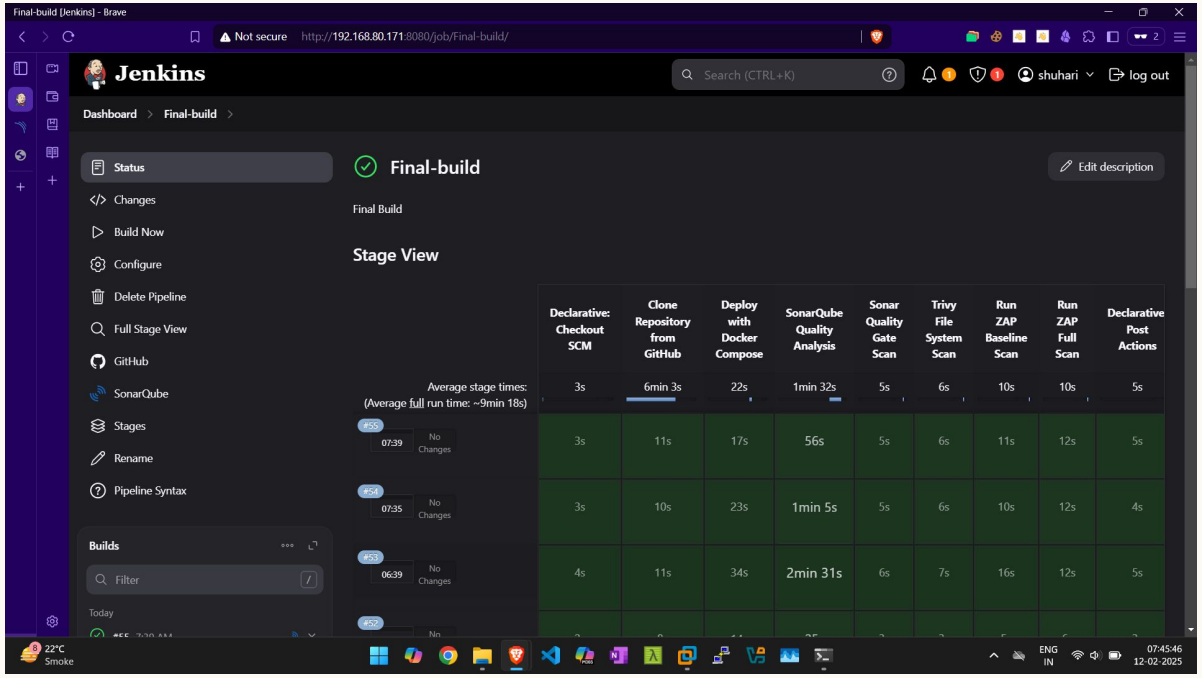
- **DockerHub** – A cloud-based repository for storing, sharing, and managing Docker container images. It provides access to pre-built images and enables collaboration in containerized application development.

OUTPUT


Stages [Final-build] [Jenkins] - Brave
Not secure http://192.168.80.171:8080/job/Final-build/multi-pipeline-graph/
Jenkins
Search (CTRL+K)
shuhari
log out
Dashboard > Final-build > Stages
Build Final-build
Build
Configure
id
pipeline
Start Checkout SCM Clone Repositor... Deploy with Doc... SonarQube Qual... Sonar Quality G... Trivy File System... Run ZAP Baselin... Run ZAP Full Scan Post Actions End
#55
#54
#53
#52
22°C Smoke
ENG IN
07:45:52 12-02-2025


192.168.80.171:9080 - Brave
Not secure http://192.168.80.171:9080
Welcome to the Vulnerable Web App
22°C Smoke
ENG IN
07:46:26 12-02-2025

OUTPUT


Final-build [Jenkins] - Brave
Not secure http://192.168.80.171:8080/job/Final-build/
Jenkins
Search (CTRL+K)
shuhari
log out
Dashboard
Final-build
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
GitHub
SonarQube
Stages
Rename
Pipeline Syntax
Builds
Filter
Today
Final-build
Edit description
Final Build
Stage View
Declarative: Checkout SCM
Clone Repository from GitHub
Deploy with Docker Compose
SonarQube Quality Analysis
Sonar Quality Gate Scan
Trivy File System Scan
Run ZAP Baseline Scan
Run ZAP Full Scan
Declarative Post Actions
Average stage times:
(Average full run time: ~9min 18s)
3s 6min 3s 22s 1min 32s 5s 6s 10s 10s 5s
#55 07:39 No Changes
3s 11s 17s 56s 5s 6s 11s 12s 5s
#54 07:35 No Changes
3s 10s 23s 1min 5s 5s 6s 10s 12s 4s
#53 06:39 No Changes
4s 11s 34s 2min 31s 6s 7s 16s 12s 5s
#52
22°C Smoke
ENG IN
07:45:46 12-02-2025


Search in emails
Primary
me 5:05 am
Pipeline Status: 51
Build Status: Successful Build Number: 51 Chec...
me 5:05 am
Build Success: Final-build - Full Scan
ZAP Full Scan completed successfully.
me 5:05 am
Build Success: Final-build - Baseline Scan
ZAP Baseline Scan completed successfully.
me 5:05 am
Build Success: Final-build - Trivy Scan
Trivy file system scan completed successfully.
me 5:05 am
Build Success: Final-build - Sonar Qualit...
Sonar Quality Gate passed successfully.
me 5:04 am
Build Success: Final-build - SonarQube A...
SonarQube analysis completed successfully.
me 5:03 am
Build Success: Final-build - Deployment...
Application deployed successfully using Docke...
me
Build Success: Final-build -
Compose

CONCLUSION

- **Seamless Automation** – The CI/CD pipeline ensures automatic code building, testing, security scanning, and deployment, minimizing manual effort.
- **Enhanced Security** – Integrated tools like SonarQube, OWASP ZAP, and Trivy detect vulnerabilities early, ensuring a secure web application.
- **Efficient Deployment** – Docker & Minikube provide a consistent and scalable deployment environment.
- **Real-time Monitoring** – ELK Stack enables continuous logging and monitoring, improving system reliability.
- **Optimized Collaboration** – GitHub & Jenkins streamline code management and CI/CD workflows, enhancing team productivity.

FUTURE SCOPE

- **Kubernetes Expansion** – Upgrade from Minikube to a full-scale Kubernetes cluster for better scalability and efficient orchestration.

- **Multi-Cloud Deployment** – Extend deployment to AWS, Azure, or GCP for enhanced availability, flexibility, and performance.

- **Advanced Security Integration** – Implement AI-driven security tools for real-time threat detection and automated vulnerability remediation.

- **Automated Incident Response** – Enhance ELK Stack with predictive analytics for faster issue detection and resolution.

These improvements will enhance scalability, security, and automation, ensuring long-term efficiency and reliability.

THANK YOU