Composable DevSecOps Architecture
The Need for Secure and Flexible Deployment

Parastou Moghaddam, Noha Elissawy, Harkaran Singh, Karan Sharma and Jeong-joo Park
Volgenau School of Engineering - Department of Cyber Security Engineering
Subject Matter Expert: Christine Kim, Vibha Dhawan **Project Manager:** Katy Warren

BACKGROUND:

• Cloud adoption is growing rapidly. Many companies are compelled to transition to a cloud environments due to the many benefits: less cost, mobility, efficiency, security, and most importantly scalability. This project will benefit our customer's research and understanding of cloud environments.
• Our team is designing and developing a secure Cloud Service Provider (CSP) independent DevSecOps Pipeline.
• The pipeline will be used along with Mattermost as an open source application. Our customer wants to automate the transfer of data to integrate development efforts.
• The pipeline needs to have a secure architecture, implementing continuous integration and delivery.

OBJECTIVES:

Strategy
• Provide a clear strategy for the construction of the pipeline and integration of cloud service providers.
Configuration
• Develop and set up a Cloud Service Provider Pipeline to automate the process of transferring data.
Integration
• The application must be integrated to work in a fast-paced environment.
Verification
• We can verify that we will meet the goal through testing the pipeline and controls.
Security
• Vulnerabilities that must be addressed are: reduced visibility and control, insider threat and trust relationship between the cloud servers and code repository.

ARCHITECTURE:

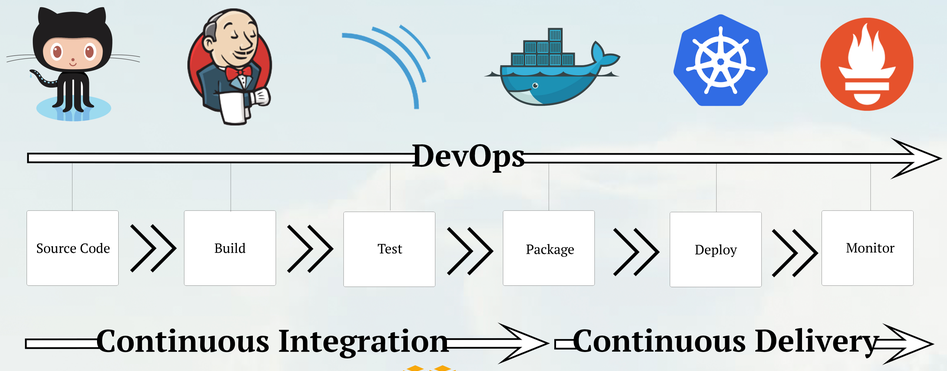

Figure 1: Architecture

ARCHITECTURE TOOLS OVERVIEW:

• GitHub - The most widely used SCM open source system.
• Jenkins - An open-source automation tool which enables developers to reliably build, test, and deploy software.
• SonarQube - An open-source platform for continuous inspection of code with static analysis.
• Docker Hub - An open-source container registry.
• Kubernetes - An open-source cluster and container management tool.
• Prometheus - A monitoring tool that gathers time-series based numerical data.

Figure 2: Prometheus monitoring the Kubernetes Cluster

VERIFICATION:

• Select an open source application for testing a DevSecOps pipeline.
• Verification through analysis.
• Analyze environment setup for AWS.
• Unit and Integration tests for DevSecOps pipeline.
• Smoke testing, security testing, performance testing.

VERIFICATION TESTING:

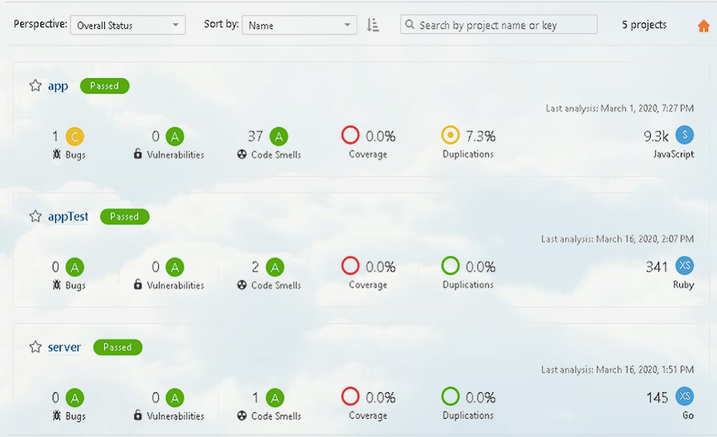

Figure 3: Security Testing from SonarQube

CONCLUSION:

Open-source applications optimize cost and eliminate vendor lock-in issues. Consequently, they empower businesses to have the flexibility of deployment at ease. Automation, containerization, and testing can all be achieved by opens-source tools. AWS allows for flexible deployment with proper cloud services, which can prevent vendor lock-in and bring any DevSecOps projects into reality.

ACKNOWLEDGEMENT:
The team would like to thank MITRE for constant guidance and for providing resources in order to complete the project. We would also like to thank Professor Sabetto for his valuable input throughout the entire design and implementation process.