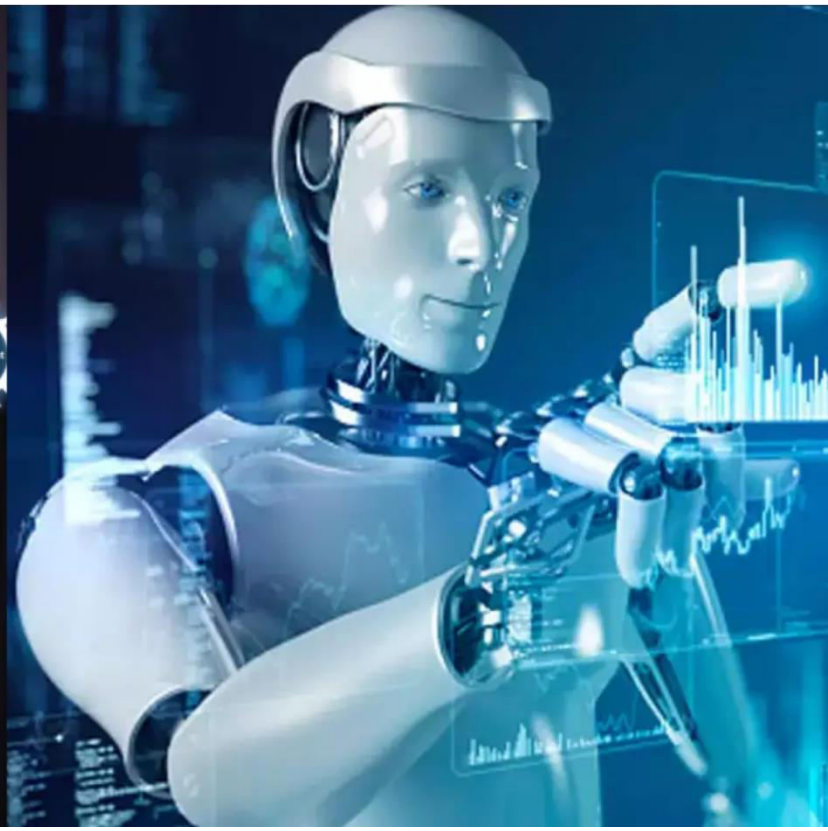
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 6, June 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506067|

Design and Implementation of a Zero-Trust DevSecOps Pipeline for Secure Cloud Application Deployment

Maya Pawar, Prof. S.G. Tathe, Dr. S.K. Korde

Student, Dept. of Computer Engineering, Vishwabharati Academy's College of Engineering, Ahilyanagar, Maharashtra, India

Professor, Dept. of Computer Engineering, Vishwabharati Academy's College of Engineering, Ahilyanagar, Maharashtra, India

ABSTRACT: With the growing use of cloud technologies, securing applications during development and deployment has become very important. Traditional security methods mainly focus on protecting the outer network boundary, but modern cloud environments require stronger and more continuous security practices. This project presents a Zero-Trust DevSecOps Pipeline for secure cloud deployment.

The proposed system follows the Zero-Trust principle of "Never Trust, Always Verify" by applying security checks at every stage of the CI/CD pipeline. The pipeline integrates automated security tools for source code analysis, vulnerability scanning, infrastructure security validation, and secure container deployment. It also includes policy enforcement and continuous monitoring to detect threats and unauthorized activities in real time.

By combining Zero-Trust concepts with DevSecOps practices, the system helps improve application security, reduce vulnerabilities, and ensure secure cloud deployment. The proposed approach also supports continuous compliance and minimizes the impact of potential cyberattacks in cloud environments.

KEYWORDS: Zero-Trust, DevSecOps, CI/CD, Cloud Security, SAST, DAST, Continuous Monitoring

I. INTRODUCTION

Cloud computing has transformed the way organizations develop, deploy, and manage software applications by providing scalability, flexibility, high availability, and cost-effective infrastructure. Modern enterprises increasingly adopt cloud-native technologies and Continuous Integration/Continuous Deployment (CI/CD) practices to accelerate software delivery and improve operational efficiency. However, the rapid adoption of cloud environments and automated deployment pipelines introduces new security challenges, including insecure code, vulnerable software dependencies, container security risks, and misconfigured deployments.

Traditional security approaches often treat security as a separate phase that occurs after development and deployment. This approach is no longer sufficient in modern cloud environments where software is continuously developed, tested, and released. To address these challenges, the DevSecOps methodology has emerged, integrating security controls directly into the software development lifecycle. DevSecOps enables organizations to identify and mitigate security vulnerabilities early in the development process while maintaining the speed and agility of DevOps practices.

In addition, the Zero-Trust security model has gained significant importance in securing modern applications and cloud infrastructures. The core principle of Zero Trust is "Never Trust, Always Verify," which assumes that no user, application, service, or workload should be trusted by default. Every component must undergo continuous verification and validation before gaining access to resources or progressing through the deployment pipeline.

This project presents the design and implementation of a Zero-Trust DevSecOps Pipeline for Secure Cloud Application Deployment on Google Cloud Platform (GCP). The proposed solution integrates source code management, continuous

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506067|

integration, security analysis, vulnerability scanning, automated deployment, and centralized security monitoring into a unified workflow. GitHub is used for source code management, Jenkins automates the CI/CD pipeline, SonarQube performs static code quality and security analysis, Trivy identifies container vulnerabilities, Docker provides application containerization, and Nginx acts as a reverse proxy for secure application access. A custom Flask-based Security Dashboard is developed to provide centralized visibility into build status, deployment status, code quality metrics, vulnerability information, and overall security posture.

The primary objective of this project is to demonstrate how Zero-Trust principles can be incorporated into a DevSecOps workflow to enhance the security and reliability of cloud-based application deployments. By implementing automated security validation at multiple stages of the CI/CD pipeline, the proposed system helps reduce security risks, improve deployment consistency, and establish a secure software delivery process suitable for modern cloud environments.

II. LITERATURE SURVEY

Recent research has highlighted the growing importance of integrating Zero Trust principles into modern DevSecOps environments to address security challenges in cloud-native applications and automated deployment pipelines. Zero Trust Architecture (ZTA) operates on the principle of continuous verification, ensuring that no user, application, or system component is trusted by default.Feng and Hu [1] proposed a Cyber-Physical Zero Trust Architecture (CP-ZTA) for Industrial Cyber-Physical Systems (ICPS). Their approach emphasizes continuous authentication, authorization, and monitoring mechanisms to secure interconnected industrial devices and communication channels. The study demonstrates the effectiveness of Zero Trust principles in reducing security risks within complex distributed environments.Sanders et al. [2] investigated the integration of Zero Trust concepts into DevSecOps practices. Their work focuses on embedding security controls directly into CI/CD pipelines through automation, identity verification, and least-privilege access models. The authors highlight that continuous security validation throughout the software development lifecycle can significantly improve application security and operational resilience.Patel [3] examined the application of Zero Trust and DevSecOps methodologies within cloud-native environments. The study presents security frameworks and best practices that incorporate automated vulnerability assessment, compliance validation, and security policy enforcement into CI/CD workflows. The findings indicate that proactive security integration enhances the overall security posture of cloud deployments.Karanam [4] explored the implementation of Zero Trust Architecture in cloud-native DevSecOps environments. The proposed framework combines continuous verification, identity-centric security controls, and automated policy enforcement to minimize attack surfaces and strengthen cloud application security. The study demonstrates the effectiveness of integrating Zero Trust mechanisms with DevSecOps automation to achieve secure software delivery.Although existing research provides valuable frameworks and conceptual models for combining Zero Trust and DevSecOps, practical implementations demonstrating end-to-end secure cloud deployment pipelines remain limited. The proposed work addresses this gap by implementing a Zero-Trust DevSecOps pipeline on Google Cloud Platform (GCP) using GitHub, Jenkins, SonarQube, Trivy, Docker, Nginx, and a centralized Security Dashboard. The implementation demonstrates how automated security validation and continuous monitoring can be integrated into a CI/CD workflow to support secure cloud application deployment.

III. METHODOLOGY

The proposed system implements a Zero-Trust DevSecOps pipeline for secure cloud application deployment on Google Cloud Platform (GCP). The methodology integrates security validation and automated deployment practices throughout the software development lifecycle to ensure that applications are continuously verified before deployment.

Initially, developers develop the application and maintain the source code in a GitHub repository. GitHub serves as the centralized version control platform where all application changes are stored and managed. Whenever new code is pushed to the repository, a webhook automatically triggers the Jenkins Continuous Integration/Continuous Deployment (CI/CD) pipeline.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506067|

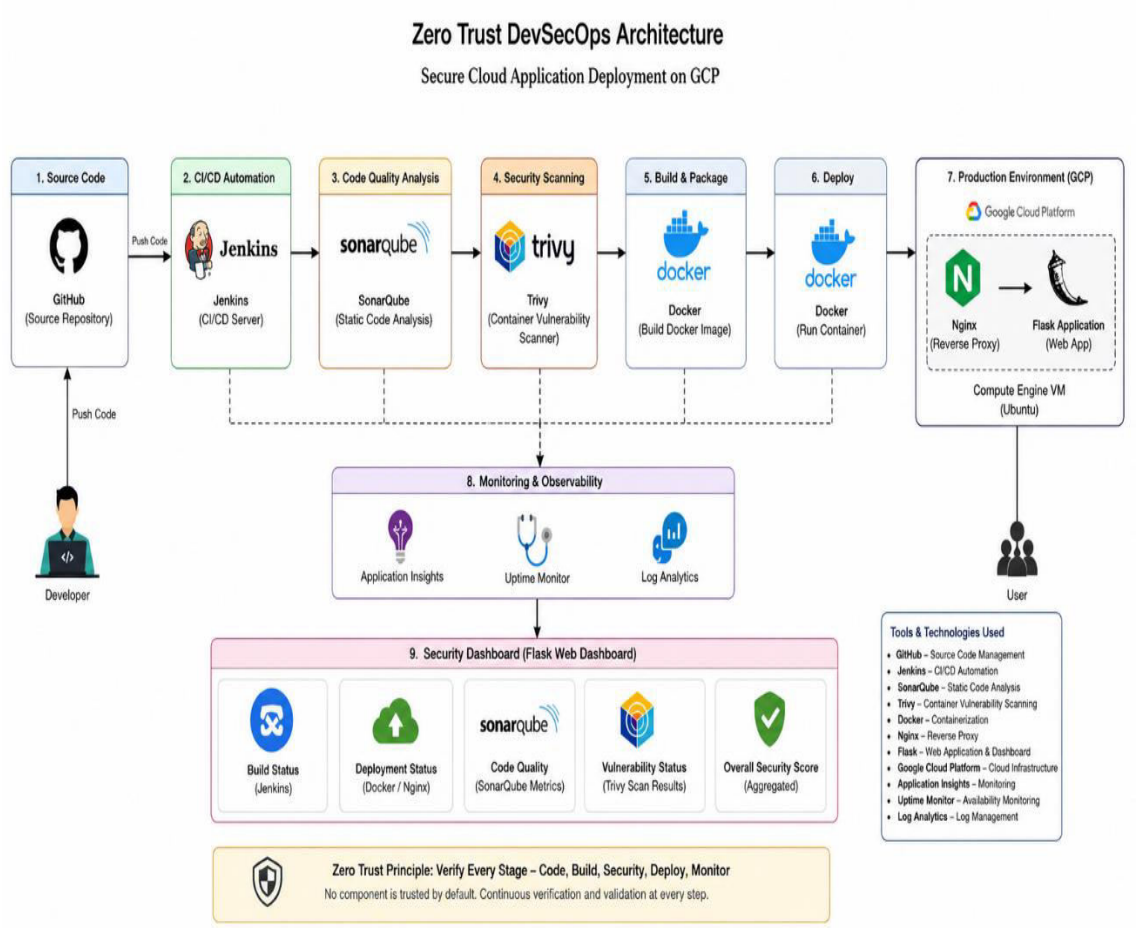


Fig.1 System Architecture

In the first stage of the pipeline, Jenkins retrieves the latest source code from GitHub and initiates automated code analysis. Static code quality and security assessment are performed using SonarQube. This stage helps identify code defects, security vulnerabilities, code smells, and maintainability issues before the application proceeds to deployment. By performing early security validation, potential risks can be detected and resolved during the development phase.

After successful code analysis, the application is containerized using Docker. Containerization ensures consistency across development, testing, and production environments by packaging the application and its dependencies into a portable container image. Once the Docker image is created, Trivy is used to perform vulnerability scanning on the container image. The scanning process identifies known Common Vulnerabilities and Exposures (CVEs), insecure packages, and outdated software components that may introduce security risks.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506067|

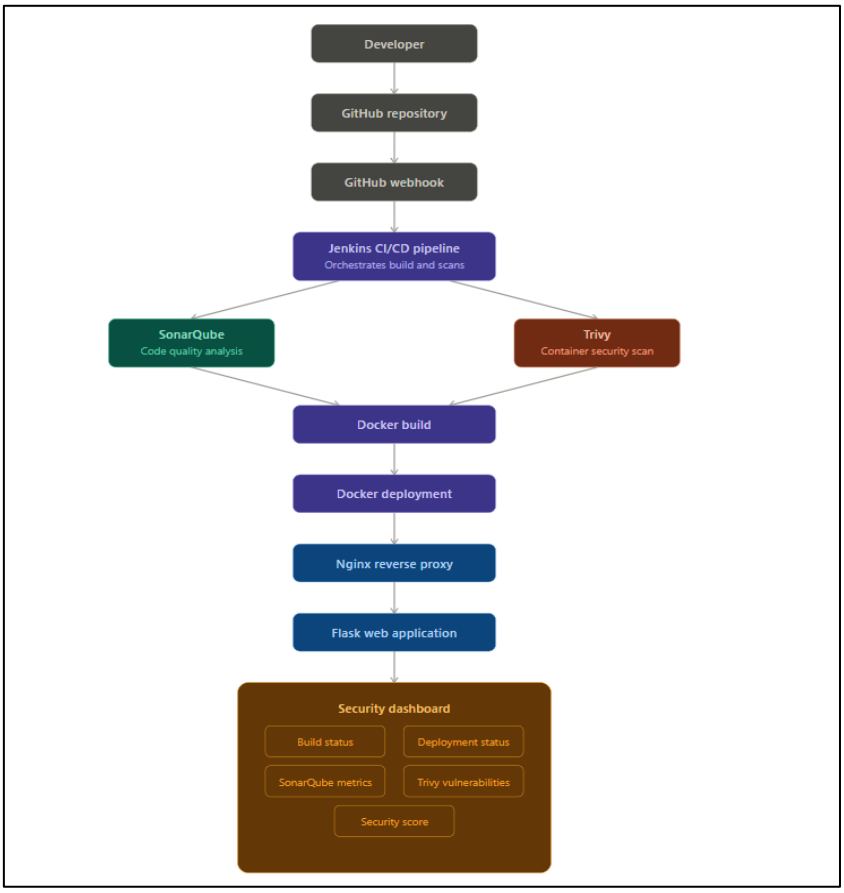


Fig. 2 Workflow Diagram

Following successful security validation, the Docker container is deployed on the cloud-hosted virtual machine. Nginx is configured as a reverse proxy server to provide secure access to the deployed application and manage incoming client requests. This layer improves accessibility and abstracts direct access to the application service.

To provide centralized visibility into the deployment process, a custom Security Dashboard is developed using Flask. The dashboard presents important security and deployment information, including build status, deployment status, SonarQube analysis results, Trivy vulnerability statistics, container information, and overall security metrics. This dashboard acts as a monitoring interface that enables users to review the security posture of the deployed application.

The implemented workflow follows Zero-Trust principles by enforcing continuous verification at multiple stages of the deployment pipeline. Source code quality is validated through SonarQube, container security is verified using Trivy, and deployment activities are continuously monitored through the Security Dashboard. By integrating security controls directly into the CI/CD pipeline, the proposed methodology establishes a secure, automated, and scalable approach for cloud application deployment.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506067|

IV. EXPERIMENTAL RESULTS

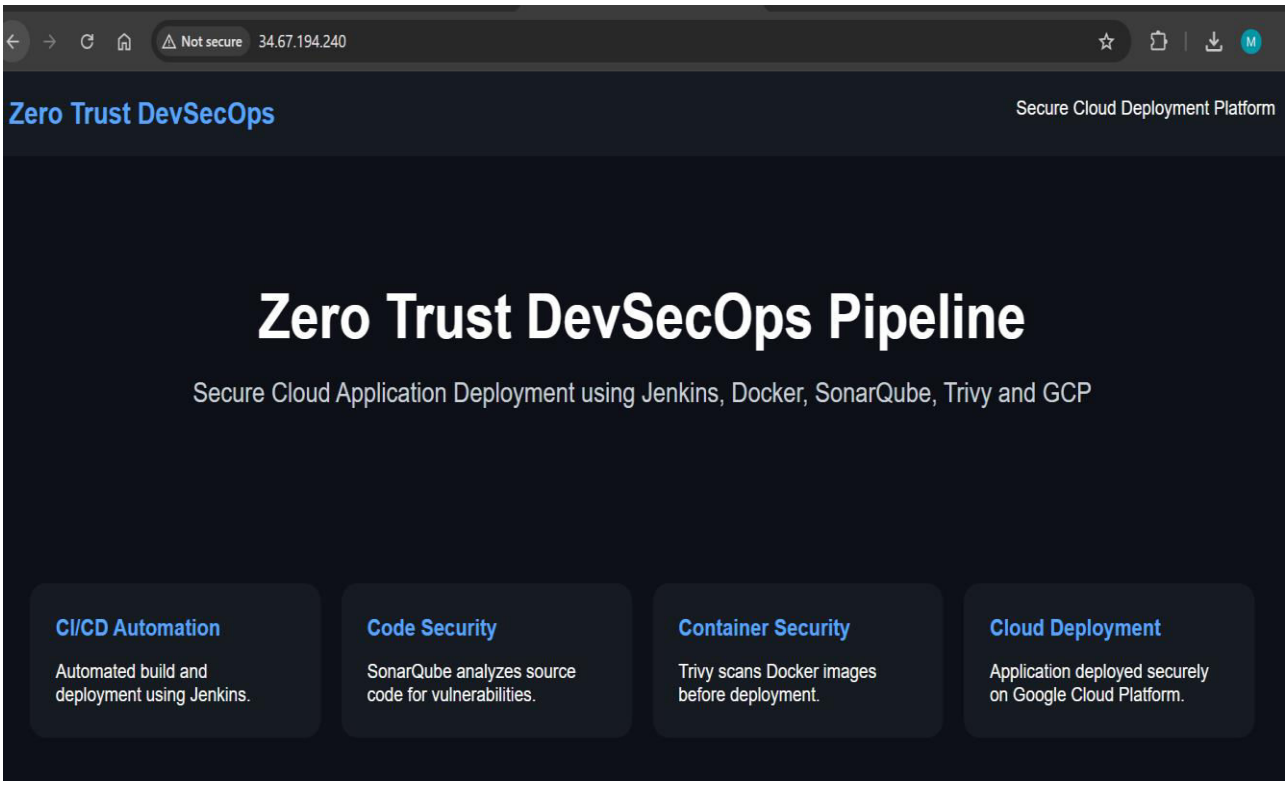
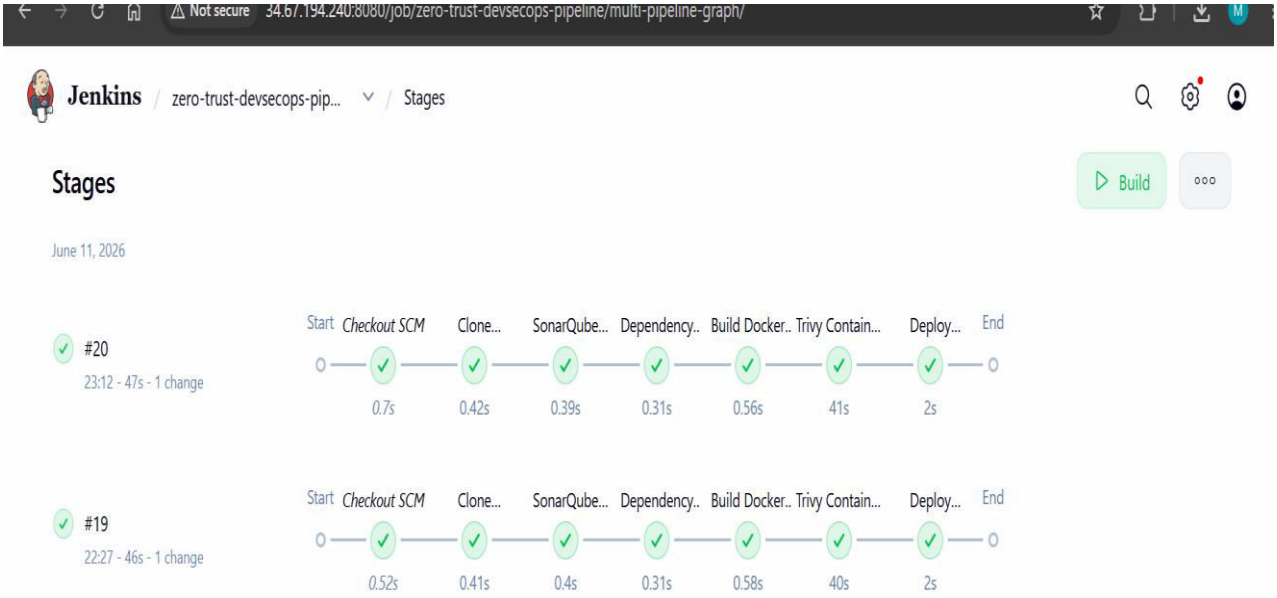


Fig.3 Deployed Application



International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506067|

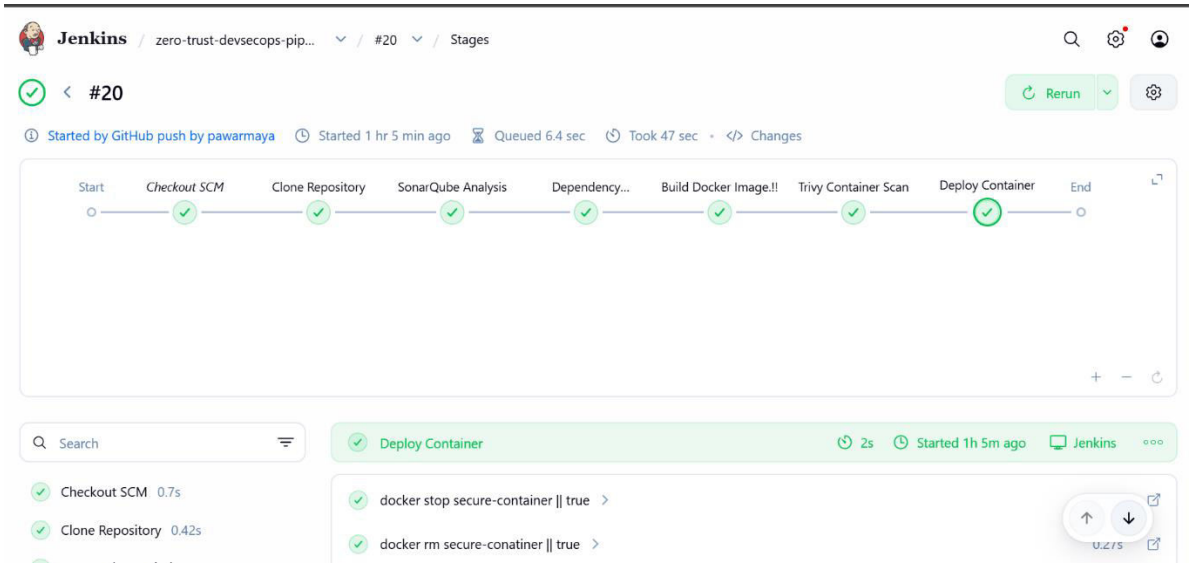


Fig .4 Jenkins Pipeline

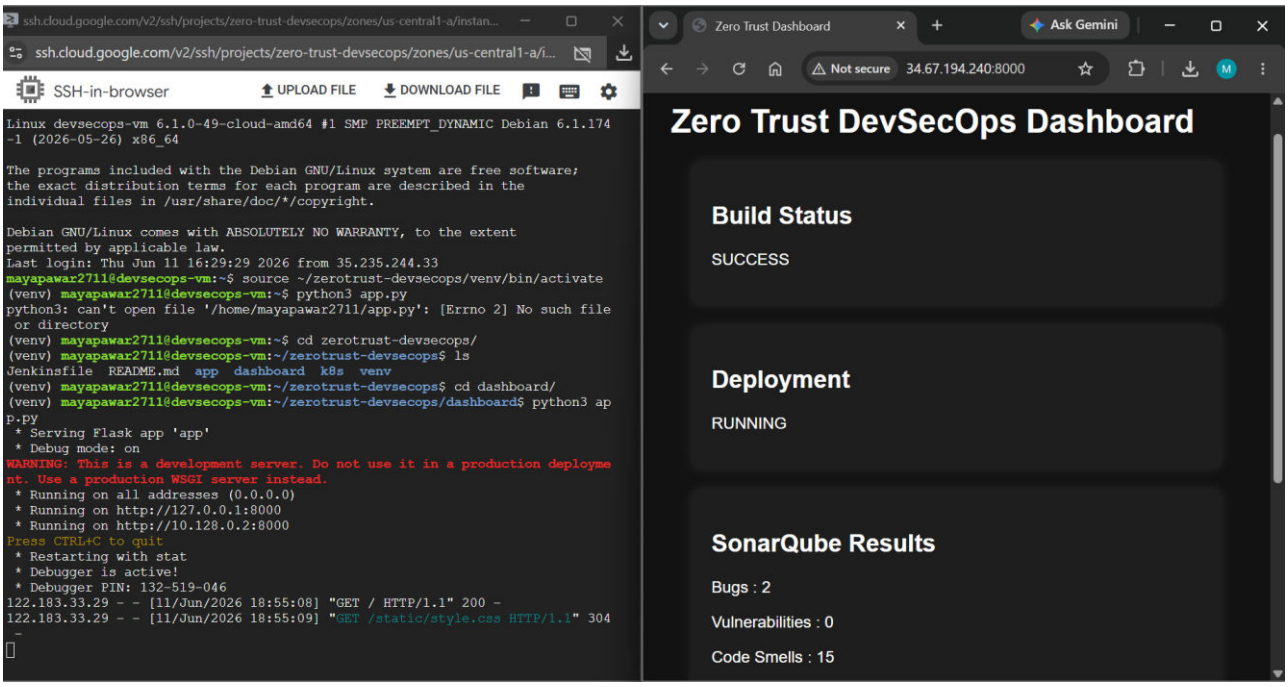


Fig.5 Dashboard

V. CONCLUSION

This project presented the design and implementation of a Zero-Trust DevSecOps Pipeline for secure cloud application deployment on Google Cloud Platform (GCP). The proposed solution integrates security mechanisms directly into the CI/CD workflow by combining GitHub for source code management, Jenkins for automation, SonarQube for static code analysis, Trivy for container vulnerability assessment, Docker for application containerization, Nginx for secure application access, and a custom Security Dashboard for centralized monitoring.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506067|

The implemented pipeline demonstrates how security validation can be incorporated throughout the software delivery lifecycle to identify and mitigate potential risks before deployment. By applying Zero-Trust principles such as continuous verification, security assessment, and automated validation, the system enhances the security and reliability of cloud-native application deployments.

The proposed approach provides an automated, scalable, and security-focused deployment framework that reduces manual intervention while improving visibility into application security and deployment status. The integration of security controls within the CI/CD pipeline helps organizations strengthen their cloud security posture and support secure software delivery practices.

Future enhancements may include Kubernetes-based orchestration, real-time security monitoring using Prometheus and Grafana, policy enforcement using Open Policy Agent (OPA), automated compliance validation, and multi-cloud deployment support to further enhance the capabilities of the proposed framework.

REFERENCES

[1] X. Feng and S. Hu, "Cyber-Physical Zero Trust Architecture for Industrial Cyber-Physical Systems," in IEEE Transactions on Industrial Cyber-Physical Systems, vol. 1, pp. 394-405, 2023, doi: 10.1109/TICPS.2023.3333850.
[2] Sanders, G., Morrow, T., Richmond, N., Woody, C., “Integrating Zero Trust and DevSecOps.” 2021. SEI
[3] Patel, D., “Zero Trust and DevSecOps in Cloud-Native Environments with Security Frameworks and Best Practices.” International Journal of Advanced Research in Science Communication and Technology (IJARSCT), Vol 3, Issue 3, Jan 2023. ResearchGate+1
[4] Karanam, Ravindra, “Zero Trust Architecture in DevSecOps: Enhancing Security in Cloud-Native Environments.” IJRASET, Aug 2024. IJRASET
[5] Shin, D., Kim, J., Pawana, I. W. A. J., You, I., “Enhancing cloud-native DevSecOps: A Zero Trust approach for the financial sector.” Computers & Security, 2025. ACM Digital Library+1
[6] Castro, Harold, “Building a Zero Trust Security Framework with DevSecOps in AWS.” Dec 2024. ResearchGate.
[7] https://www.bacancytechnology.com/blog/zero-trust-security-in-devops.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details