DevSecOps Platform for Kubernetes: eBPF Enforcement, Zero Trust, GitOps, and Quality Gates

Vasile Mihai Glodici · Graduation PoC · Bureau Veritas Cybersecurity · Fontys ICT 2026

Mentor: Stijn van Es · 1st Assessor: Xuemei Pu · 2nd Assessor: Michiel Koehorst

open-source security stack · CNCF-native · no vendor lock-in

MAIN RESEARCH QUESTION

How can eBPF-based security, GitOps automation, zero-trust identity management, and automated quality control be combined into a DevSecOps platform on AWS EKS that is secure, observable, and scales with the organisation?

SQ1 · CI/CD Quality Gates: Blocking critical vulnerabilities while managing policy exceptions.

SQ2 · eBPF Security and Zero-Trust: Combining kernel-level enforcement with identity-aware access control.

SQ3 · GitOps and Observability: Reproducible infrastructure with drift correction and integrated monitoring.

SQ4 · AI-Assisted Development (nice to have): Using AI tooling to improve development quality and troubleshooting.

Context

- Bureau Veritas is migrating from VMs to Kubernetes; containers need **purpose-built security**, not retrofitted perimeter tools.
- Three gaps drive this research:
  - **No east-west visibility:** perimeter firewalls cannot see traffic inside a cluster.
  - **No automated quality control:** vulnerabilities reach production, and AI-assisted code accelerates the risk.
  - **Manual policy drift:** hand-edited configuration drifts and is unreproducible.

Approach

- **Defense in depth:** Keycloak and Envoy centralize access; Cilium enforces service-to-service policy; Tetragon blocks runtime abuse; GitLab CI gates every deploy.
- **GitOps:** all cluster configuration is declared in Git and reconciled continuously by ArgoCD.
- **Workload:** Hipster Shop, an **11-service** microservice webshop, restricted to required service-to-service communication.

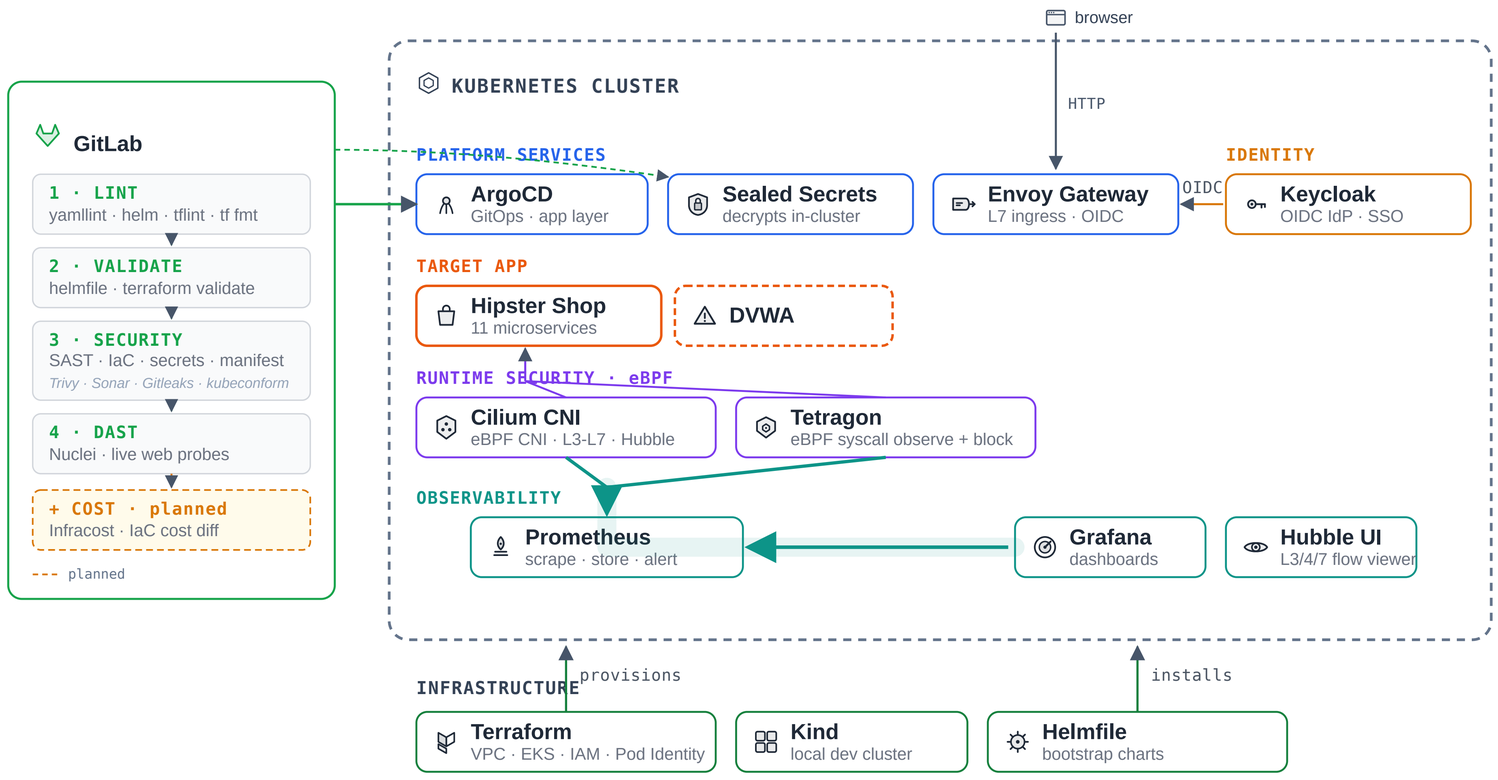
Results

Runtime enforcement
Attack process execution blocked at launch

- **6 Tetragon policies** blocked MITRE ATT&CK-aligned process attacks on both Kind and Amazon EKS.
- **Quality gates:** Trivy and SonarQube cover static analysis; Gitleaks catches secrets; kubeconform validates manifests; Nuclei adds DAST evidence.
- **Cloud proof:** the same Helm chart ran on Kind and **Amazon EKS**, with the pipeline green and GitOps in sync.

Platform Architecture

GitLab CI → ArgoCD → Kubernetes; Cilium + Tetragon enforce in-kernel; Prometheus, Grafana, Hubble observe.



1 → 2
one Helm chart, proven on two clusters: laptop and Amazon EKS

Implemented Stack

KERNEL & NETWORK: Cilium · Tetragon · eBPF

GITOPS & IAC: ArgoCD · Helm · Terraform · GitLab

OBSERVABILITY: Prometheus · Grafana

IDENTITY & EDGE: Keycloak · Envoy

CI SECURITY: Trivy · SonarQube

PLATFORM: Kubernetes · Docker · Linux · AWS · CNCF

Evidence

Live east-west flows. *Traffic between the webshop services (Kind capture).*

Grafana on Amazon EKS. *Node memory, disk, CPU and network trends from the Amazon EKS run.*

STANDARDS & THREAT MODEL
6 MITRE ATT&CK tactics (incl. Execution, Privilege Escalation, Exfiltration), grounded in NIST guidance on containers, CI/CD, and zero-trust.

METHOD
DOT Framework (Library, Field, Workshop, Lab, Showroom) and the Newman Design Challenge.

PORTFOLIO
Scan for the portfolio, evidence pack, and the Kind + Amazon EKS walkthrough video.