Designing a DevSecOps Platform with eBPF, Zero Trust, and GitOps

Vasile Mihai Glodici · Graduation PoC

Bureau Veritas Cybersecurity · Fontys ICT 2026

Mentor: Stijn van Es

WHY

Bureau Veritas is migrating to Kubernetes. That demands a new security approach.

Internal tooling runs on VMs with traditional hypervisors. Product Development and AI teams are planning a Kubernetes migration, and containers need purpose-built security, not retrofitted perimeter tools.

1	**No internal visibility** Perimeter firewalls can't see east-west traffic inside a Kubernetes cluster or monitor container-level interactions.
2	**Automated quality control** Without CI/CD quality gates, vulnerabilities reach production undetected. AI-assisted code generation accelerates this risk.
3	**Manual policy drift** Manual configuration causes drift, human error, and unreproducible environments. No unified workflow for runtime monitoring and automated scanning.

WHAT

A scalable, open-source DevSecOps platform built on CNCF projects.

NEWMAN DESIGN CHALLENGE

Design	a DevSecOps platform combining IaC, GitOps, eBPF security, and CI/CD scanning
that enables	DevOps, development, and security teams
in	a cloud-native Kubernetes environment
to	securely build, scan, deploy, enforce, and observe containerized applications
with	CI quality gates, kernel-level threat blocking, and zero-trust identity.

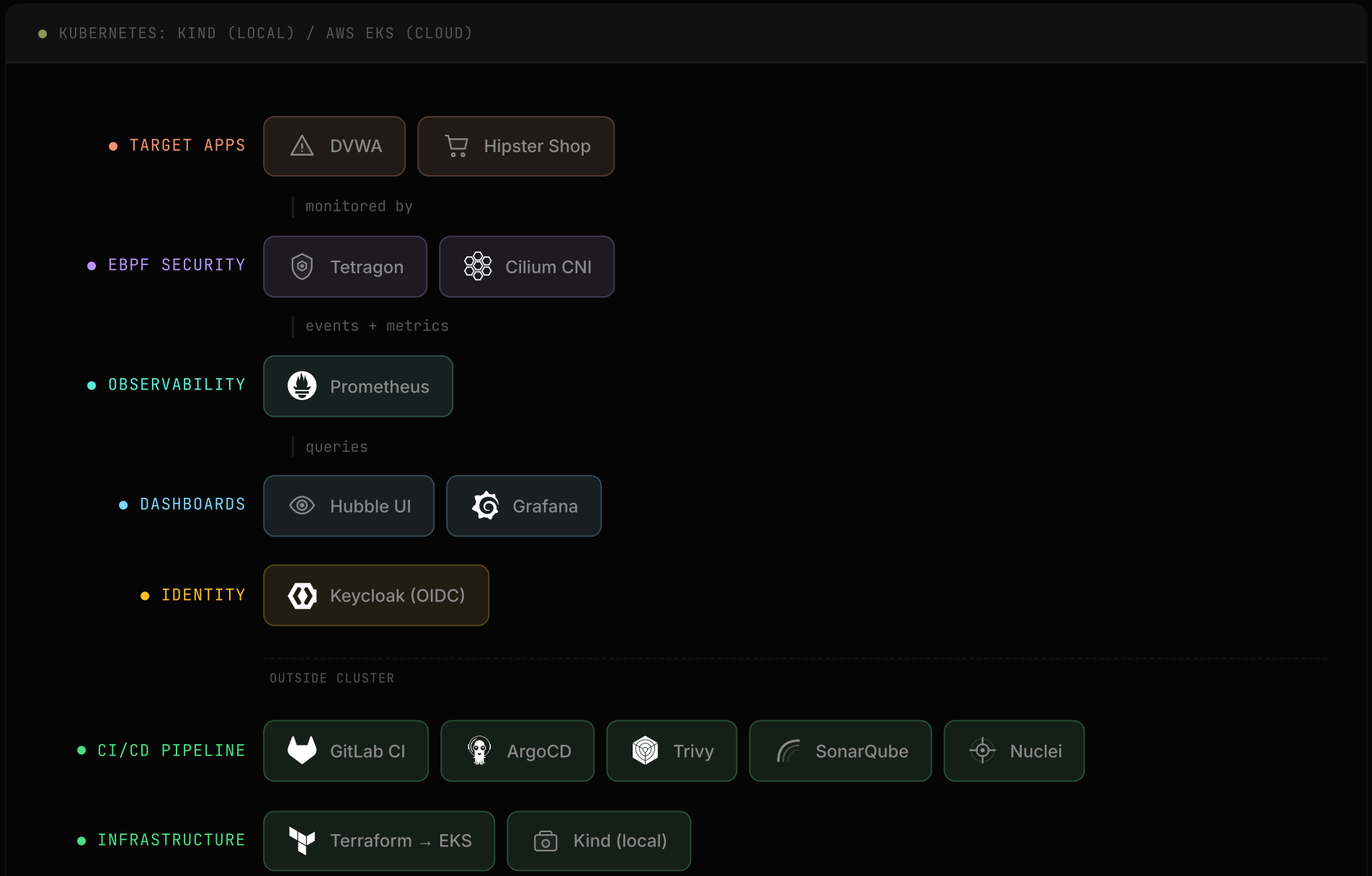
MAIN RESEARCH QUESTION

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?

HOW

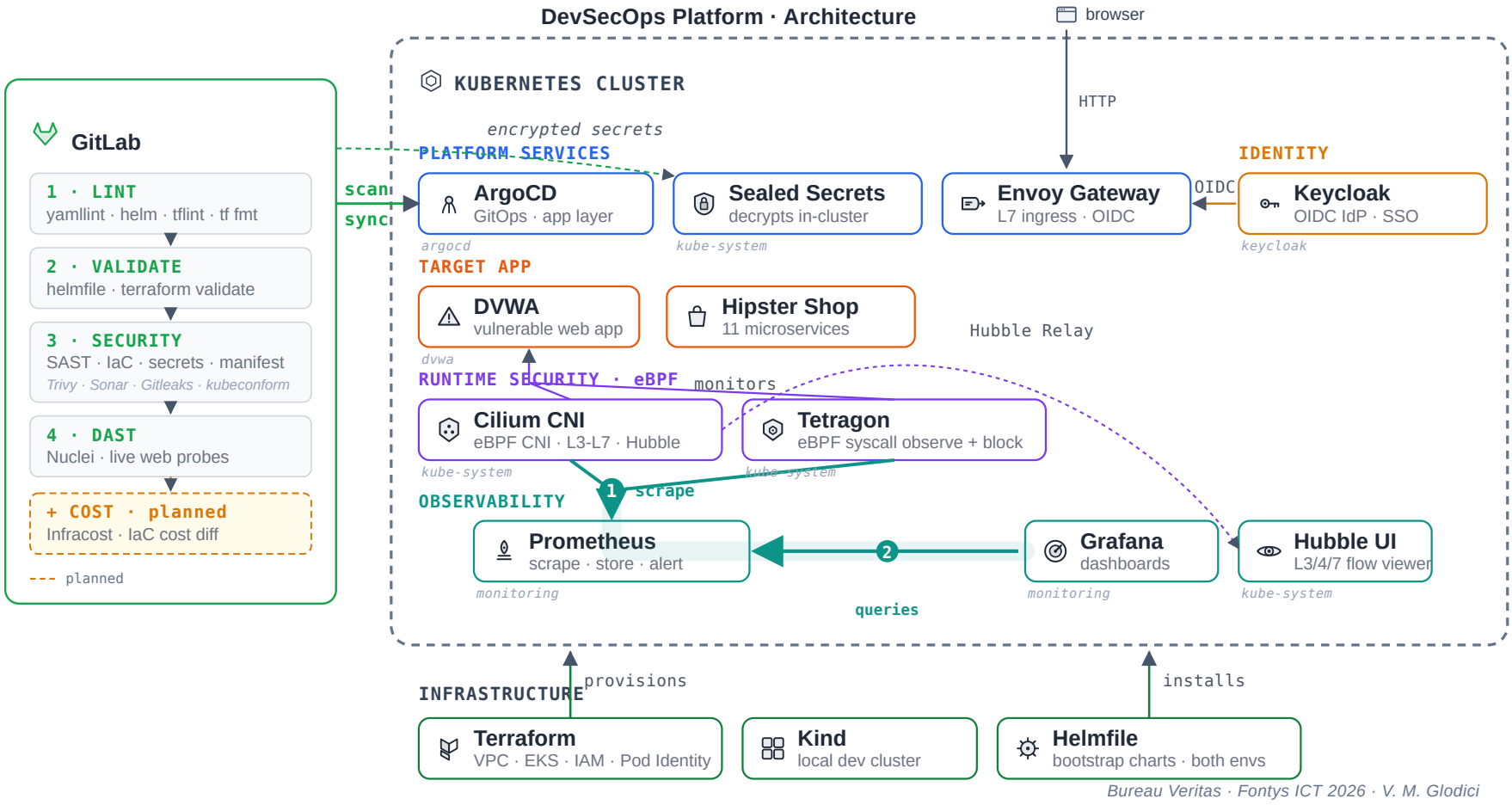
Platform architecture

Four pillars: maintainability, automated CI/CD, kernel-level security, and full observability.



HOW

Architecture detail



RESEARCH

Sub-questions

SQ1	**CI/CD Quality Gates** Blocking critical vulnerabilities while managing policy exceptions Library Workshop Lab Field
SQ2	**eBPF Security & Zero-Trust** Combining kernel-level enforcement with identity-aware access control Library Workshop Lab
SQ3	**GitOps & Observability** Reproducible infrastructure with drift correction and integrated monitoring Library Workshop Field
SQ4	**AI-Assisted Development** NICE TO HAVE Using AI tooling to improve development quality and troubleshooting Library Workshop Field

STATUS

Project progress

237+	6	9	9
COMMITS	TETRAGON POLICIES	K6 HTTP SCENARIOS	RUNTIME TESTS

PHASE 1: COMPLETE

CI/CD Security Pipeline

- yamllint, Helm lint, kubeconform validation
- Trivy full-stack scanning (images + IaC)
- Nuclei authenticated DAST with custom templates

Runtime Security & Networking

- Cilium eBPF CNI replacing kube-proxy
- 6 Tetragon TracingPolicies for syscall/file monitoring
- Workload hardening + Sealed Secrets

Observability

- Prometheus + Grafana metrics stack
- Cilium, Hubble, and Tetragon scraped via the ServiceMonitors each chart ships
- Tetragon Security Events dashboard with alert rules

GitOps & Infrastructure

- ArgoCD managing clusters from GitLab
- Keycloak SSO + Grafana OIDC integration
- k6 load test suite (9 scenarios)

SonarQube Integration

- CI pipeline SAST quality gate
- Code quality and security vulnerability detection

PHASE 2: COMPLETE

Microservice Target App

DVWA swapped for Hipster Shop: 11 digest-pinned microservices plus redis, with a service-graph CiliumNetworkPolicy set (L7 gRPC method matching) and Tetragon policies scoped to the namespace.

Terraform Cloud Migration

AWS EKS Terraform modules (vpc, eks, cilium, addons) applied to staging with read-only CI gates (fmt, validate, tflint, trivy-config). Cluster up, ArgoCD reconciling, dashboards green.

NOW & BEYOND

IN PROGRESS

Restricted-cloud bring-up

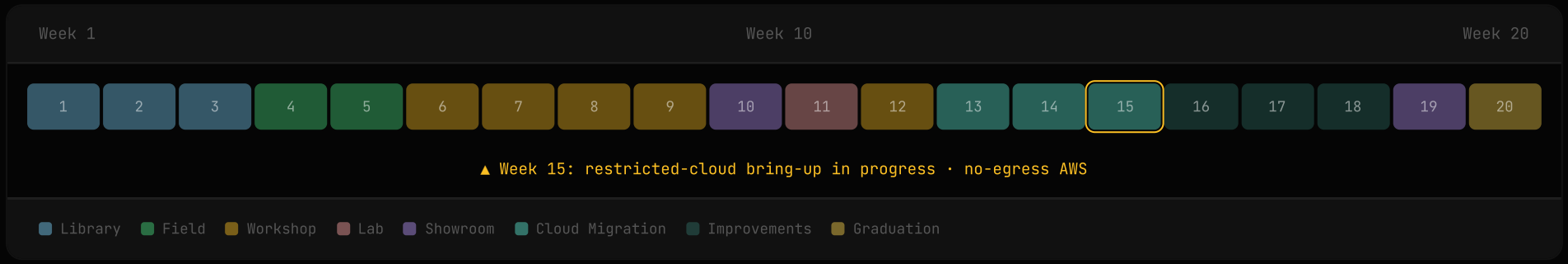
Moving the platform into the locked-down AWS environment with no internet egress. Images and Helm charts pulled through the GitLab Dependency Proxy and Container Registry mirror.

TrojanCTF: 3rd place BONUS

- Beginner bracket, 2-person team against teams of 4
- Finished 1 challenge short of 1st place
- Blue-team in the lab, offensive security at the CTF

TIMELINE

20-week plan & methodology



- **Li** Library /literature & docs review
- **Fi** Field /stakeholder interviews
- **W** Workshop /hands-on build
- **La** Lab /security & load testing
- **Sh** Showroom /live demo & poster