THE LINUX FOUNDATION

OPEN SOURCE SUMMIT INDIA

Secure by Default
Building an AI-Augmented, OSS-Powered Reusable CI/CD Pipeline

Jenisten Xavier
Full Creative

Jenisten Xavier
Sr IT Analyst
Full Creative
DevOps, Cloud & AI

GDG Cloud Chennai Organiser

Challenges

- Projects are unique
- Different pipelines
- Tool sprawl
  Example - Secret scans, SAST, SBOM
- Deadlines vs Shift left on Security
- Manual fix for CVEs, version bumps, lints, tests.

The Supply-Chain Story, End to End

- Each link answers one supply-chain question:
- GitLeaks - did we leak a credential?
- OSV Scanner - did we add a known-vulnerable dependency?
- SonarQube - is the code itself sound?
- CycloneDX + Dependency-Track - what exactly did we ship, now and forever?
- Trivy - is the image clean?

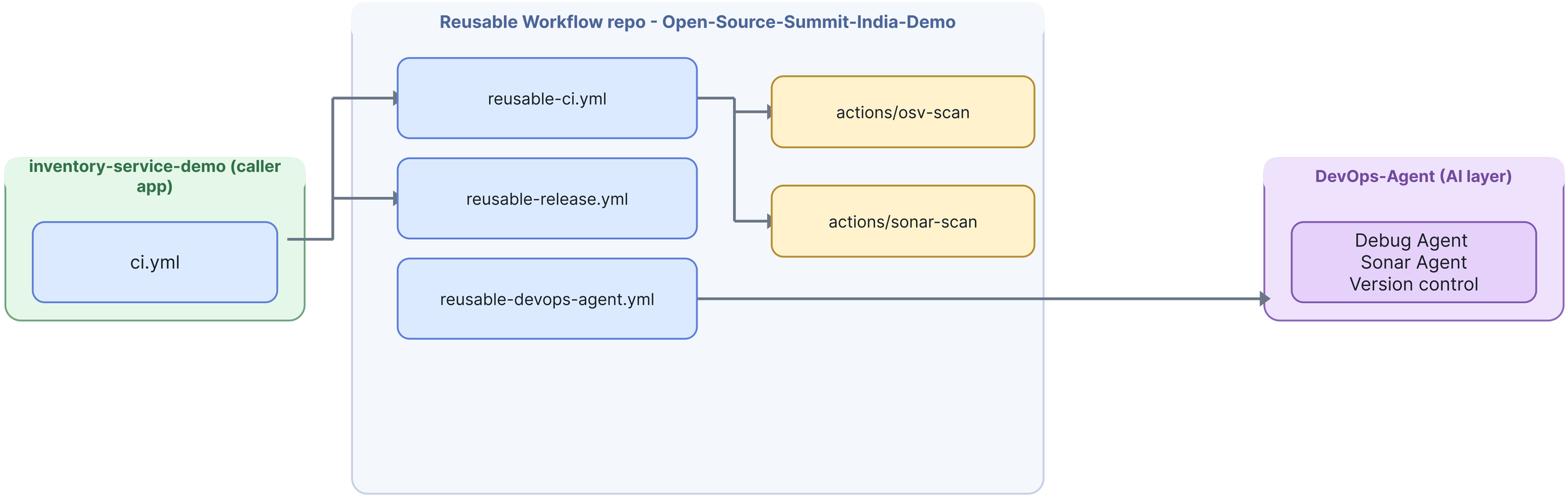
Solution - Secure-by-Default CI/CD

- Reusable CI/CD pipeline
- Powered by open-source Tools
- Autonomous AI agents (DevOps Agents)

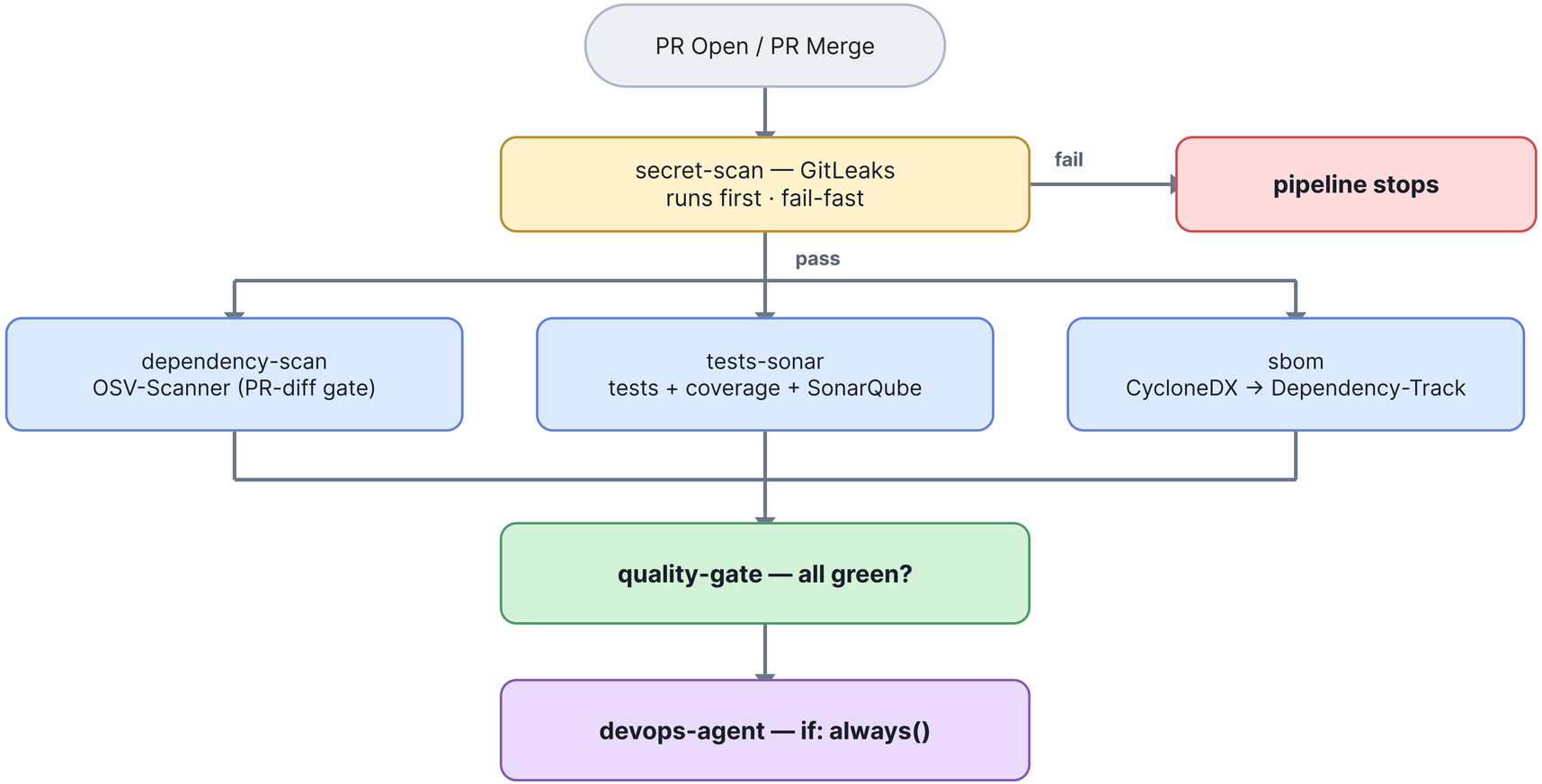
CI/CD is no longer an automation layer

- It's the platform.

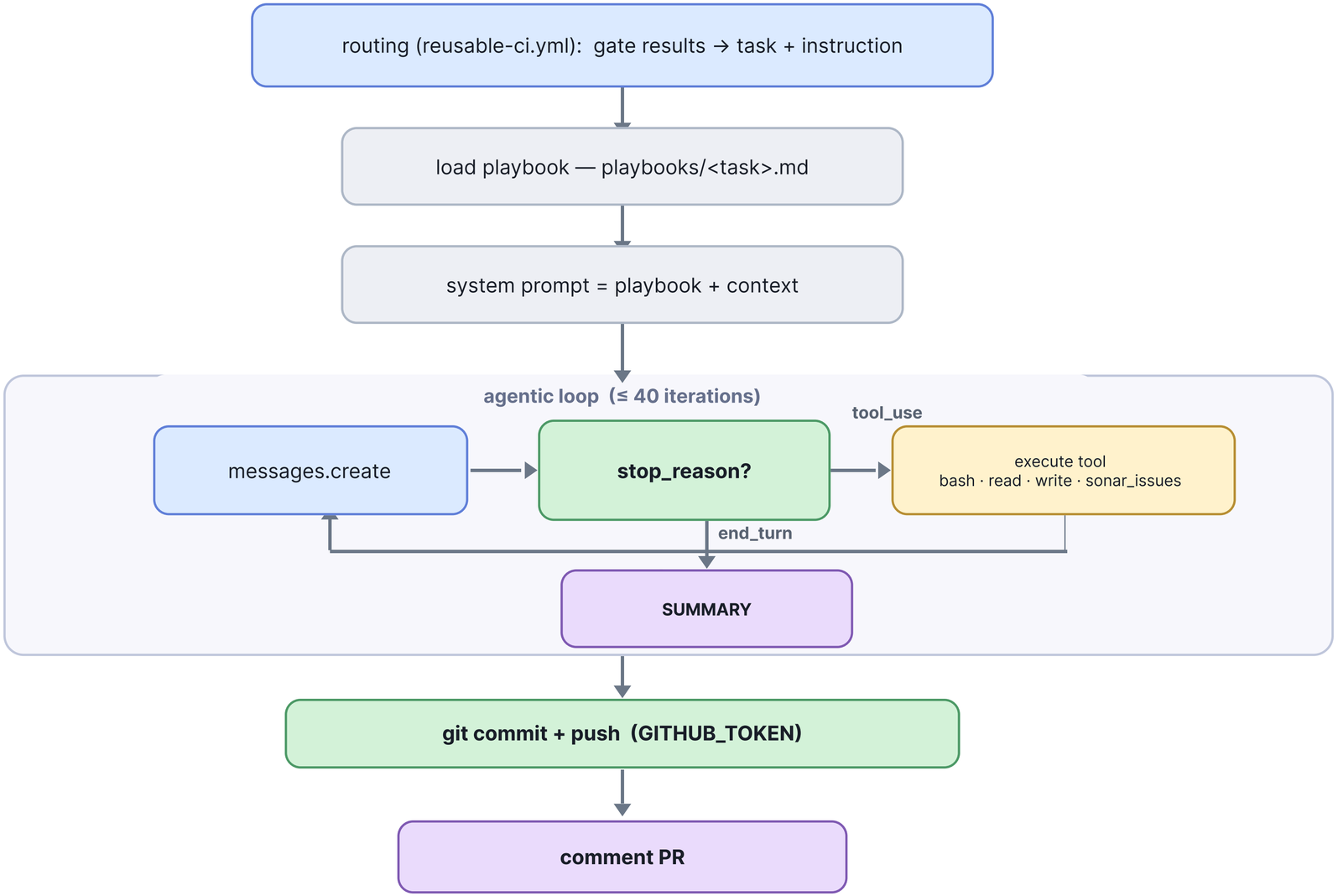
System Overview — Three Repositories



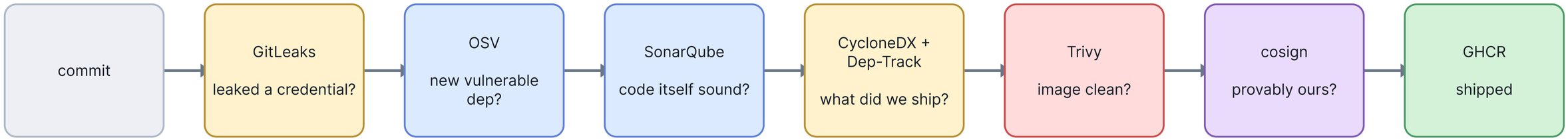
Execution Flow



Agent Architecture — The Agentic Loop



Execution Flow



The Demo App- inventory-service-demo

- A deliberately minimal Express service
  - the point is the pipeline, not the app.

- Stack:
  Node 20 · Express 4 · Jest + supertest · Dockerfile · semantic-release

- GitHub Actions
  - Caller Workflow

GitLeaks — Secret Scanning

- Scans the repo and full git history (fetch-depth: 0) for secrets
  - API keys, tokens, private keys (regex + entropy).

- gitleaks/gitleaks-action@v3. Runs first. Any finding → hard fail.

Pros
- fast, zero-config defaults; scans history, not just the diff.

Cons
- false positives on fixtures (needs a .gitleaks.toml allowlist);

Approach
- a pushed secret is already compromised — rotate, don't just delete.

URL - https://github.com/gitleaks/gitleaks

OSV-Scanner — Dependency CVEs

- Google's scanner against the OSV.dev open database.
- Reads lockfiles, matches known vulnerabilities.

URL - https://github.com/google/osv-scanner

SonarQube CE — SAST + Coverage

- Self-hosted Community Edition.
- Static analysis for bugs, smells, vulnerabilities + ingests test coverage.
- Quality gate turns findings into pass/fail.
- Pros: mature multi-language SAST, free CE tier; findings feed the AI self-heal.
- Cons: self-hosted; CE lacks branch/PR analysis; gate tuning needed.

CycloneDX — Software Bill of Materials

- cdxgen generates a CycloneDX SBOM - a machine-readable inventory of every dependency, transitive included.

- npx -y @cyclonedx/cdxgen@latest -o bom.json. Uploaded as a build artifact.

- Why
  - when the next Log4Shell drops, you answer "are we affected?" in seconds, not days.

OWASP Dependency-Track — SBOM Home

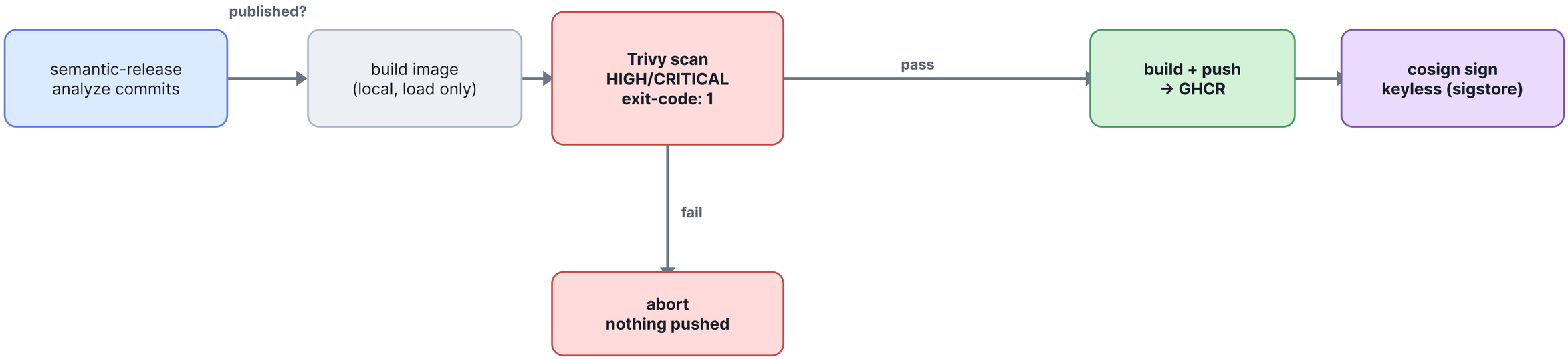
- The platform the SBOM is sent to.
  Continuously monitors every uploaded BOM against new vulnerability intel.
- DependencyTrack/gh-upload-sbom@v3.
- OSV gates the PR at a point in time;
  Dependency-Track watches your shipped components forever - a CVE disclosed tomorrow against a dep you shipped today lights up here.
- Pros: continuous, retroactive monitoring; org-wide inventory + policy.
- Cons: another service to host + secure; needs governance to avoid alert fatigue.

URL - https://dependencytrack.org/

Release Execution Flow

- semantic-release analyzes commits → decides if there's a release and the version.
- The image is built locally only (load: true, push: false).
- Trivy scans that exact image — HIGH/CRITICAL, exit-code: 1; fail = abort.
- Only a clean image is pushed to GHCR.
- cosign signs it by digest — keyless (sigstore).
- A vulnerable image is built, scanned, and thrown away — it never reaches the registry.

Release Execution Flow



Trivy — Container Image Scanning

- Aqua's scanner for container images - OS packages + app deps. Here it's the release gate.
- severity: HIGH,CRITICAL · ignore-unfixed: true · exit-code: 1 (fail-closed).
- Scans the locally-built image before any push.
- Pros: scans the image where OSV can't reach (base OS layers); single binary, fast.
- Cons: HIGH,CRITICAL only — tune for your risk; base-image CVEs are upstream's — pick slim/distroless bases.

Recap - What we Just Saw

- A secure-by-default CI/CD Pipeline
- One **uses:** line inherits the whole chain.
- Every gate required, fail-fast + fail-closed.
- 100% open-source:
  GitLeaks · OSV Scanner · SonarQube · Dependency-Track - SBOM · Trivy.
- An AI layer to handle:
  - Sonar issues
  - Debug failed test cases.
  - Automated Version Management.
- The takeaway: Use a secure pipeline.

Questions ?

References

URLs

1. https://github.com/jenistenxavier/DevOps-Agent
2. https://github.com/jenistenxavier/Open-Source-Summit-India-Demo
3. https://github.com/jenistenxavier/inventory-service-demo

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT INDIA