**SAST Implementation JuiceShop**

**Stage 1: gitleaks\_stage – Static Secrets Detection**

**Configuration Breakdown**

gitleaks\_stage:

stage: sast\_stage

* Assigns this job to the sast\_stage, grouping all static analysis jobs together.

image: docker:24

services:

- docker:24-dind

* docker:24: Uses Docker CLI version 24 to run container-based tools.
* docker:24-dind: Enables Docker-in-Docker so the CI container can run Docker images.

**Variables**

variables:

CI\_REGISTRY\_USER : beshoynfarag

CI\_REGISTRY\_PASSWORD : dckr\_pat\_H9qEvw8YgvWnAKPaAj6xTYCdUZk

DOCKER\_HOST: tcp://docker:2375

DOCKER\_TLS\_CERTDIR: ""

* **CI\_REGISTRY\_USER / PASSWORD**: Credentials to authenticate Docker CLI to pull images (not used directly by Gitleaks but needed for private registries).
* **DOCKER\_HOST / TLS**: Set up for Docker-in-Docker communication without TLS, typical in internal CI environments.

**Script Commands**

script:

- echo "$CI\_REGISTRY\_PASSWORD" | docker login -u "$CI\_REGISTRY\_USER" --password-stdin

* Logs in to Docker to ensure access to private registries or rate-limit-free pulls.

- docker pull zricethezav/gitleaks:latest

* Downloads the latest official Gitleaks container image.

- docker run --rm -v "$CI\_PROJECT\_DIR":/src -w /src zricethezav/gitleaks:latest detect --source=. --report-path=gitleaks-report.json

* Mounts the project directory to /src inside the container.
* Runs gitleaks detect against the source directory.
* Outputs results to a file: gitleaks-report.json.

**Artifacts & Rules**

artifacts:

reports:

secret\_detection: gitleaks-report.json

paths:

- gitleaks-report.json

* Makes the gitleaks-report.json available as a CI artifact and integrates it with GitLab's **Secret Detection** UI (if applicable).

allow\_failure: true

* **allow\_failure: true**: The job won’t fail the pipeline if secrets are found—useful in awareness-building phases.

**Stage 2: sast\_stage – Code Vulnerability Detection Using Semgrep**

**Configuration Breakdown**

sast\_stage:

stage: sast\_stage

* Belongs to the same sast\_stage phase as Gitleaks, keeping all static analysis tasks grouped.

image: beshoynfarag/semgrep-juiceshop:latest

* Custom image prebuilt with Semgrep and juice-shop code or ruleset. This reduces setup time during the CI job.

**Script Commands**

script:

- echo "Listing files at root:"

- ls -la

- echo "Listing files inside juice-shop-master/:"

- ls -la juice-shop-master/

* Debugging steps to verify folder structure and visibility of the scan target directory (juice-shop-master/).

- semgrep --config p/ci --json -o semgrep-results.json juice-shop-master/ || true

* Runs Semgrep using the official p/ci config from the Semgrep Registry.
* Targets juice-shop-master/, scanning for common security issues.
* Outputs results in JSON format to semgrep-results.json.
* || true ensures the job doesn't fail even if vulnerabilities are detected (useful during pipeline hardening or testing phases).

**Artifacts & Rules**

artifacts:

paths:

- semgrep-results.json

reports:

sast: semgrep-results.json

expire\_in: 2 hrs

* Stores semgrep-results.json as a CI artifact and integrates it with GitLab’s **SAST report interface**.
* Automatically expires in 2 hours to save storage.

allow\_failure: true

needs: ["gitleaks\_stage"]

* **allow\_failure: true**: Pipeline doesn’t fail if Semgrep finds issues.
* **needs**: Ensures this job runs only after gitleaks\_stage, maintaining logical dependency and ordering.