**DAST Implementation**

**CI/CD Security Stage: dast\_stage**

**Stage Name : dast\_stage**

dast\_stage:

stage: dast\_stage

* **Why**: This job is explicitly assigned to the dast\_stage phase. Grouping jobs by purpose improves readability and enables stage-specific rules, like running security checks after deployment.

**Docker Environment Setup**

image: docker:latest

services:

- docker:dind

* **Why**:
  + docker:latest: Provides the Docker CLI required to pull and run the ZAP container.
  + docker:dind (Docker-in-Docker): Required to allow this job to run Docker containers *inside* the GitLab job container. Necessary for dynamic scanning via OWASP ZAP’s Docker image.

**Docker Access Variables**

variables:

DOCKER\_HOST: tcp://docker:2375

DOCKER\_TLS\_CERTDIR: ""

DOCKERHUB\_USERNAME: beshoynfarag

DOCKERHUB\_TOKEN: dckr\_pat\_H9qEvw8YgvWnAKPaAj6xTYCdUZk

* **DOCKER\_HOST**: Directs the Docker CLI to connect to the Docker daemon over TCP (needed for Docker-in-Docker).
* **DOCKER\_TLS\_CERTDIR**: Disables TLS setup for simplicity. Suitable for controlled CI/CD environments, but not secure for production runners.
* **DOCKERHUB\_USERNAME / TOKEN**: Used to authenticate to Docker Hub and pull the private/custom image beshoynfarag/zap2docker-stable.

**Artifacts Configuration**

artifacts:

paths:

- dast-report.html

* **Why**: Saves the ZAP HTML report as a job artifact, making it available for download or display in GitLab after the pipeline finishes. Helps team members review security scan results without rerunning the job.

**Script Section – DAST Execution**

script:

- echo "$DOCKERHUB\_TOKEN" | docker login -u "$DOCKERHUB\_USERNAME" --password-stdin

* **Why**: Securely logs in to Docker Hub using credentials from GitLab CI/CD variables. The --password-stdin option avoids echoing secrets into command history or logs.

- docker pull beshoynfarag/zap2docker-stable:latest

* **Why**: Ensures the most up-to-date version of your custom or prebuilt OWASP ZAP Docker image is available for scanning.

- |

docker run -v $(pwd):/zap/wrk/:rw beshoynfarag/zap2docker-stable \

zap-baseline.py -t http://10.97.7.109:8080 -r dast-report.html || true

* **Line-by-line breakdown**:
  + docker run: Runs a container from the pulled ZAP image.
  + -v $(pwd):/zap/wrk/:rw: Mounts the current project directory into the container so the generated report can be written and preserved.
  + zap-baseline.py: ZAP's baseline script performs a lightweight passive scan without authentication or aggressive fuzzing.
  + -t http://10.97.7.109:8080: Target URL of the deployed application (running in the environment being tested).
  + -r dast-report.html: Specifies the filename for the HTML report.
  + || true: Ensures that the job does not fail the pipeline even if ZAP finds security issues—useful in early testing or CI pipelines not yet enforcing fail-on-vulnerability policies.