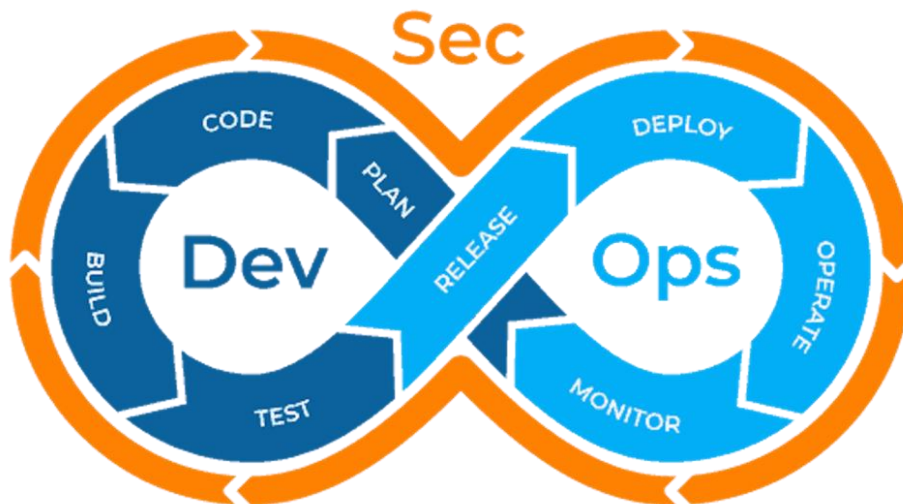


# Compte Rendu du TP1

**Sécurisation CI/CD et Analyse de Conteneurs avec GitLab CI,  
Trivy et Docker Bench**



Encadrée par :

Pr. AHMED AMAMOU

Réalisée par :

BOUTAYEB Wissal

## Introduction et Objectif :

Ce TP a pour objectif de mettre en place un pipeline CI/CD sécurisé avec GitLab CI, intégrant des outils d'analyse de sécurité tels que **Snyk, GitLeaks, OWASP ZAP, Trivy et Docker Bench**. L'objectif est d'appliquer les principes du **DevSecOps** en intégrant la sécurité à chaque étape du pipeline.

- Comprendre les composants d'un pipeline CI/CD et leur fonctionnement dans GitLab.
- Créer et configurer des **jobs** CI/CD avec intégration de la sécurité.
- Utilisation des outils d'analyse de sécurité : **Snyk, GitLeaks, OWASP ZAP, Trivy,**

### **Docker Bench.**

- Identification des vulnérabilités dans le code, les dépendances, les secrets, les conteneurs et les configurations.
- Application des principes de **DevSecOps** à toutes les couches du pipeline.

## 1. Compréhension des Composants d'un Pipeline CI/CD :

### 1.1 Pipeline CI/CD dans GitLab :

Un **pipeline** dans GitLab est une suite automatisée de **jobs** (tâches) organisés en **stages** (étapes). Il permet :

- La compilation du code.
- L'exécution de tests.
- Le déploiement de l'application.
- La détection de vulnérabilités.

### 1.2 Jobs et Stages


**Job** : Tâche unique exécutée dans le pipeline (ex: compilation, test) Il est défini dans un fichier nommé **'gitlab-ci.yml'**


**Stage** : Groupe de jobs qui s'exécutent séquentiellement (ex: **build, test, security**).










## 2. Configuration du Pipeline CI/CD :

### 2.1 Structure de notre projet : Après avoir pushé/commité notre application web sur GitLab

Wissal BOUTAYEB / WebAppVulnerabilityAnalysis / Repository

 Edit .gitlab-ci.yml  
wissal Boutayeb authored in 1 hour

8a88f1ce  History

Name	Last commit	Last update
node_modules	Initial commit	2 months ago
private/protected	Initial commit	2 months ago
public	Initial commit	2 months ago
 .gitlab-ci.yml	Edit .gitlab-ci.yml	20 hours ago
 Dockerfile	Add new file	10 hours ago
 Rapport_Vulnerabilite_WEB.pdf	Add files via upload	1 month ago
 ReadME.md	Update ReadME.md	2 weeks ago
 app.js	Initial commit	2 months ago
 database.json	Initial commit	2 months ago
 package-lock.json	Initial commit	2 months ago
 package.json	Initial commit	2 months ago
 requirements.txt	Add new file	1 day ago

Ce Project est une application web qui contient :

Des fichiers de configuration :

- **.gitlab-ci.yml** : Fichier de configuration pour GitLab CI/CD, utilisé pour automatiser les tests, builds et déploiements.
- **Dockerfile** : Fichier utilise pour créer une image Docker, permettant de containeriser l'application.
- **requirements.txt** : fichier contient Liste des dépendances nécessaires pour notre projet
- **app.js** : Fichier principal de notre application développé en utilisant NodeJS (
- **package.json** et **package-lock.json** : Fichiers de configuration pour Node.js, listant les dépendances et leurs versions.
- **database.json** : Fichier de configuration pour la base de données.
- **node\_modules/** : Dossier contenant les dépendances Node.js

- **public/** : Dossier pour les fichiers accessibles publiquement (comme HTML, CSS, JS frontend).
- **private/protected/** : Dossier pour les fichiers sensibles ou protégés (accès restreint).

## **2.2 Initialisation du Pipeline :**

Dans ce fichier **.gitlab-ci.yml** on définit les étapes suivantes :

**Stages(Test, Build, Security)**

## **3. Intégration des Outils de Sécurité :**

### **3.1 Scan des Dépendances avec Snyk (SCA) :**

Notre Objectif dans cette étape est de Détecter les vulnérabilités dans les dépendances.

Pour cela nous devons accéder au site [snyk.io](https://snyk.io) afin de générer et récupérer le token et l'ajouter dans Settings → CI/CD → Variables de Gitlab

### **3.2 Détection de Secrets avec GitLeaks :**

Notre Objectif dans cette étape est d'Identifier les clés API, mots de passe exposés.

### **3.3 Analyse Dynamique avec OWASP ZAP (DAST) :**

Notre Objectif est de Scanner l'application déployée.



### **3.4 Construction et Scan d'Image Docker avec Trivy :**

Notre Objectif est d'Analyser les vulnérabilités dans l'image Docker. Trivy identifie les CVEs dans les couches système et bibliothèques embarquées.




### **3.5 Analyse de Configuration Docker avec Docker Bench**

Notre Objectif est de Vérifier la conformité des configurations Docker.



Le Fichier **.gitlab-ci.yml** complète :

 .gitlab-ci.yml  2.54 KIB

Edit Replace Delete



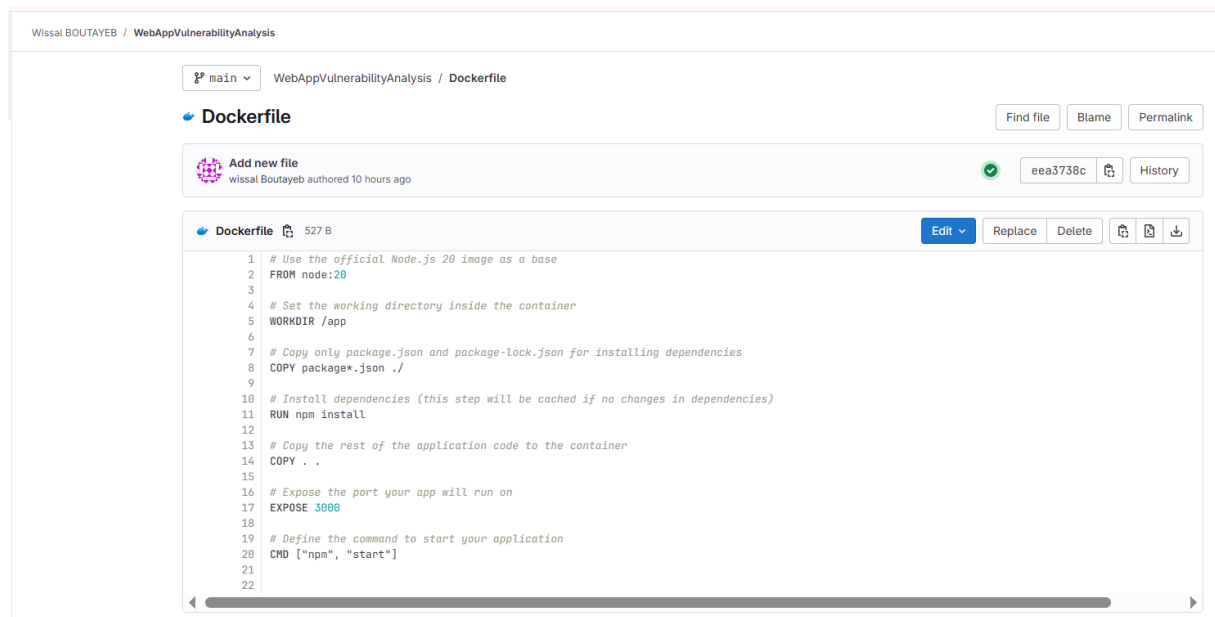
```
1 stages:
2   - test
3   - build
4   - security
5
6 # Job to build the application image
7 build_docker:
8   stage: build
9   image: docker:latest # Using the Docker official image
10  services:
11    - docker:dind
12  variables:
13    DOCKER_TLS_CERTDIR: "/certs"
14  script:
15    - docker build -t monapp:latest . # Build your app image
16    - mkdir -p ./artifacts
17    - docker save monapp:latest > ./artifacts/monapp-image.tar # Save the image as an artifact
18  artifacts:
19    paths:
20      - ./artifacts/monapp-image.tar
21    expire_in: 1 day
22  only:
23    - main
24
25 # Add a simple test stage
26 test_app:
27   stage: test
28   image: node:20 # Using a Node.js image for tests
29   script:
30     - echo "Running application tests..."
31     - echo "Tests passed!"
32  only:
33    - main
```

 Your [changes](#) have been committed successfully. 

```
35 # Job to scan the Docker image using Trivy
36 trivy_scan:
37   stage: security
38   image: alpine:latest # Using Alpine for the Trivy scan
39   script:
40     - apk add --no-cache wget
41     - wget -qO - https://github.com/aquasecurity/trivy/releases/download/v0.41.0/trivy_0.41.0_Linux-64bit.tar.gz | tar -zxvf -
42     - mkdir -p ./reports
43     - ./trivy image --no-progress --format json --output ./reports/trivy-node-report.json node:16 # Scan the Node.js image
44  artifacts:
45    paths:
46      - ./reports/
47    expire_in: 1 week
48    allow_failure: true
49
50 # Job to run Docker Bench for Security
51 docker_bench_security:
52   stage: security
53   image: docker:latest # Correcting this to use the official Docker image
54   services:
55     - docker:dind
56   variables:
57     DOCKER_TLS_CERTDIR: "/certs"
58   script:
59     - mkdir -p ./reports
60     - apk add --no-cache git bash
61     - git clone https://github.com/docker/docker-bench-security.git
62     - cd docker-bench-security
63     - chmod +x docker-bench-security.sh
64     - ./docker-bench-security.sh | tee ./reports/docker-bench-report.txt
65  artifacts:
66    paths:
67      - ./reports/docker-bench-report.txt
68    expire_in: 1 week
69    allow_failure: true
70
71 # Additional security scan using GitLeaks to find secrets
72 secrets_check:
73   stage: security
74   image: alpine:latest # Correcting this to use Alpine for the GitLeaks scan
75   script:
```

```
# Additional security scan using GitLeaks to find secrets
secrets_check:
  stage: security
  image: alpine:latest # Correcting this to use Alpine for the GitLeaks scan
  script:
    - apk add --no-cache curl tar
    - curl -s$FL https://github.com/zricethezav/gitleaks/releases/download/v8.16.3/gitleaks_8.16.3_linux_x64.tar.gz | tar -xzf - -C /usr/local/bin
    - mkdir -p ./reports
    - gitleaks detect --source=. --report-path=./reports/gitleaks-report.json || true # Run GitLeaks for secret scanning
  artifacts:
    paths:
      - ./reports/gitleaks-report.json
    expire_in: 1 week
    allow_failure: true
```

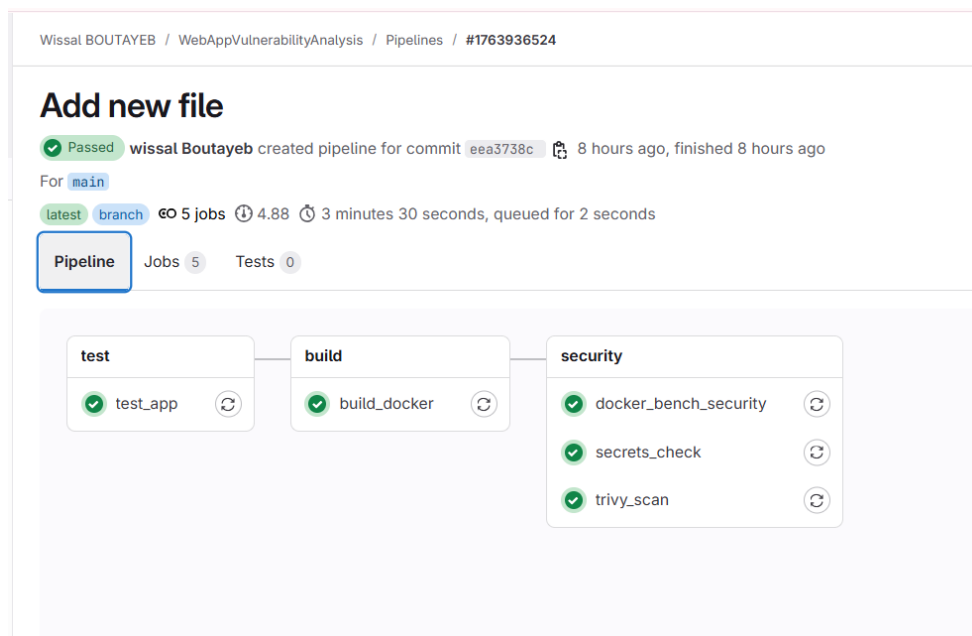
Le Fichier **Dockerfile** complète :



Ce Fichier **Dockerfile** est **utilisée** pour containeriser Notre Application

## Résultats Attendus :

- Pipeline automatisé avec intégration de la sécurité.



For **main**

**latest** **branch** **eo** 5 jobs 4.88 3 minutes 30 seconds, queued for 2 seconds

Pipeline **Jobs** 5 Tests 0

Status	Job	Stage	Coverage
Passed	#9694543480: secrets_check main → eea3738c	security	
Passed	#9694543477: docker_bench_security main → eea3738c	security	
Passed	#9694543470: trivy_scan main → eea3738c	security	
Passed	#9694543458: build_docker main → eea3738c	build	
Passed	#9694543443: test_app main → eea3738c	test	

Search visible log output

15 Initialized empty Git repository in /builds/wissal-boutayeb/WebAppVulnerabilityAnalysis/.git/  
16 Created fresh repository.  
17 Checking out eea3738c as detached HEAD (ref is main)...  
18 Skipping Git submodule setup  
19 \$ git remote set-url origin "\${CI\_REPOSITORY\_URL}" || echo 'Not a git repository; skipping'  
20 Executing 'step\_script' step of the job script  
21 Using docker image sha256:915da63721b1b4d3f4b02a6d0d338b7ae93c3353bde89d490f8b99f9eb63 for docker:latest with digest docker:sha256:ddb0833088b4fab7481ade341a582e3c6c8021b23770bba1a18ab33ed44 ...  
22 \$ docker build -t monapp:latest .  
23 #0 building with "default" instance using docker driver  
24 #1 [internal] load build definition from Dockerfile  
25 #1 transferring dockerfile: 566B done  
26 #1 DONE 0.8s  
27 #2 [internal] load metadata for docker.io/library/node:28  
28 #2 DONE 0.5s  
29 #3 [internal] load .dockerignore  
30 #3 transferring context: 28B done  
31 #3 DONE 0.8s  
32 #4 [1/5] FROM docker.io/library/node:28@sha256:a5fb835ac1dff34a4ecaea85f90f7321185695d3fd22c12ba12f4535a4647cc5  
33 #4 resolve docker.io/library/node:28@sha256:a5fb835ac1dff34a4ecaea85f90f7321185695d3fd22c12ba12f4535a4647cc5 0.8s done  
34 # sha256:244951c16cd04201215a3473d69d336a4e2fceb7f7acc0f44163e0385327 2.49kB / 2.49kB done  
35 # sha256:73b7d26ef1d29425dada0d70ce374277b9aab5ca68301507331a005cb3d33849 0B / 48.49MB 0.1s  
36 # sha256:a5fb835ac1dff34a4ecaea85f90f7321185695d3fd22c12ba12f4535a4647cc5 6.41kB / 6.41kB done  
37 # sha256:dd0c568d4c23d878128b09f5dd33688b6a83698c5e38a82433beb58eb59a81b 6.39kB / 6.39kB done  
38 # sha256:07d15af933d2dfc3d0dd509d6e28534825e4a53777b08a6ac5b8e5a1f20905 0B / 24.81MB 0.1s  
39 # sha256:1eb98adb0eb44a2e4facf9ca3a26a4a6feddbd5d159cca90a35285744e7 0B / 64.40MB 0.1s  
40 # sha256:23b7d26ef1d29425dada0d70ce374277b9aab5ca68301507331a005cb3d33849 5.80MB / 48.49MB 0.2s  
41 # sha256:23b7d26ef1d29425dada0d70ce374277b9aab5ca68301507331a005cb3d33849 10.50MB / 48.49MB 0.3s  
42 # sha256:07d15af933d2dfc3d0dd509d6e28534825e4a53777b08a6ac5b8e5a1f20905 5.24MB / 24.81MB 0.3s  
43 # sha256:1eb98adb0eb44a2e4facf9ca3a26a4a6feddbd5d159cca90a35285744e7 4.49MB / 64.40MB 0.3s  
44 # sha256:23b7d26ef1d29425dada0d70ce374277b9aab5ca68301507331a005cb3d33849 30.41MB / 48.49MB 0.5s  
45 # sha256:07d15af933d2dfc3d0dd509d6e28534825e4a53777b08a6ac5b8e5a1f20905 24.81MB / 24.81MB 0.5s  
46 # sha256:1eb98adb0eb44a2e4facf9ca3a26a4a6feddbd5d159cca90a35285744e7 23.82MB / 64.40MB 0.5s  
47 #4 ...  
48 #5 [internal] load build context

Duration: 1 minute 58 seconds  
Finished: 10 hours ago  
Queued: 0 seconds  
Timeout: 1h (from project)  
Runner: #12270840 (-AzERasQ) 5-blue.saas-linux-small-amd64.runners-manager.gitlab.com/default  
Source: Push  
Job artifacts  
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.  
Keep Download Browse  
Commit eea3738c  
Add new file  
Pipeline #1763936524 **Passed** for main  
build  
Related jobs  
→ build\_docker

Wissal BOUTAYEB / WebAppVulnerabilityAnalysis / Jobs / #9694543458

Search visible log output

103 # 1.731 npm warn deprecated nlog@5.0.1: This package is no longer supported.  
104 # 1.734 npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.  
105 # 1.828 npm warn deprecated are-we-there-yet@2.0.0: This package is no longer supported.  
106 # 1.836 npm warn deprecated gauge@3.0.2: This package is no longer supported.  
107 # 1.862 npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported  
108 # 2.348  
109 # 2.348 added 158 packages, and audited 159 packages in 2s  
110 # 2.350  
111 # 2.350 21 packages are looking for funding  
112 # 2.350 run 'npm fund' for details  
113 # 2.350  
114 # 2.350 found 0 vulnerabilities  
115 # 2.351 npm notice  
116 # 2.351 npm notice New major version of npm available! 10.8.2 -> 11.3.0  
117 # 2.351 npm notice ChangeLog: <https://github.com/npm/cli/releases/tag/v11.3.0>  
118 # 2.351 npm notice To update run: npm install -g npm@11.3.0  
119 # 2.351 npm notice  
120 # DONE 2.5s  
121 # [5/5] COPY . .  
122 # DONE 0.3s  
123 #10 exporting to image  
124 #10 exporting layers  
125 #10 exporting layers 0.3s done  
126 #10 writing image sha256:6f30bd496e8d288b27673176fa48f504b5596f3af73785f5eb48dccc34ad done  
127 #10 naming to docker.io/library/monapp:latest done  
128 #10 DONE 0.3s  
129 \$ mkdir -p ./artifacts  
130 \$ docker save monapp:latest > ./artifacts/monapp-image.tar  
131 Uploading artifacts for successful job  
132 Uploading artifacts...  
133 ./artifacts/monapp-image.tar: found 1 matching artifact files and directories  
134 Uploading artifacts as "archive" to coordinator... 201 Created 169694543458 responseStatus:201 Created tokenref:jrnQ10  
135 Cleaning up project directory and file based variables  
136 Job succeeded

Duration: 1 minute 58 seconds  
Finished: 10 hours ago  
Queued: 0 seconds  
Timeout: 1h (from project)  
Runner: #12270840 (-AzERasQ) 5-blue.saas-linux-small-amd64.runners-manager.gitlab.com/default  
Source: Push  
Job artifacts  
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.  
Keep Download Browse  
Commit eea3738c  
Add new file  
Pipeline #1763936524 **Passed** for main  
build  
Related jobs  
→ build\_docker

Search visible log output

167 [INFO] 4.9 - Ensure that COPY is used instead of ADD in Dockerfiles (Manual)  
168 [PASS] 4.9 - Ensure that COPY is used instead of ADD in Dockerfiles (Manual)  
169 [NOTE] 4.10 - Ensure secrets are not stored in Dockerfiles (Manual)  
170 [NOTE] 4.11 - Ensure only verified packages are installed (Manual)  
171 [NOTE] 4.12 - Ensure all signed artifacts are validated (Manual)  
172 [INFO] 5 - Container Runtime  
173 [INFO] \* No containers running, skipping Section 5  
174 [PASS] 5.1 - Ensure swarm mode is not Enabled, if not needed (Automated)  
175 [INFO] 6 - Docker Security Operations  
176 [INFO] 6.1 - Ensure that image sprawl is avoided (Manual)  
177 [INFO] \* There are currently: 0 Images  
178 [INFO] 6.2 - Ensure that container sprawl is avoided (Manual)  
179 [INFO] \* There are currently a total of 0 containers, with 0 of them currently running  
180 [INFO] 7 - Docker Swarm Configuration  
181 [PASS] 7.1 - Ensure that the minimum number of manager nodes have been created in a swarm (Automated) (Swarm mode not enabled)  
182 [PASS] 7.2 - Ensure that swarm services are bound to a specific host interface (Automated) (Swarm mode not enabled)  
183 [PASS] 7.3 - Ensure that all Docker swarm overlay networks are encrypted (Automated)  
184 [PASS] 7.4 - Ensure that Docker's secret management commands are used for managing secrets in a swarm cluster (Manual) (Swarm mode not enabled)  
185 [PASS] 7.5 - Ensure that swarm manager is run in auto-lock mode (Automated) (Swarm mode not enabled)  
186 [PASS] 7.6 - Ensure that the swarm manager auto-lock key is rotated periodically (Manual) (Swarm mode not enabled)  
187 [PASS] 7.7 - Ensure that node certificates are rotated as appropriate (Manual) (Swarm mode not enabled)  
188 [PASS] 7.8 - Ensure that CA certificates are rotated as appropriate (Manual) (Swarm mode not enabled)  
189 [PASS] 7.9 - Ensure that management plane traffic is separated from data plane traffic (Manual) (Swarm mode not enabled)  
190 Section C - Score  
191 [INFO] Checks: 86  
192 [INFO] Score: 2  
193 Uploading artifacts for successful job  
194 Uploading artifacts...  
195 ./reports/docker-bench-report.txt: found 1 matching artifact files and directories  
196 Uploading artifacts as "archive" to coordinator... 201 Created id=9694543477 responseStatus=201 Created token=eyJraWQ1  
197 Cleaning up project directory and file based variables  
198 Job succeeded

Duration: 48 seconds  
Finished: 10 hours ago  
Queued: 0 seconds  
Timeout: 1h (from project)  
Runner: #12270837 (J2nyww-s) 4-blue-saas-linux-small-amd64.runners-manager.gitlab.com/default  
Source: Push  
Job artifacts  
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.  
Keep Download Browse  
Commit aa33738c  
Add new file  
Pipeline #1763936524 Passed for main  
security  
Related jobs  
→ docker\_bench\_security  
secrets\_check  
trivy\_scan

secrets\_check

Passed Started 10 hours ago by Wissal Boutayeb

Search visible log output

1 Running with gitlab-runner 17.10.0-prerelease (5c23fd8e)  
2 on blue-4-saas-linux-small-amd64.runners-manager.gitlab.com/default J2nyww-s, system ID: s-ef1798852952  
3 Preparing the "docker-machine" executor  
4 Using Docker executor with image alpine:latest ...  
5 Pulling docker image alpine:latest ...  
6 Using docker image sha256:a856b3e08316479c315983c0ba2ee5428ecdb for alpine:latest with digest alpine@sha256:a856b3e08316479c315983c0ba2ee5428ecdb for alpine:latest with digest alpine@sha256:a856b3e08316479c315983c0ba2ee5428ecdb ...  
7 Preparing environment  
8 Running on runner-j2nyww-s-project-68863428-concurrent-0 via runner-j2nyww-s-s-l-s-amd64-1744363288-adb0b079...  
9 Getting source from git repository  
10 Fetching changes with git depth set to 20...  
11 Initialized empty Git repository in /builds/wissal-boutayeb/WebAppVulnerabilityAnalysis/.git/  
12 Created fresh repository.  
13 Checking out aa33738c as detached HEAD (ref is main)...  
14 Skipping Git submodules setup  
15 \$ git remote set-url origin '\$CI\_REPOSITORY\_URL' || echo 'Not a git repository; skipping'  
16 Downloading artifacts  
17 Downloading artifacts for build\_docker (9694543458)...  
18 Downloading artifacts from coordinator... ok host=storage.googleapis.com id=9694543458 responseStatus=200 OK token=eyJraWQ1  
19 Executing "setup\_script" stage of the job script  
20 Using docker image sha256:a856b3e08316479c315983c0ba2ee5428ecdb for alpine:latest with digest alpine@sha256:a856b3e08316479c315983c0ba2ee5428ecdb for alpine:latest with digest alpine@sha256:a856b3e08316479c315983c0ba2ee5428ecdb ...  
21 \$ apk add --no-cache curl tar  
22 fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86\_64/APKINDEX.tar.gz  
23 fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/community/x86\_64/APKINDEX.tar.gz  
24 (2/1) Installing brotli-libs (1.1.0-r2)  
25 (2/1) Installing c-ares (1.34.6-r0)  
26 (3/1) Installing libunistring (1.2-r0)  
27 (4/1) Installing libidn2 (2.3.3-r0)  
28 (5/1) Installing nghttp2-libs (1.44.0-r0)

Duration: 33 seconds  
Finished: 10 hours ago  
Queued: 1 second  
Timeout: 1h (from project)  
Runner: #12270837 (J2nyww-s) 4-blue-saas-linux-small-amd64.runners-manager.gitlab.com/default  
Source: Push  
Job artifacts  
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.  
Keep Download Browse  
Commit aa33738c  
Add new file  
Pipeline #1763936524 Passed for main  
security  
Related jobs  
→ docker\_bench\_security  
secrets\_check  
trivy\_scan

Search visible log output

37 OK: 28 hits of 1 packages  
38 \$ egrep -oD - https://github.com/aquasecurity/trivy/releases/download/v0.41.8/trivy\_0.41.8\_Linux-64bit.tar.gz | tar -zxvf -  
39 LICENSE  
40 README.md  
41 contrib/assf.tpl  
42 contrib/gittlab-codequality.tpl  
43 contrib/gittlab.tpl  
44 contrib/html.tpl  
45 contrib/junit.tpl  
46 trivy  
47 \$ mkdir -p ./reports  
48 \$ ./trivy image --no-progress --format json --output ./reports/trivy-node-report.json mode:ia  
49 2025-04-11T09:22:45.567Z INFO Need to update DB  
50 2025-04-11T09:22:45.567Z INFO DB Repository: ghcr.io/aquasecurity/trivy-db  
51 2025-04-11T09:22:45.567Z INFO Downloading DB...  
52 2025-04-11T09:22:45.667Z INFO Vulnerability scanning is enabled  
53 2025-04-11T09:22:45.667Z INFO Secret scanning is enabled  
54 2025-04-11T09:22:45.667Z INFO If your scanning is slow, please try '--scanners vuln' to disable secret scanning  
55 2025-04-11T09:22:45.667Z INFO Please see also https://aquasecurity.github.io/trivy/v0.41/docs/secret/scanning/#recommendation for faster secret detection  
56 2025-04-11T09:23:07.753Z INFO Detected OS: debian  
57 2025-04-11T09:23:07.753Z INFO Detecting Debian vulnerabilities...  
58 2025-04-11T09:23:07.889Z INFO Number of language-specific files: 1  
59 2025-04-11T09:23:07.889Z INFO Detecting node-pkg vulnerabilities...  
60 2025-04-11T09:23:08.148Z WARN This OS version is no longer supported by the distribution: debian 10.13  
61 2025-04-11T09:23:08.148Z WARN The vulnerability detection may be insufficient because security updates are not provided  
62 Uploading artifacts for successful job  
63 Uploading artifacts...  
64 ./reports/: found 2 matching artifact files and directories  
65 Uploading artifacts as "archive" to coordinator... 201 Created id=9694543470 responseStatus=201 Created token=eyJraWQ1  
66 Cleaning up project directory and file based variables  
67 Job succeeded

Duration: 59 seconds  
Finished: 10 hours ago  
Queued: 0 seconds  
Timeout: 1h (from project)  
Runner: #12270840 (AzE8a2) 5-blue-saas-linux-small-amd64.runners-manager.gitlab.com/default  
Source: Push  
Job artifacts  
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.  
Keep Download Browse  
Commit aa33738c  
Add new file  
Pipeline #1763936524 Passed for main  
security  
Related jobs  
→ docker\_bench\_security  
secrets\_check  
trivy\_scan



Wissal BOUTAYEB / WebAppVulnerabilityAnalysis / Jobs / #9694543470

### trivy\_scan

Passed Started 10 hours ago by wissal Boutayeb

Search visible log output

Q

📄

⇌

↑

↓

↗

```
1 Running with gitlab-runner 17.10.0-pre.41.g5c23fd8e (5c23fd8e)
2 on blue-5.saas-linux-small-amd64.runners-manager.gitlab.com/default -AzERasQ, system ID: s_4cb9c9ee29e2
3 Preparing the "docker+machine" executor
4 Using Docker executor with image alpine:latest ...
5 Pulling docker image alpine:latest ...
6 Using docker image sha256:aded1e1a5b3795116fa8a92ba974a5e0b0831647d9c315983c8ba2ee5428ec8b for alpine:latest with digest alpine@sha256:a856b36e8b8210a34f77d9f779ef07ffa463a380b75e2e74aff4511df3ef88c ...
7 Preparing environment
8 Running on runner-azerasq-project-68863428-concurrent-0 via runner-azerasq-s-l-s-amd64-1744363790-d879d688...
9 Getting source from Git repository
10 Fetching changes with git depth set to 20...
11 Initialized empty Git repository in /builds/wissal-boutayeb/WebAppVulnerabilityAnalysis/.git/
12 Created fresh repository.
13 Checking out eea3738c as detached HEAD (ref is main)...
14 Skipping Git submodules setup
15 $ git remote set-url origin "${CI_REPOSITORY_URL}" || echo 'Not a git repository; skipping'
16 Downloading artifacts
17 Downloading artifacts for build_docker (9a94543458)...
18 Downloading artifacts from coordinator... ok host=storage.googleapis.com id=9a94543458 responseStatus=200 OK token=eyJraWQ10
19 Executing "step_script" stage of the job script
20 Using docker image sha256:aded1e1a5b3795116fa8a92ba974a5e0b0831647d9c315983c8ba2ee5428ec8b for alpine:latest with digest alpine@sha256:a856b36e8b8210a34f77d9f779ef07ffa463a380b75e2e74aff4511df3ef88c ...
21 $ apk add --no-cache wget
22 Fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86_64/APKINDEX.tar.gz
23 Fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/community/x86_64/APKINDEX.tar.gz
24 (1/4) Installing libunistring (1.2-r0)
25 (2/4) Installing libidn2 (2.3.7-r0)
26 (3/4) Installing pcre2 (10.43-r0)
27 (4/4) Installing wget (1.25.0-r0)
28 Executing busybox-1.37.0-r12.trigger
29 OK: 18 MiB in 19 packages
30 $ wget -O - https://github.com/aquasecurity/trivy/releases/download/v0.43.0/trivy_0.43.0_Linux-64bit.tar.gz | tar -zxvf -
```

Duration: 59 seconds

Finished: 10 hours ago

Queued: 0 seconds

Timeout: 1h (from project)

Runner: #12270840 (-AzERasQ) 5-blue.saas-linux-small-amd64.runners-manager.gitlab.com/default

Source: Push

Job artifacts

These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.

Keep Download Browse

Commit eea3738c

Add new file

Pipeline #1763936524 Passed for main

security

Related jobs

docker\_bench\_security

secrets\_check

trivy\_scan

Wissal BOUTAYEB / WebAppVulnerabilityAnalysis / Jobs / #9694543443

### test\_app

Passed Started 10 hours ago by wissal Boutayeb

Search visible log output

Q

📄

⇌

↑

↓

↗

```
1 Running with gitlab-runner 17.10.0-pre.41.g5c23fd8e (5c23fd8e)
2 on blue-2.saas-linux-small-amd64.runners-manager.gitlab.com/default XxUrkriX, system ID: s_f46a988edce4
3 Preparing the "docker+machine" executor
4 Using Docker executor with image node:20 ...
5 Pulling docker image node:20 ...
6 Using docker image sha256:d08c568d4c23d878128b09f5dd336086b683698c5e38a824433beb58eb59a81b for node:20 with digest node@sha256:a5fb835ac1dff34a4ecaea85f90f7321185695d3fd2c212ba12f4535a4647cc5 ...
7 Preparing environment
8 Running on runner-xxurkriX-project-68863428-concurrent-0 via runner-xxurkriX-s-l-s-amd64-1744363107-172fedab...
9 Getting source from Git repository
10 Fetching changes with git depth set to 20...
11 Initialized empty Git repository in /builds/wissal-boutayeb/WebAppVulnerabilityAnalysis/.git/
12 Created fresh repository.
13 Checking out eea3738c as detached HEAD (ref is main)...
14 Skipping Git submodules setup
15 $ git remote set-url origin "${CI_REPOSITORY_URL}" || echo 'Not a git repository; skipping'
16 Executing "step_script" stage of the job script
17 Using docker image sha256:d08c568d4c23d878128b09f5dd336086b683698c5e38a824433beb58eb59a81b for node:20 with digest node@sha256:a5fb835ac1dff34a4ecaea85f90f7321185695d3fd2c212ba12f4535a4647cc5 ...
18 $ echo "Running application tests..."
19 Running application tests...
20 $ echo "Tests passed!"
21 Tests passed!
22 Cleaning up project directory and file based variables
23 Job succeeded
```

Duration: 32 seconds

Finished: 10 hours ago

Queued: 0 seconds

Timeout: 1h (from project)

Runner: #12270831 (XxUrkriX) 2-blue.saas-linux-small-amd64.runners-manager.gitlab.com/default

Source: Push

Commit eea3738c

Add new file

Pipeline #1763936524 Passed for main

test

Related jobs

test\_app

## Questions d'évaluation

### Quelle est la différence entre un job et un stage dans GitLab CI ?

- **Job** : C'est une Tâche individuelle (ex: test\_unitaire).
- **Stage** : C'est un Groupe de jobs (ex: test, security).

### Quelle est la différence entre SAST, DAST, SCA ?

- **SAST** : Analyse statique du code source.
- **DAST** : Test dynamique en environnement d'exécution.
- **SCA** : Scan des dépendances logicielles.

**Que permet l'outil Trivy ? Et Docker Bench ?**

- **Trivy** : Scan des vulnérabilités dans les images Docker.
- **Docker Bench** : Audit des configurations Docker.

**Que faire si une faille critique est détectée ?**

Stopper le déploiement de mon application et essayer de trouver la vulnérabilité et la faille de Sécurité qui a été détectée et la corriger avant qu'il soit exploité

**Pourquoi faut-il éviter l'usage du user root dans un conteneur ?**

Réduire les risques d'exploitation tout en attribuant juste le privilège nécessaire à une tâche précise

**Conclusion :**

Ce TP nous a permis de mettre en pratique une approche **DevSecOps** en intégrant des outils de sécurité automatisés dans un pipeline GitLab CI/CD. Les principales tâches réalisées incluent :

**1. Configuration du Pipeline CI/CD :**

- Structuration des **stages** (test, build, security).
- Exécution de **jobs** automatisés (tests unitaires, build Docker).

**2. Intégration de la Sécurité dans le Pipeline :**

- **SCA avec Snyk** : Détection des vulnérabilités dans les dépendances.
- **Détection de secrets avec GitLeaks** : Prévention des fuites de données sensibles.
- **DAST avec OWASP ZAP** : Analyse dynamique des failles applicatives.
- **Scan d'images Docker avec Trivy** : Identification des CVE dans les conteneurs.
- **Audit de configuration avec Docker Bench** : Vérification des bonnes pratiques Docker.

**Lien du Projet GitLab**

**Dépôt GitLab** : <https://gitlab.com/wissal-boutayeb/WebAppVulnerabilityAnalysis.git>