

# Rapport DevOps :

Encadré par : Monsieur DELIGNY Manoel

## Auteurs :

BOUKHRIS Mohamed Lyazid

BELAHSEN Ilyes

# **1. Environnement et Infrastructure :**

## **1.1 GitLab Runner et Registre Docker :**

Nous avons installé le GitLab Runner sur la VM en utilisant la commande sudo apt install gitlab-runner, puis enregistré le runner avec notre projet GitLab.

Il est configuré avec l'exécuteur shell, ce qui nous permet d'exécuter les jobs directement dans l'environnement local de la VM.

Le runner est bien associé au projet et utilisé par notre pipeline pour exécuter les étapes de test, build et déploiement.

Le registre Docker de GitLab a été activé dans les paramètres du projet.

Les images construites par le pipeline sont poussées automatiquement dans ce registre interne, et seront ensuite utilisées pour le déploiement sur Kubernetes.

```
ibelahsen@fisa2025-ibelahsen-1:~/devOps$ sudo gitlab-runner status
Runtime platform                               arch=amd64 os=linux pid=621376 revision=96856197 version=17.11.1
gitlab-runner: Service is running
ibelahsen@fisa2025-ibelahsen-1:~/devOps$ sudo cat /etc/gitlab-runner/config.toml
concurrent = 1
check_interval = 0
connection_max_age = "15m0s"
shutdown_timeout = 0

[session_server]
  session_timeout = 1800

[[runners]]
  name = "runner-devops"
  url = "https://gitlab.com/"
  id = 47439816
  token = "glrt-VD-K9qBWIVPockD5RXzcLm86MQpw0jE1anF2MQp00jMKdTpnanBocRg.01.1j1kwcih"
  token_obtained_at = 2025-05-14T07:16:40Z
  token_expires_at = 0001-01-01T00:00:00Z
  executor = "shell"
  tags = ["polytech", "tp"]
  [runners.cache]
    MaxUploadedArchiveSize = 0
    [runners.cache.s3]
    [runners.cache.gcs]
    [runners.cache.azure]
ibelahsen@fisa2025-ibelahsen-1:~/devOps$
```

```
ibelahsen@fisa2025-ibelahsen-1:~/devOps$ sudo gitlab-runner verify
Runtime platform                               arch=amd64 os=linux pid=983747 revision=96856197 version=17.11.1
Running in system-mode.

Verifying runner... is valid
runner=VD-K9qBWIVPockD5RXzcLm86MQpw0jE1anF2MQp00jMKdTpnanBocRg.01.1j1kwcih
ibelahsen@fisa2025-ibelahsen-1:~/devOps$ sudo gitlab-runner list
Runtime platform                               arch=amd64 os=linux pid=983884 revision=96856197 version=17.11.1
Listing configured runners
ConfigFile=/etc/gitlab-runner/config.toml
runner-devops                           Executor=shell Token=glrt-VD-K9qBWIVPockD5RXzcLm86MQpw0jE1anF2MQp00jMKdTpnanBocRg.01.1j1kwcih URL=https://gitlab.com/
ibelahsen@fisa2025-ibelahsen-1:~/devOps$
```

- **Figure 1.1 : État du GitLab Runner et sa configuration**

## ✓ Runners

Les runners sont des processus qui vont chercher des jobs CI/CD et les exécutent pour GitLab. Qu'est-ce que GitLab Runner ?

Runners du projet 1  
Ces runners sont assignés à ce projet.

Runners du groupe 0  
Ces runners sont partagés par tous les projets de ce groupe. Les runners de groupe peuvent être gérés avec l'API Runner.

Ce projet ne fait pas partie d'un groupe et ne peut pas utiliser des runners de groupe.

Runners d'instance 108  
Ces runners sont disponibles pour tous les groupes et tous les projets.  
Each CI/CD job runs on a separate, isolated virtual machine.

Activer les runners d'instance pour ce projet

- **Figure 1.2 :** Runner enregistré dans l'interface GitLab

## **1.2 Kubernetes local avec k3d :**

Nous avons installé k3d sur la VM pour créer un cluster Kubernetes local avec un seul nœud.

Après création du cluster, nous avons vérifié son bon fonctionnement avec les commandes kubectl cluster-info et kubectl get nodes, qui confirment que le cluster est opérationnel.

Ce cluster sera utilisé pour déployer nos images Docker à travers le pipeline CI/CD.

```
ibelahsen@fisa2025-ibelahsen-1:~/devOps$ k3d version
k3d version v5.8.3
k3s version v1.31.5+k3s1 (default)
ibelahsen@fisa2025-ibelahsen-1:~/devOps$ k3d cluster list
NAME      SERVERS  AGENTS  LOADBALANCER
polytech-cluster  1/1     0/0    true
ibelahsen@fisa2025-ibelahsen-1:~/devOps$ kubectl cluster-info
Kubernetes control plane is running at https://0.0.0.0:37845
CoreDNS is running at https://0.0.0.0:37845/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://0.0.0.0:37845/api/v1/namespaces/kube-system/services/https:metrics-server:https/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
ibelahsen@fisa2025-ibelahsen-1:~/devOps$ kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
k3d-polytech-cluster-server-0   Ready   control-plane,master   2d10h   v1.31.5+k3s1
ibelahsen@fisa2025-ibelahsen-1:~/devOps$
```

- **Figure 1.3 :** Crédit et état du cluster Kubernetes local avec k3d

## 2. Conteneurisation :

### 2.1 Log Complet docker build :

```
ibelahsen@fisa2025-ibelahsen-1:~/devOps/devops_rendu$ docker compose up
WARN[0000] /home/ibelahsen/devOps/devops_rendu/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 5/5
  ✓ Network devops_rendu_default      Created                               0.3s
  ✓ Volume "devops_rendu_mariadb_data" Created                               0.05s
  ✓ Container devops-db               Created                               0.1s
  ✓ Container devops-backend          Created                               0.05s
  ✓ Container devops-frontend         Created                               0.1s
Attaching to devops-backend, devops-db, devops-frontend
devops-db | 2025-05-17 12:42:24+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.9.8+maria~ubu2204 started.
devops-db | 2025-05-17 12:42:24+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
devops-db | 2025-05-17 12:42:24+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.9.8+maria~ubu2204 started.
devops-db | 2025-05-17 12:42:24+00:00 [Note] [Entrypoint]: Initializing database files
devops-frontend |/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
devops-frontend |/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
devops-frontend |/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
devops-frontend | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
devops-frontend |/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
devops-frontend |/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
devops-frontend |/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
devops-frontend |/docker-entrypoint.sh: Configuration complete; ready for start up
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: using the "epoll" event method
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: nginx/1.27.5
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: built by gcc 14.2.0 (Alpine 14.2.0)
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: OS: Linux 5.10.0-34-amd64
devops-backend | Serveur lancé sur http://0.0.0.0:4242
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: start worker processes
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: start worker process 29
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: start worker process 30
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: start worker process 31
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: start worker process 32
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: start worker process 33
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: start worker process 34
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: start worker process 35
devops-frontend | 2025/05/17 12:42:24 [notice] 1#1: start worker process 36
```

```
devops-db | 2025-05-17 12:42:30 0 [Note] Starting MariaDB 10.9.8-MariaDB-1:10.9.8+maria~ubu2204 source revision 3e0009dc3a771e4dbf2fa4a4cf87e750453fb2eb as process 1
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: Number of transaction pools: 1
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
devops-db | 2025-05-17 12:42:30 0 [Note] mariadbd: Q_TMPFILE is not supported on /tmp (disabling future attempts)
devops-db | 2025-05-17 12:42:30 0 [Warning] mariadbd: io_uring_queue_init() failed with errno 1
devops-db | 2025-05-17 12:42:30 0 [Warning] InnoDB: liburing disabled: Falling back to innodb_use_native_aio=OFF
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: Initializing buffer pool, total size = 128.000MiB, chunk size = 2.000MiB
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: Completed initialization of buffer pool
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: File system buffers for log disabled (block size=512 bytes)
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: End of log at LSN=51423
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: 128 rollback segments are active.
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: Setting file './ibtmp1' size to 12.000MiB. Physically writing the file full; Please wait ...
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: File './ibtmp1' size is now 12.000MiB.
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: log sequence number 51423; transaction id 24
devops-db | 2025-05-17 12:42:30 0 [Note] Plugin 'FEEDBACK' is disabled.
devops-db | 2025-05-17 12:42:30 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
devops-db | 2025-05-17 12:42:30 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.
devops-db | 2025-05-17 12:42:31 0 [Note] Server socket created on IP: '0.0.0.0'.
devops-db | 2025-05-17 12:42:31 0 [Note] Server socket created on IP: '::'.
devops-db | 2025-05-17 12:42:31 0 [Note] mariadbd: ready for connections.
devops-db | Version: '10.9.8-MariaDB-1:10.9.8+maria~ubu2204' socket: '/run/mysqld/mysqld.sock' port: 3306 mariadb.org binary distribution
devops-db | 2025-05-17 12:42:31 0 [Note] InnoDB: Buffer pool(s) load completed at 250517 12:42:31
```

Figure 2.1 : log complet de docker build

Cette figure illustre la construction complète des images Docker du **backend (Node.js)** et du **frontend (Nginx)**.

Les conteneurs sont bien lancés, et la base de données **MariaDB** démarre correctement avec le **volume persistant**.

## 2.2 Docker volume ps :

```
● ibelahsen@fisa2025-ibelahsen-1:~/devOps/devops_rendu$ docker volume ls
DRIVER      VOLUME NAME
local        c9788a80b06ae21924bed7c0af3901ae4130d6fa33cb7793e7913d0619d31605
local        d328b18cf7261fb2ebcb6c52581d9247aa5627db8353a1c6c55dfcf7694bf80b
local        dadf51a730b1b035134fe02d85b2f069e2dfef2740a87192f73276c024caf7a2
local        devops_rendu_mariadb_data
local        ec521c152881ca132778b89b03fdf2fd1b8418a95bb6a07a50c96b83cd8e1db0
local        k3d-polytech-cluster-images
local        mysql-data
local        tp2_mysql-data
```

Figure 2.2 :

La commande docker volume ls montre que le volume **devops\_rendu\_mariadb\_data** a bien été créé.

Ce volume persiste les données de la base MariaDB entre les redémarrages des conteneurs, assurant la durabilité des données.

## 2.3 Docker run :

```
ibelahsen@fisa2025-ibelahsen-1:~/devOps/devops_rendu$ docker run --rm \
-v $(pwd)/backend/init.sql:/docker-entrypoint-initdb.d/init.sql:ro \
-e MARIADB_ROOT_PASSWORD=root \
-mariadb:10.9
2025-05-17 12:53:21+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.9.8+maria~ubu2204 started.
2025-05-17 12:53:21+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2025-05-17 12:53:21+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.9.8+maria~ubu2204 started.
2025-05-17 12:53:22+00:00 [Note] [Entrypoint]: Initializing database files
2025-05-17 12:53:22 0 [Warning] mariadb: ioURING_QUEUE_INIT() failed with errno 1
2025-05-17 12:53:22 0 [Warning] InnoDB: liburing disabled: falling back to innodb_use_native_aio=OFF

PLEASE REMEMBER TO SET A PASSWORD FOR THE MariaDB root USER !
To do so, start the server, then issue the following command:
'/usr/bin/mariadb-secure-installation'

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the MariaDB Knowledgebase at https://mariadb.com/kb
Please report any problems at https://mariadb.org/jira
The latest information about MariaDB is available at https://mariadb.org/ .
Consider joining MariaDB's strong and vibrant community:
https://mariadb.org/get-involved/

2025-05-17 12:53:23+00:00 [Note] [Entrypoint]: Database files initialized
2025-05-17 12:53:23+00:00 [Note] [Entrypoint]: Starting temporary server
2025-05-17 12:53:23+00:00 [Note] [Entrypoint]: Waiting for server startup
2025-05-17 12:53:23 0 [Note] Starting MariaDB 10.9.8-MariaDB-1:10.9.8+maria~ubu2204 source revision 3e0009dc3a771e4dbf2fa4a4cf87e750453fb2eb as process 107
2025-05-17 12:53:23 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
2025-05-17 12:53:23 0 [Note] InnoDB: Number of transaction pools: 1
2025-05-17 12:53:23 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
2025-05-17 12:53:23 0 [Note] mariadb: O_TMPFILE is not supported on /tmp (disabling future attempts)
2025-05-17 12:53:23 0 [Warning] mariadb: ioURING_QUEUE_INIT() failed with errno 1
2025-05-17 12:53:23 0 [Warning] InnoDB: liburing disabled: falling back to innodb_use_native_aio=OFF
2025-05-17 12:53:23 0 [Note] InnoDB: Initializing buffer pool, total size = 128.000MiB, chunk size = 2.000MiB
2025-05-17 12:53:23 0 [Note] InnoDB: Completed initialization of buffer pool
2025-05-17 12:53:23 0 [Note] InnoDB: File system buffers for log disabled (block size=512 bytes)
2025-05-17 12:53:23 0 [Note] InnoDB: End of log at LSN=45466
2025-05-17 12:53:23 0 [Note] InnoDB: 128 rollback segments are active.
2025-05-17 12:53:23 0 [Note] InnoDB: Setting file './ibtmp1' size to 12.000MiB. Physically writing the file full; Please wait ...
2025-05-17 12:53:23 0 [Note] InnoDB: File './ibtmp1' size is now 12.000MiB.
2025-05-17 12:53:23 0 [Note] InnoDB: log sequence number 45466; transaction id 14
2025-05-17 12:53:23 0 [Note] Plugin 'FEEDBACK' is disabled.
```

```

2025-05-17 12:53:23 0 [Note] Plugin 'FEEDBACK' is disabled.
2025-05-17 12:53:23 0 [Warning] 'user' entry 'root@eee64c93790c' ignored in --skip-name-resolve mode.
2025-05-17 12:53:23 0 [Warning] 'proxies_priv' entry '% root@eee64c93790c' ignored in --skip-name-resolve mode.
2025-05-17 12:53:23 0 [Note] mariadb: ready for connections.
Version: '10.9.8-MariaDB-1:10.9.8+maria~ubu2204' socket: '/run/mysqld/mysqld.sock' port: 0 mariadb.org binary distribution
2025-05-17 12:53:24+00:00 [Note] [Entrypoint]: Temporary server started.
2025-05-17 12:53:26+00:00 [Note] [Entrypoint]: Securing system users (equivalent to running mysql_secure_installation)

2025-05-17 12:53:26+00:00 [Note] [Entrypoint]: /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/init.sql

2025-05-17 12:53:26+00:00 [Note] [Entrypoint]: Stopping temporary server
2025-05-17 12:53:26 0 [Note] mariadb (initiated by: unknown): Normal shutdown
2025-05-17 12:53:26 0 [Note] InnoDB: FTS optimize thread exiting.
2025-05-17 12:53:26 0 [Note] InnoDB: Starting shutdown...
2025-05-17 12:53:26 0 [Note] InnoDB: Dumping buffer pool(s) to /var/lib/mysql/ib_buffer_pool
2025-05-17 12:53:26 0 [Note] InnoDB: Buffer pool(s) dump completed at 250517 12:53:26
2025-05-17 12:53:26 0 [Note] InnoDB: Removed temporary tablespace data file: "./ibtmp1"
2025-05-17 12:53:26 0 [Note] InnoDB: Shutdown completed; log sequence number 51389; transaction id 23
2025-05-17 12:53:26 0 [Note] mariadb: Shutdown complete

2025-05-17 12:53:26+00:00 [Note] [Entrypoint]: Temporary server stopped
2025-05-17 12:53:26+00:00 [Note] [Entrypoint]: MariaDB init process done. Ready for start up.

2025-05-17 12:53:26 0 [Note] Starting MariaDB 10.9.8+maria~ubu2204 source revision 3e0009dc3a771e4dbf2fa4a4cf87e750453fb2eb as process 1
2025-05-17 12:53:26 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
2025-05-17 12:53:26 0 [Note] InnoDB: Number of transaction pools: 1
2025-05-17 12:53:26 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
2025-05-17 12:53:26 0 [Note] mariadb: O_TMPFILE is not supported on /tmp (disabling future attempts)
2025-05-17 12:53:26 0 [Warning] mariadb: io_uring_queue_init() failed with errno 1
2025-05-17 12:53:26 0 [Warning] liburing disabled: falling back to innodb_use_native_aio=OFF
2025-05-17 12:53:26 0 [Note] InnoDB: Initializing buffer pool, total size = 128.000MiB, chunk size = 2.000MiB
2025-05-17 12:53:26 0 [Note] InnoDB: Completed initialization of buffer pool
2025-05-17 12:53:26 0 [Note] InnoDB: File system buffers for log disabled (block size=512 bytes)
2025-05-17 12:53:26 0 [Note] InnoDB: End of log at LSN=51389
2025-05-17 12:53:26 0 [Note] InnoDB: 128 rollback segments are active.
2025-05-17 12:53:26 0 [Note] InnoDB: Setting file './ibtmp1' size to 12.000MiB. Physically writing the file full; Please wait ...
2025-05-17 12:53:26 0 [Note] InnoDB: File './ibtmp1' size is now 12.000MiB.
2025-05-17 12:53:26 0 [Note] InnoDB: log sequence number 51389; transaction id 24
2025-05-17 12:53:26 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
2025-05-17 12:53:26 0 [Note] Plugin 'FEEDBACK' is disabled.
2025-05-17 12:53:26 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.
2025-05-17 12:53:26 0 [Note] Server socket created on IP: '::'.
2025-05-17 12:53:26 0 [Note] Server socket created on IP: '::'.
2025-05-17 12:53:26 0 [Note] InnoDB: Buffer pool(s) load completed at 250517 12:53:26
2025-05-17 12:53:26 0 [Note] mariadb: ready for connections.
Version: '10.9.8-MariaDB-1:10.9.8+maria~ubu2204' socket: '/run/mysqld/mysqld.sock' port: 3306 mariadb.org binary distribution

```

### Figure 2.3 : Démonstration du montage du volume et l'exécution de init.sql

La commande docker run monte le fichier init.sql dans le dossier /docker-entrypoint-initdb.d/.

On observe que le script est bien détecté et exécuté automatiquement **running /docker-entrypoint-initdb.d/init.sql** ce qui confirme le bon montage du volume.

MariaDB termine l'exécution sans erreur ce qui démontre la validité du script et de l'environnement.

## ***3. Qualité du code :***

### ***3.1 Back-end : Application Node.js + Express***

Le back-end de l'application a été développé avec Node.js et le framework Express.

L'architecture du code est organisée en trois dossiers :

**routes/** : contient les routes REST /lists et /todos

**controllers/** : contient la logique associée à chaque route (exécution, validation, réponse)

**models/** : exécute les requêtes SQL vers la base MariaDB via mysql2

Un fichier `.env` permet de configurer les paramètres de connexion à la base de données.

- **Gestion de la base de données :**

La base de données MariaDB contient deux tables :

lists : pour stocker les listes de tâches

todos : pour stocker les tâches associées à une liste

- **Tests unitaires avec Jest :**

Des tests unitaires ont été mis en place avec **Jest** et Supertest pour valider le bon fonctionnement des routes de l'API **back-end**.

Ces tests sont exécutés automatiquement à chaque pipeline CI/CD, ce qui permet de garantir la non-régression du code au fil des commits.

La couverture de code du **back-end** atteint plus de **91 %**, avec **100 %** de couverture sur les fonctions, ce qui démontre une bonne robustesse des tests.

De plus, **ESLint** a été configuré pour analyser le style et la qualité du code. Aucun avertissement ni erreur n'est détecté lors de l'exécution, assurant un code propre et maintenable.

Code Coverage Report					
File	% Stmt	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	91.95	58.33	100	92.94	
src	100	50	100	100	
app.js	100	100	100	100	
db.js	100	50	100	100	5-8
src/controllers	82.05	75	100	83.78	
ListsController.js	82.35	100	100	81.25	8,20,29
TodosController.js	81.81	50	100	85.71	25,35,44
src/models	100	100	100	100	
listsModel.js	100	100	100	100	
todosModel.js	100	100	100	100	
src/routes	100	100	100	100	
lists.js	100	100	100	100	
todos.js	100	100	100	100	

```
Test Suites: 4 passed, 4 total
Tests:       18 passed, 18 total
Snapshots:   0 total
Time:        3.742 s
Ran all test suites.
```

**Figure 3.1 :** Résultats des tests unitaires avec Jest et couverture du code

```
Lyazid@OrdiLyazid:~/devops_rendu/backend$ npm run lint

> backend@1.0.0 lint
> eslint .

Lyazid@OrdiLyazid:~/devops_rendu/backend$
```

**Figure 3.2 :** Résultat du linting avec ESLint sur le back-end

## 3.2 Front-end : Application web simple :

Le front-end de l'application a été développé avec **React** en utilisant une architecture modulaire.

L'organisation du code est structurée autour de trois dossiers principaux dans src/ :

- ❖ **components/** : contient les composants React ListComponent.js et TodoComponent.js, responsables de l'affichage et de l'interaction avec l'utilisateur
- ❖ **services/** : contient api.js, qui gère les appels HTTP vers l'API back-end
- ❖ **tests/unit/** : contient les tests unitaires des composants et services

L'interface permet de réaliser toutes les opérations CRUD sur les listes et les tâches.

Les appels aux routes back-end sont effectués avec fetch dans les fonctions du fichier api.js.

### - Tests unitaires avec Jest :

Des tests unitaires ont été mis en place avec **Jest** pour chaque composant (AppComponent, ListComponent, TodoComponent) ainsi que pour le service api.js.

Ils permettent de valider le bon fonctionnement de l'interface utilisateur, la logique de rendu et les appels API vers le back-end.

Les tests ont été exécutés avec succès (13 passed / 13 total).

La couverture de code globale sur le front-end est de plus de 80 %, avec 100 % de couverture des fonctions sur les composants.

Une configuration ESLint a également été mise en place. Aucun avertissement ou erreur n'a été détecté lors de son exécution, ce qui assure un code propre et conforme aux bonnes pratiques.

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	80.37	59.09	100	79.8	
src	85	50	100	85	
App.js	94.44	50	100	94.44	15
index.js	0	100	100	0	6-7
src/components	100	66.66	100	100	
ListComponent.js	100	50	100	100	13
TodoComponent.js	100	75	100	100	17
src/services	68.42	57.14	100	68.42	
api.js	68.42	57.14	100	68.42	28,32-33,44,48-49,58,62-63,78,82-83,98,102-103,114,118-119

Test Suites: 4 passed, 4 total  
Tests: 13 passed, 13 total  
Snapshots: 0 total  
Time: 42.557 s  
Ran all test suites related to changed files.

**Figure 3.3 :** Résultats des tests Jest et couverture du code sur le front-end

```
lyazid@OrdiLyazid:~/devops_rendu/frontend$ npm run lint

> frontend@0.1.0 lint
> eslint src/**/*.js

lyazid@OrdiLyazid:~/devops_rendu/frontend$
```

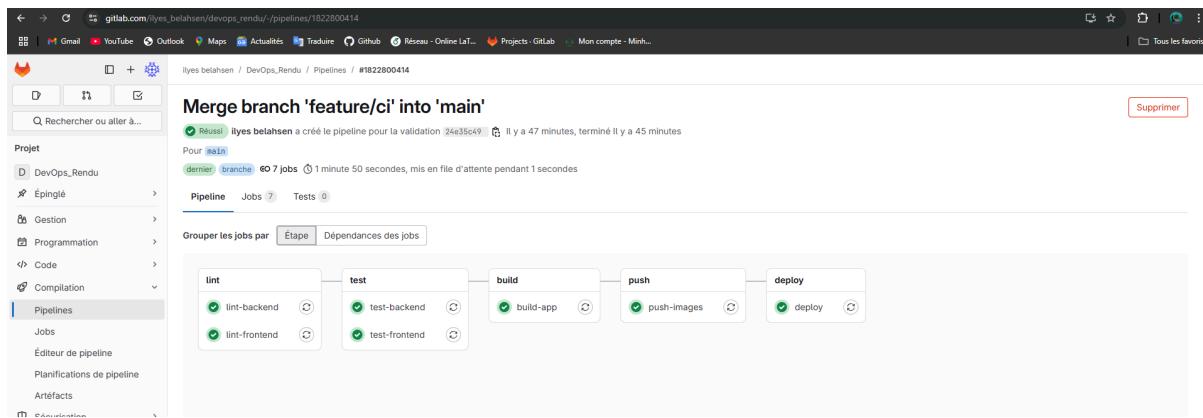
**Figure 3.4 :** Résultat du linting avec ESLint sur le back-end

## 4. CI/CD:

### 4.1 Pipeline GitLab CI - Validation des stages :

Les pipelines CI/CD ont été configurés pour automatiser le lint, les tests, le build des images Docker, le push vers le registre GitLab Container Registry, puis déploiement automatique.

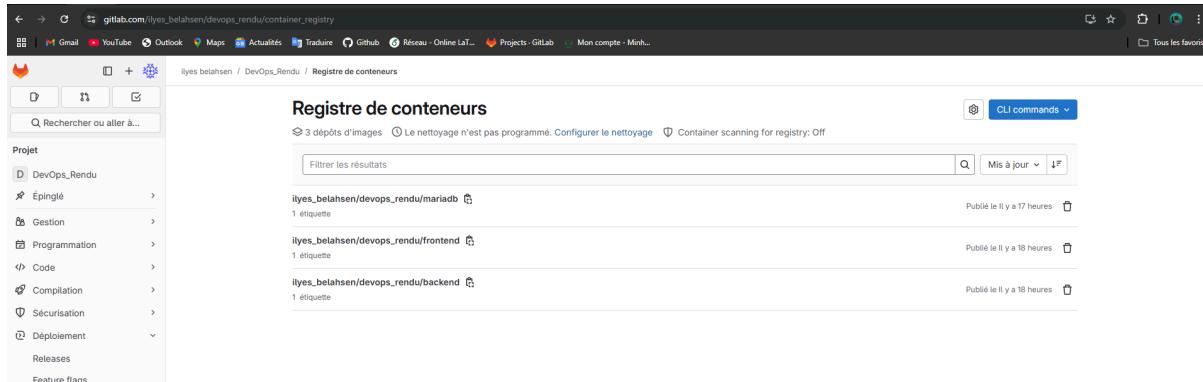
Ci-dessous, on retrouve une capture d'écran des logs du pipeline, montrant le succès de chaque étape : lint, test, build, push et déploiement.



**Figure 4.1 :** Résultats des pipelines CI/CD

## 4.2 Container Registry - Images et Tags

En complément, une capture d'écran de l'onglet Container Registry de GitLab montre les images Docker correspondantes avec leurs tags, attestant que les images sont bien construites et stockées dans le registre pour un déploiement ultérieur.

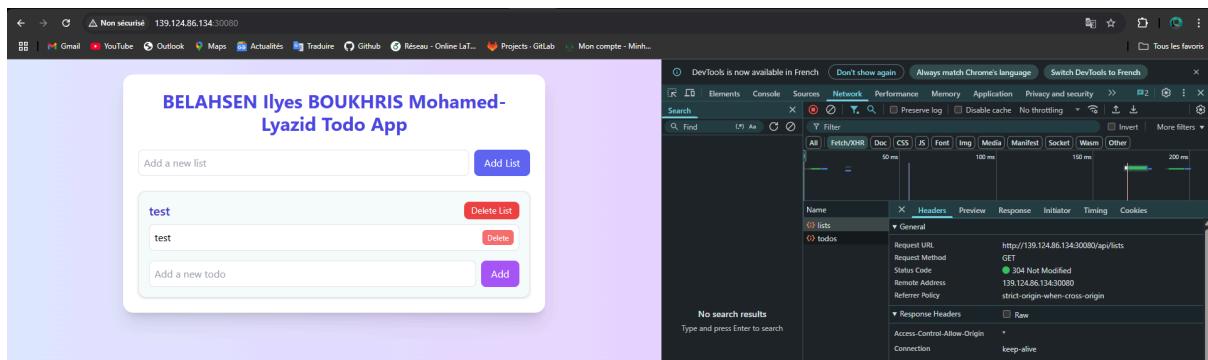


**Figure 4.2 :** Container Registry de GitLab

## **5. Déploiement & Validation:**

### **5.1 Interface CRUD - Accessibilité via l'IP de la VM**

L'application est accessible via l'URL : <http://139.124.86.134:30080/>



**Figure 5.1 :** Application accessible à l'[URL](http://139.124.86.134:30080/)

### **5.2 Accès Réseau - Instructions pour le réseau de l'école**

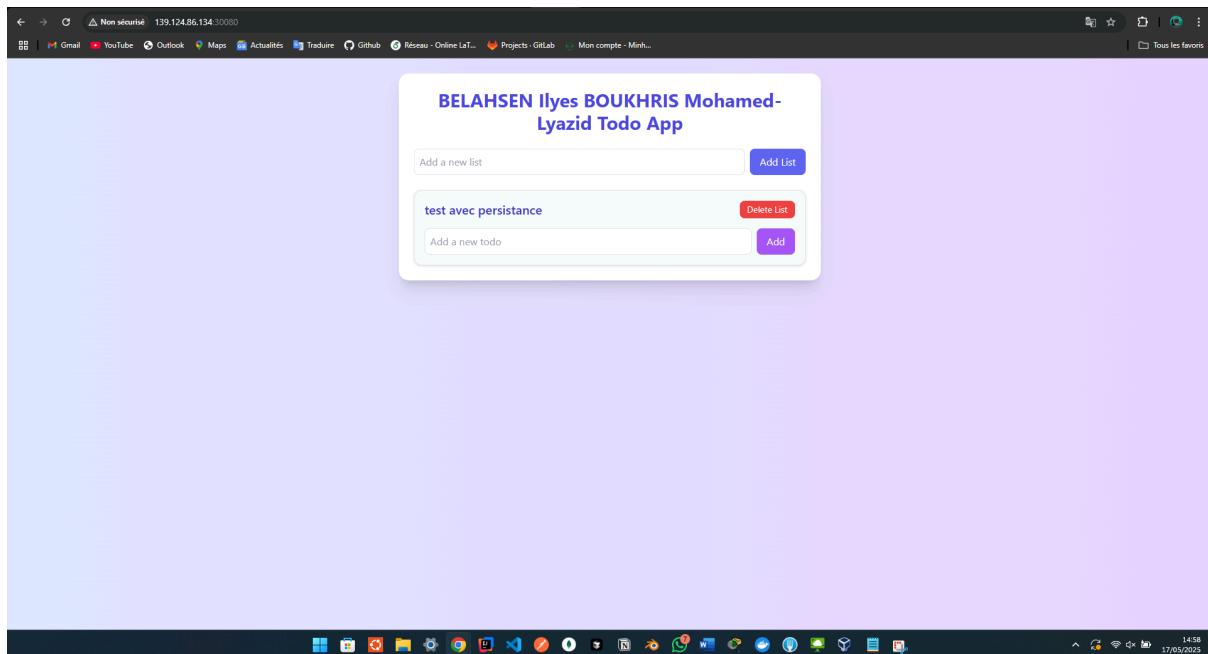
Afin de pouvoir accéder correctement à l'application il est nécessaire de se connecter au réseau wifi de l'université : Aix Marseille Université

### **5.3 Validation de la Persistance des Données**

Etant donné que c'est difficile de vous montrer à travers un rapport la preuve de la persistance de données, nous avons opté pour vous montrer des screens indiquant l'heure.

L'application est accessible via l'URL : <http://139.124.86.134:30080/>

Nous avons ajouté une entrée dans la liste avec la valeur « test avec persistance », ajouté à 14h58



**Figure 5.2 :** Image application prouvant la persistance de donnée à 14h58

Nous avons relancer les pipeline CI/CD, c'est à dire que l'application a été build puis les images Docker ont été poussé, puis l'application a été redéployé

Le redémarrage des pods a été validé via le job de déploiement, dont le succès est confirmé. Les logs du job montrent clairement les commandes exécutées pour relancer les pods. On peut voir sur le screen que sa a été réalisé à 15h04

```

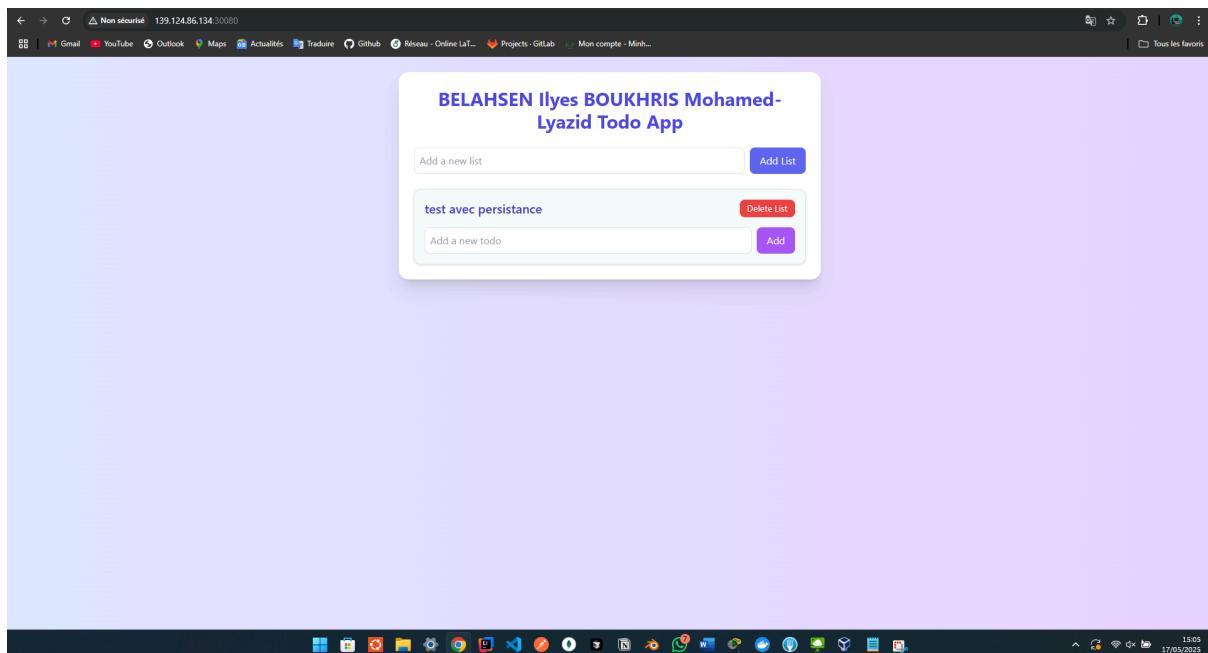
53 deployment.apps/devops-deployment unchanged
54 $ echo "==" Restarting deployments ==
55 === Restarting deployments ==
56 $ kubectl rollout restart deployment/backend-deployment
57 deployment.apps/backend-deployment restarted
58 $ kubectl rollout restart deployment/frontend-deployment
59 deployment.apps/frontend-deployment restarted
60 $ kubectl rollout restart deployment/mariadb-deployment
61 deployment.apps/mariadb-deployment restarted
62 $ echo "==" Pods detailed status ==
63 === Pods detailed status ==
64 $ kubectl get pods -o wide
65 NAME          READY   STATUS    RESTARTS   AGE     IP           NODE      NOMINATED NODE   READINESS
66 S GATES
67 backend-deployment-6fd8cc8678-k4lhv 0/1   ContainerCreating 0       8s    <none>        k3d-polytech-cluster-server-0  <none>    <none>
68 backend-deployment-7dbb9d79ff-tjzsd2 1/1   Running   0       60m   10.42.0.132  k3d-polytech-cluster-server-0  <none>    <none>
69 frontend-deployment-66f576d055-kzk85 0/1   ContainerCreating 0       8s    <none>        k3d-polytech-cluster-server-0  <none>    <none>
70 Frontend-deployment-984ff8fb86-zcsf2 1/1   Running   0       60m   10.42.0.133  k3d-polytech-cluster-server-0  <none>    <none>
71 mariadb-deployment-6ff9b7cd89-mkwdx 0/1   ContainerCreating 0       8s    <none>        k3d-polytech-cluster-server-0  <none>    <none>
72 mariadb-deployment-cf89b7f46-8zxvv 1/1   Running   0       60m   10.42.0.134  k3d-polytech-cluster-server-0  <none>    <none>
73 $ echo "==" Services ==
74 === Services ==
75 $ kubectl get svc
76 NAME          TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
77 devops-backend ClusterIP  10.43.89.170  <none>        4242/TCP  92m
78 frontend-service NodePort   10.43.181.138  <none>        80:30888/TCP  92m
79 kubernetes     ClusterIP  10.43.0.1    <none>        443/TCP   92m
80 mariadb-service ClusterIP  10.43.177.242  <none>        3306/TCP  92m
81 $ echo "==" Nodes ==
82 === Nodes ==
83 $ kubectl get nodes
84 NAME            STATUS   ROLES      AGE   VERSION
85 k3d-polytech-cluster-server-0 Ready    control-plane,master 3d5h  v1.31.5+k3s1
86 Cleaning up project directory and file based variables
87 Job succeeded

```

**Figure 5.3 :** Résultats du Job deploy

Après redémarrage des pods, nous avons accédé à nouveau à l'application à l'adresse <http://139.124.86.134:30080/> à 15h05

La donnée « test avec persistance » est toujours présente, ce qui prouve que la persistance fonctionne correctement.



**Figure 5.4 :** Image application prouvant la persistance de donnée à 15h05

## **Conclusion**

À l'issue de ce projet DevOps, nous avons réussi à mettre en place une infrastructure complète incluant la gestion de conteneurs via Docker, le déploiement sur Kubernetes avec k3d, et l'automatisation du pipeline CI/CD via GitLab. La persistance des données a été validée à travers le maintien des entrées dans la base de données MariaDB même après le redémarrage des pods. Les tests de qualité de code avec Jest et ESLint ont démontré une bonne couverture et un code maintenable, tandis que le pipeline CI/CD a fonctionné sans erreur, garantissant une intégration continue fluide. Le projet a ainsi permis de consolider nos compétences en conteneurisation, orchestration et automatisation des déploiements.