

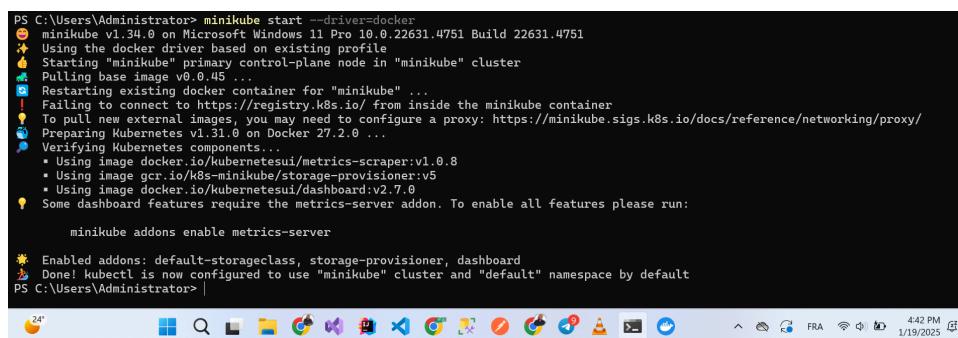
Processus : Intégration Kubernetes avec Helm et GitLab

Ce document décrit les étapes nécessaires pour configurer Minikube, Helm et GitLab CI/CD pour le déploiement d'applications dans un cluster Kubernetes. Les captures d'écran sont incluses pour chaque étape afin d'illustrer le processus.

1. Configuration de Minikube

Dans cette étape, Minikube est configuré pour démarrer avec le driver Docker. La commande utilisée et le résultat sont visibles ci-dessous :

Commande utilisée : `minikube start --driver=docker`



```
PS C:\Users\Administrator> minikube start --driver=docker
minikube v1.34.0 on Microsoft Windows 11 Pro 10.0.22631.4751 Build 22631.4751
Using the docker driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v8.0.45 ...
Restarting existing docker container for "minikube" ...
Failing to connect to https://registry.k8s.io/ from inside the minikube container
To pull new extensions images you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
Verifying Kubernetes components...
  * Using Image docker.io/kubernetesci/metrics-scraperv1.0.8
  * Using Image gcr.io/k8s-minikube/storage-provisioner:v5
  * Using Image docker.io/kubernetesci/dashboard:v2.7.0
Some dashboard features require the metrics-server addon. To enable all features please run:
  minikube addons enable metrics-server
Enabled addons: default-storageclass, storage-provisioner, dashboard
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\Administrator>
```

2. Vérification de l'état de Minikube

Après avoir démarré Minikube, nous vérifions son état pour confirmer que le contrôle plane et les composants associés sont opérationnels. Voici les commandes et résultats :

Commande utilisée : `minikube status`



```
PS C:\Users\Administrator> minikube status
[...]
PS C:\Users\Administrator>
```

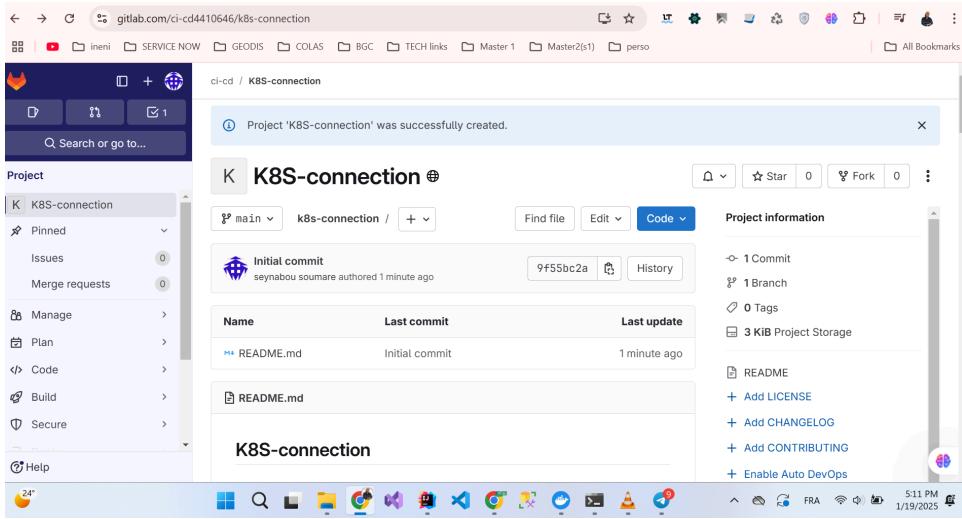
Commande utilisée : `kubectl get nodes`



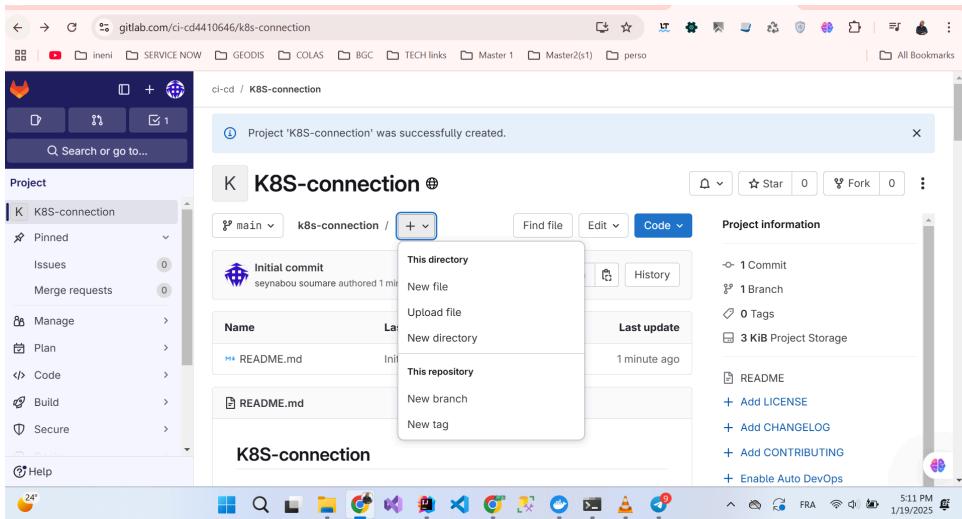
```
PS C:\Users\Administrator> kubectl get nodes
NAME      STATUS    ROLES      AGE     VERSION
minikube  Ready     control-plane   4d21h   v1.31.0
PS C:\Users\Administrator>
```

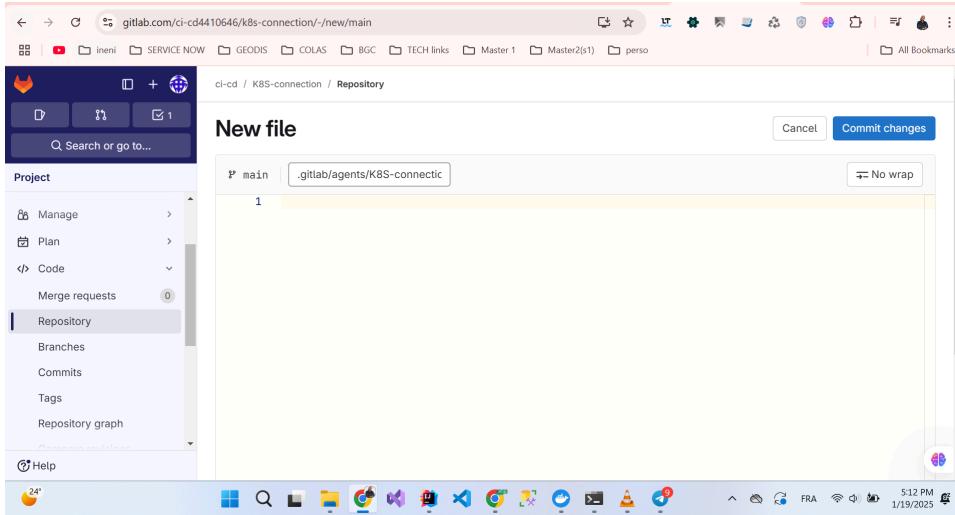
3. Intégration avec GitLab

Pour intégrer Kubernetes avec GitLab, un nouveau projet est créé sur GitLab. Un agent GitLab est configuré pour connecter le cluster Kubernetes. Voici les étapes :



Un fichier de configuration YAML est ajouté au projet GitLab pour lier l'agent Kubernetes au cluster.





4. Déploiement avec Helm

Helm est utilisé pour gérer les configurations et déployer les applications sur le cluster Kubernetes. Voici les commandes exécutées pour ajouter le repo Helm et déployer l'agent GitLab.

Commande Helm utilisée pour installer un agent GitLab sur Kubernetes, configurant un token

A screenshot of a Windows PowerShell window titled "Administrator: Windows Pow". The command executed is "PS C:\Users\Administrator> helm upgrade --install k8s-connection gitlab/gitlab-agent --namespace gitlab-agent-k8s-connection --create-namespace --set config.token=lagent-1k2Z7Gs4sPpqz3J5XKmq6NtsYrmZfLHTBN9uDe7_szk7zqQXY6Q --set config.kasAddress=wss://kas.gitlab.com". The output shows the release "k8s-connection" does not exist and is being installed now. It provides details about the deployment: NAME: k8s-connection, LAST DEPLOYED: Sun Jan 19 17:59:46 2025, NAMESPACE: gitlab-agent-k8s-connection, STATUS: deployed, REVISION: 1, TEST SUITE: None, NOTES: Thank you for installing gitlab-agent. The release is named k8s-connection. A changelog for version 1.17.0 is shown, noting a change in default replica count from 1 to 2. The PowerShell prompt "PS C:\Users\Administrator>" is visible at the bottom.

Tableau de bord GitLab montrant un agent Kubernetes connecté avec son statut en ligne et ses informations.

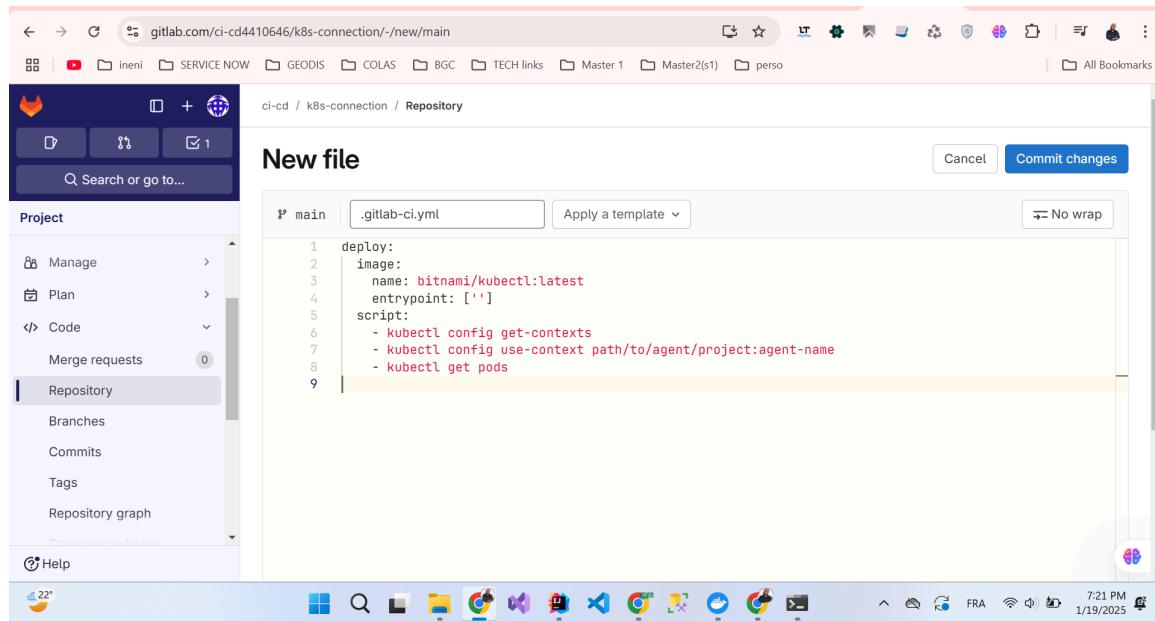
The screenshot shows the GitLab CI/CD interface for a project named "ci-cd / k8s-connection / Kubernetes". On the left, there's a sidebar with "Project" sections like "Secure", "Deploy", "Operate", "Environments", and "Kubernetes clusters". The "Kubernetes clusters" section is currently selected. The main area is titled "Agent" and shows a table for "Project agents". It lists one agent named "k8s-connection" which is "Connected" and was last contacted "just now". The table includes columns for Name, Connection status, Last contact, Version, Agent ID, and Configuration. A "Connect a cluster" button is also visible. The top navigation bar shows the URL "gitlab.com/ci-cd4410646/k8s-connection/-/clusters".

6. Test connection entre Gitlab et K8S

Création d'un nouveau fichier dans le dépôt GitLab pour configurer les interactions avec Kubernetes.

The screenshot shows the GitLab repository interface for the "k8s-connection" project. The user is in the "main" branch and is about to create a new file. A modal dialog is open, showing options to "Add new file" or "Upload file". The "This directory" section shows a "New file" option. The "Last update" section shows a "1 hour ago" entry. The "This repository" section shows "New branch" and "New tag" options. The repository page itself has a heading "k8s-connection" and a "Getting started" section. The top navigation bar shows the URL "gitlab.com/ci-cd4410646/k8s-connection/-/tree/main?ref_type=heads".

Début de l'édition du fichier `.gitlab-ci.yml` pour définir les étapes du pipeline CI/CD.

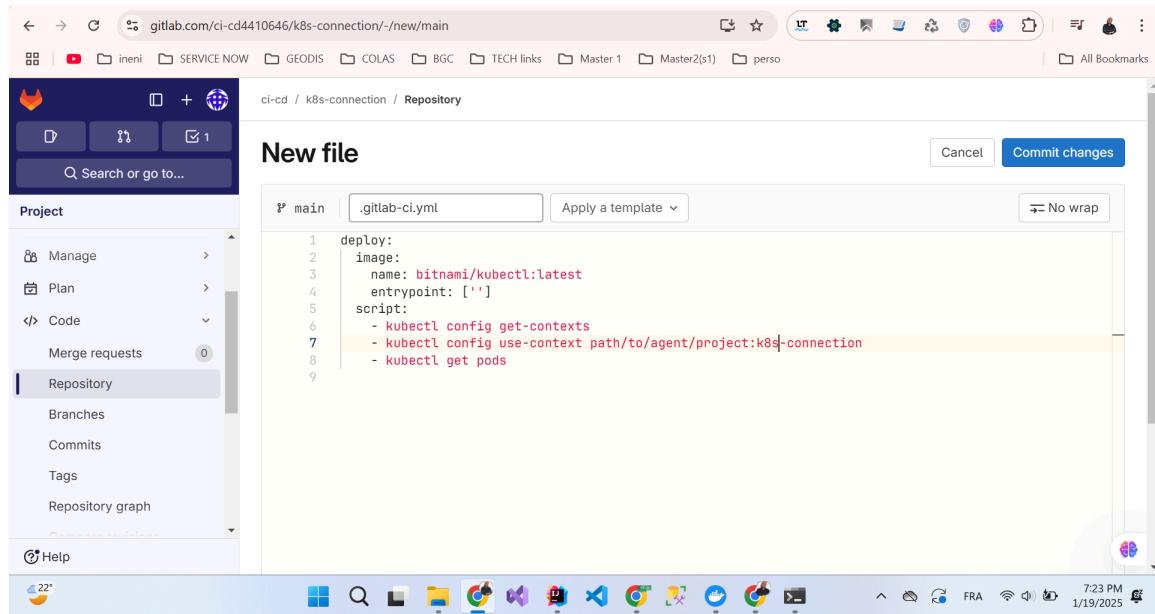


The screenshot shows a web browser window with the URL `gitlab.com/ci-cd4410646/k8s-connection/-/new/main`. The page title is "ci-cd / k8s-connection / Repository". A modal window titled "New file" is open, showing the content of a file named ".gitlab-ci.yml". The code in the file is:

```
main .gitlab-ci.yml Apply a template ▾
1 deploy:
2   image:
3     name: bitnami/kubectl:latest
4     entrypoint: []
5   script:
6     - kubectl config get-contexts
7     - kubectl config use-context path/to/agent/project:agent-name
8     - kubectl get pods
9
```

The "Commit changes" button is visible at the top right of the modal. The browser's address bar and taskbar are visible at the bottom.

Mise à jour du fichier `.gitlab-ci.yml` pour inclure une configuration d'image Docker et des commandes `kubectl`.



The screenshot shows a web browser window with the URL `gitlab.com/ci-cd4410646/k8s-connection/-/new/main`. The page title is "ci-cd / k8s-connection / Repository". A modal window titled "New file" is open, showing the content of a file named ".gitlab-ci.yml". The code in the file is:

```
main .gitlab-ci.yml Apply a template ▾
1 deploy:
2   image:
3     name: bitnami/kubectl:latest
4     entrypoint: []
5   script:
6     - kubectl config get-contexts
7     - kubectl config use-context path/to/agent/project:k8s-connection
8     - kubectl get pods
9
```

The "Commit changes" button is visible at the top right of the modal. The browser's address bar and taskbar are visible at the bottom.

Vue de la configuration validée du fichier `.gitlab-ci.yml` dans le tableau de bord GitLab.

The screenshot shows the GitLab interface for a project named "ci-cd / k8s-connection". A success message at the top states: "The file has been successfully created." Below it, a note says: "This GitLab CI configuration is valid. Learn more". The ".gitlab-ci.yml" file content is displayed in a code editor:

```
1 deploy:
2   image:
3     name: bitnami/kubectl:latest
4     entrypoint: []
5   script:
6     - kubectl config get-contexts
7     - kubectl config use-context path/to/agent/project:k8s-connection
8     - kubectl get pods
```

The file has 216 lines and 8 characters per line. There are buttons for "Edit", "Replace", "Delete", and file operations like copy, download, etc.

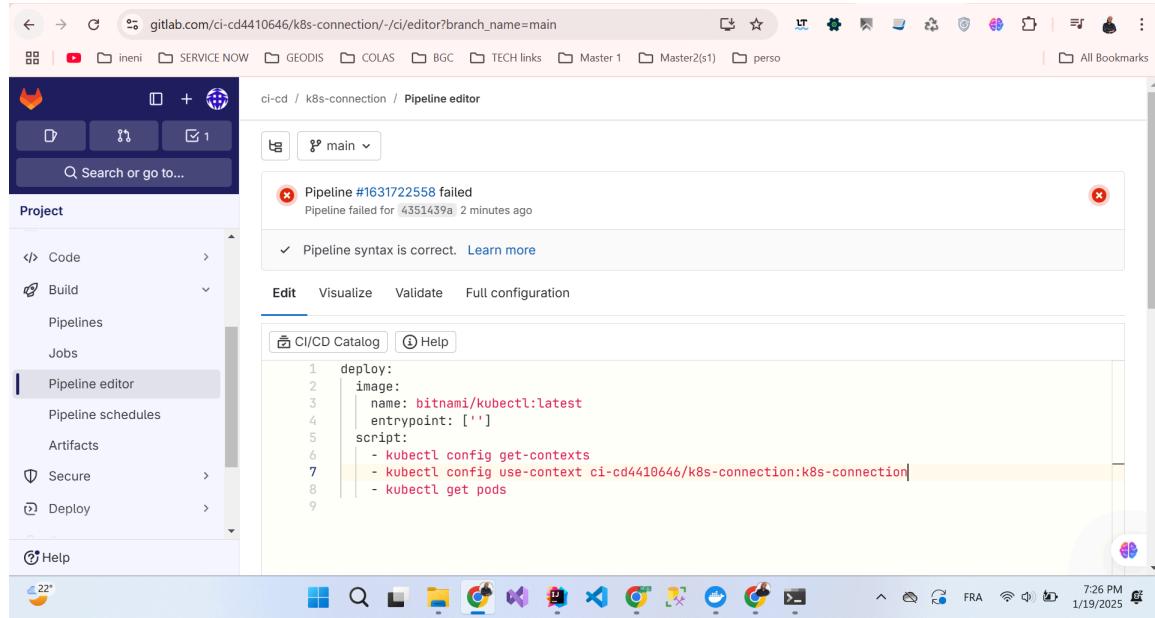
Pipeline CI/CD échoué après une tentative de déploiement à partir du fichier `.gitlab-ci.yml`.

The screenshot shows the GitLab Pipelines page for the same project. It displays a single pipeline entry:

Status	Pipeline	Created by	Stages	Actions
Failed	Add new file ⌚ 00:00:18 ⌚ just now	seynabou soumare	latest	↻ ⏪

The pipeline failed after 00:00:18 and was created just now. The "Actions" column shows a refresh icon and a download icon. The pipeline ID is #1631722558.

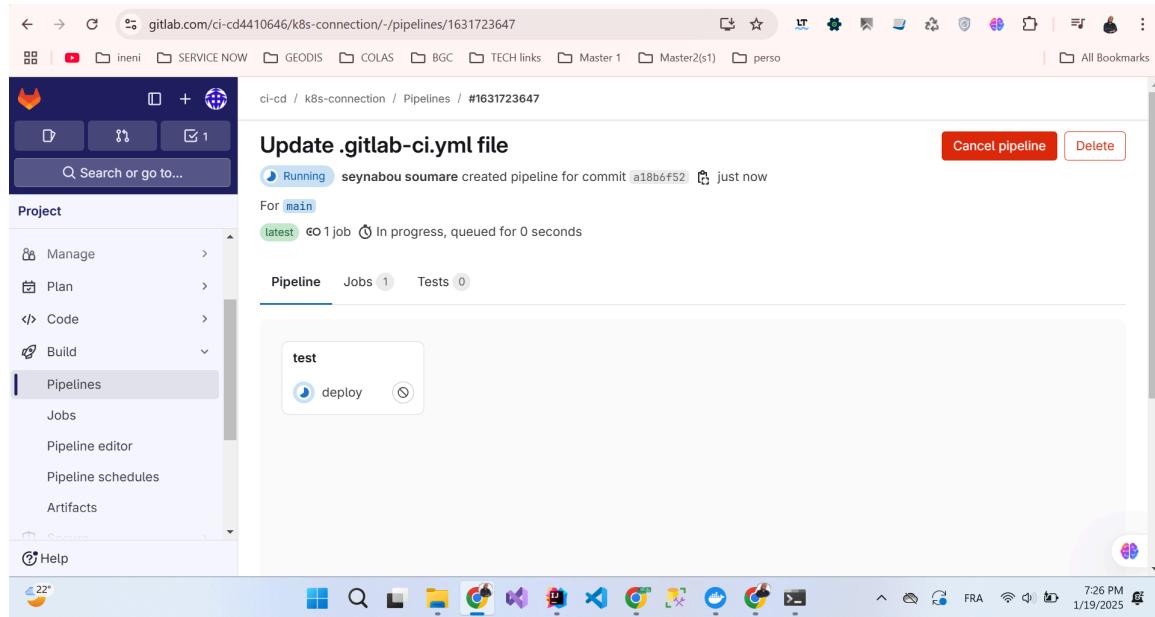
Correction de la configuration dans le fichier `.gitlab-ci.yml` pour ajuster les chemins de contexte Kubernetes.



The screenshot shows the GitLab Pipeline editor interface. On the left, a sidebar lists project sections: Code, Build, Pipelines, Jobs, Pipeline editor (which is selected), Pipeline schedules, Artifacts, Secure, Deploy, and Help. The main area displays a pipeline named "main". A red error message at the top states: "Pipeline #1631722558 failed" and "Pipeline failed for 4351439a 2 minutes ago". Below this, a green success message says "Pipeline syntax is correct." with a "Learn more" link. The "Edit" tab is active. In the code editor, the `.gitlab-ci.yml` file contains the following script:

```
1 deploy:
2   image:
3     name: bitnami/kubectl:latest
4     entrypoint: []
5   script:
6     - kubectl config get-contexts
7     - kubectl config use-context ci-cd4410646/k8s-connection:k8s-connection
8     - kubectl get pods
```

Relancement du pipeline CI/CD, montrant une exécution réussie des étapes configurées.



The screenshot shows the GitLab Pipeline editor interface after a pipeline has been re-run. The sidebar remains the same. The main area shows a pipeline named "main" with a green status bar indicating it is "Running" and was created by "seynabou soumare" for commit "a18b6f52" just now. The pipeline has one job named "test" which is currently "In progress, queued for 0 seconds". The "Jobs" tab is active, showing the job details. The code editor shows the same `.gitlab-ci.yml` file as before. The bottom status bar shows the date and time as "1/19/2025 7:26 PM".

Le job `deploy` utilise un GitLab Runner avec Docker pour exécuter des commandes `kubectl`. Le pipeline est terminé avec succès.

The screenshot shows a browser window for a GitLab pipeline job. The URL is `gitlab.com/ci-cd4410646/k8s-connection/-/jobs/8892412018`. The job is named `deploy` and has passed. The log output shows the following steps:

```
1 Running with gitlab-runner 17.7.0-pre.103.g896916a8 (896916a8)
2 on green-1.saaS-linux-small-amd64.runners-manager.gitlab.com/default JL
gUOpM, system ID: s_deaa2ca09de7
3 Preparing the "docker-machine" executor 00:11
4 Using Docker executor with image bitnami/kubectl:latest ...
5 Pulling docker image sha256:bad64b3870108682d37832d210d5959a606286745e5c7
443a0abf1965ab5b8 for bitnami/kubectl:latest with digest bitnami/kubectl@sha256:c1ad8e399fa68095782dcabfb43720fd419ba889e061e381f35ae1693af9298
...
7 Preparing environment 00:04
8 Running on runner-jlgoupm-project-66288999-concurrent-0 via runner-jlgoupm
pm-s-l-s-amd64-1737314723-3592e7ee...
9 Getting source from Git repository 00:01
10 Fetching changes with git depth set to 20...
11 Initialized empty Git repository in /builds/ci-cd4410646/k8s-connection/.
```

Details on the right side of the interface include:

- Duration: 20 seconds
- Finished: just now
- Queued: 0 seconds
- Timeout: 1h (from project)
- Runner: #12270845 (JLgUOpM) 1-green.saaS-linux-small-amd64.runners-manager.gitlab.com/default
- Commit: a18b6f52
- Update .gitlab-ci.yml file
- Pipeline #1631723647 Passed for main
- Related jobs

Un projet GitLab nommé `k8s-data` est créé avec une visibilité privée.

The screenshot shows a browser window for creating a new GitLab project. The URL is `gitlab.com/projects/new?namespace_id=97317905#blank_project`. The project is titled `Create blank project`. The form fields are as follows:

- Project name:** `k8s-data`
- Project URL:** `https://gitlab.com/ci-cd4410646`
- Project slug:** `k8s-data`
- Project deployment target (optional):** Select the deployment target
- Visibility Level:** Private (selected)
- Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.
- Internal (disabled)
- The project can be accessed by any logged in user except external users.

Le projet **stock-ms** a un commit initial et une branche **main**.

The screenshot shows a GitLab project page for 'microservice_jee_2025'. The main branch is 'main'. The project information sidebar indicates 3 commits, 2 branches, and 0 tags. The code table lists several files and their commit history.

Name	Last commit	Last update
.mvn/wrapper	Initial commit	1 month ago
src	ci-cd 2	1 week ago
.gitattributes	Initial commit	1 month ago
.gitignore	Initial commit	1 month ago
.gitlab-ci.yml	ci-cd 3	1 week ago
Dockerfile	ci-cd	1 week ago

The screenshot shows a GitLab project page for 'microservice_jee_2025'. The 'Deploy' menu item is selected, opening a dropdown menu for Container Registry. The URL at the bottom of the page is https://gitlab.com/ci-cd4410646/backend/microservice_jee_2025/container_registry.

gitlab.com/ci_cd4410646/backend/microservice_jee_2025/container_registry

Container Registry

Cleanup is not scheduled. Container Scanning for Registry: Off

Filter results Updated

?

There are no container images stored for this project

With the Container Registry, every project can have its own space to store its Docker images. [More Information](#)

Project

- Deploy
- Releases
- Feature flags
- Package Registry
- Container Registry
- Model registry
- Pages

Operate Monitor Help

gitlab.com/ci_cd4410646/backend/microservice_jee_2025/container_registry

Container Registry

this project

With the Container Registry, every project can have its own space to store its Docker images. [More Information](#)

CLI Commands

If you are not already logged in, you need to authenticate to the Container Registry by using your GitLab username and password. If you have [Two-Factor Authentication](#) enabled, use a [personal access token](#) instead of a password.

`docker login registry.gitlab.com`

You can add an image to this registry with the following commands:

`docker build -t registry.gitlab.com/ci_cd4410646/backend/micro`

`docker push registry.gitlab.com/ci_cd4410646/backend/micro`

Project

- Deploy
- Releases
- Feature flags
- Package Registry
- Container Registry
- Model registry
- Pages

Operate Monitor Help

ecommerce > Dockerfile

```

Project
  ecommerce C:\Users\Administrator\Documents\ISI\Master 2\Semestre 1\JEE\tp\ecommerce
    > .idea
    > .mvn
    > src
    > target
      .gitattributes
      .gitignore
      Dockerfile
      namespace-devops.yaml
      stock-ms-service.yaml
      stock-ms-deployment.yaml
  Terminal Local + 
PS C:\Users\Administrator\Documents\ISI\Master 2\Semestre 1\JEE\tp\ecommerce> docker login registry.gitlab.com
Authenticating with existing credentials...
Login Succeeded
PS C:\Users\Administrator\Documents\ISI\Master 2\Semestre 1\JEE\tp\ecommerce>

```

ecommerce > Dockerfile

```

PS C:\Users\Administrator\Documents\ISI\Master 2\Semestre 1\JEE\tp\ecommerce> docker build -t registry.gitlab.com/ci-cd4410646/backend/microservice_jee_2025/stock-ms:3.0 .
[+] Building 8.2s (9/9) FINISHED
   docker:desktop-linux
   => [internal] load build definition from Dockerfile
   => => transferring dockerfile: 394B
   => [internal] load metadata for docker.io/library/openjdk:17-jdk-slim
   => [auth] library/openjdk:pull token for registry-1.docker.io
   => [internal] load .dockerignore
   => => transferring context: 2B
   => [1/3] FROM docker.io/library/openjdk:17-jdk-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774
   => => resolve docker.io/library/openjdk:17-jdk-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774
   => [internal] load build context

```

ecommerce > Dockerfile

```

COPY target/stock-ms.jar app.jar
Terminal Local + 
=> => naming to registry.gitlab.com/ci-cd4410646/backend/microservice_jee_2025/stock-ms:3.0
=> => unpacking to registry.gitlab.com/ci-cd4410646/backend/microservice_jee_2025/stock-ms:3.0

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/g13ejomc60d56w5ciq4qfjcn7
PS C:\Users\Administrator\Documents\ISI\Master 2\Semestre 1\JEE\tp\ecommerce> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
registry.gitlab.com/ci-cd4410646/backend/microservice_jee_2025/stock-ms   3.0      bccb0eee8927   38 seconds ago  752MB
<none>              <none>  1c9b2d073c0c   6 days ago   338MB
sante                v1      fd8b03f666ad   6 days ago   338MB
registry.gitlab.com/ci-cd4410646/k8s-data/sante                   v1      fd8b03f666ad   6 days ago   338MB
nabousoum/stock-ms                                3.0      c816580dd2502  6 days ago   752MB
stock-ms              latest   c816580dd2502  6 days ago   752MB
<none>              <none>  b5a3e9615a28  8 days ago   752MB
nginx                latest   fb197595ebe7   2 months ago  278MB
mariadb              11.5.2  2d50fe0f77da   2 months ago  550MB

```

ecommerce > Dockerfile

```
PS C:\Users\Administrator\Documents\ISI\Master 2\Semestre 1\JEE\tp\ecommerce> docker push registry.gitlab.com/ci-cd4410646/backend/microservice_jee_2025/stock-ms:3.0
The push refers to repository [registry.gitlab.com/ci-cd4410646/backend/microservice_jee_2025/stock-ms]
c4878385aa3a: Pushed
6ce99fd16e08: Pushing [>] 2.097MB/187.9MB
695902abbc4c5: Pushed
6aaaeb6275c7: Pushing [=>] 1.049MB/52.17MB
44d3aa8d0766: Pushing [======>] 1.582MB/1.582MB
1fe172e4850f: Pushing [=>] 1.049MB/31.38MB
```

The screenshot shows a web browser window with the URL `gitlab.com/ci-cd4410646/backend/microservice_jee_2025/container_registry`. The page title is "Container Registry". On the left, there is a sidebar with project navigation, including "Pinned" (Issues: 0, Merge requests: 0), "Manage", "Plan", "Code", "Build", "Secure", and "Help". The main content area displays a table for the "microservice_jee_2025/stock-ms" repository. The table has one row with the following columns: "Image repository" (with a link to the repository), "Cleanup is not scheduled. Set up cleanup", "Container Scanning for Registry: Off", "Filter results" (with a search bar and dropdown menu), and "Published 19 minutes ago" (with a trash bin icon). The table also shows "1 tag". At the bottom of the browser window, there is a taskbar with various icons.

The screenshot shows a code editor interface with the following details:

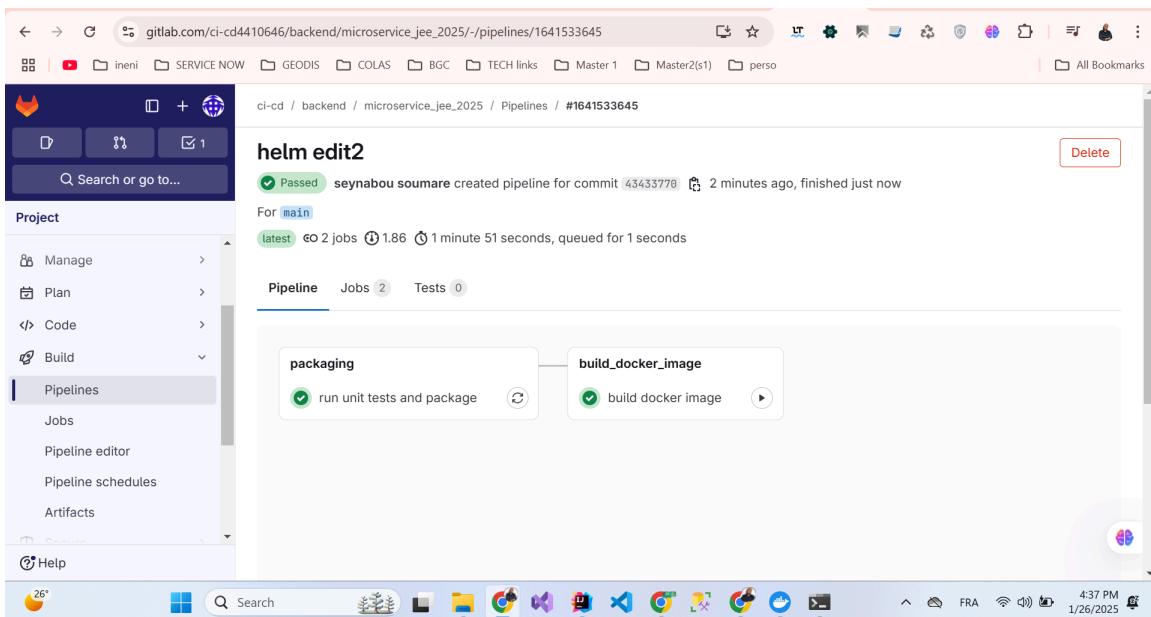
- Project:** ecommerce
- File:** .gitlab-ci.yml
- Content (Excerpt):**

```
build docker image:
script:
  #-- docker save $DOCKER_HUB_USER/stock-ms:0.0.1 > stock-ms.tar
  #-- docker login -u $DOCKER_HUB_USER -p $DOCKER_HUB_TOKEN
  #-- docker push $DOCKER_HUB_USER/stock-ms:0.0.1
  - docker login -u $CI_REGISTRY_USER -p $CI_REGISTRY_PASSWORD $CI_REGISTRY
  - docker build -t $CI_REGISTRY/ci-cd4410646/backend/microservice_jee_2025/stock-ms:3.0
  - docker push $CI_REGISTRY/ci-cd4410646/backend/microservice_jee_2025/stock-ms:3.0
  - echo "Image built"
artifacts:
  paths:
    - stock-ms.tar
when: manual
```

- Terminal:** Local
- Output:**

```
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 770 bytes | 770.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/ci-cd4410646/backend/microservice_jee_2025.git
  b6776e9..de248a5  main --> main
PS C:\Users\Administrator\Documents\ISI\Master 2\Semestre 1\JEE\tp\ecommerce> docker build -t registry.gitlab.com/ci-cd4410646/backend/microservice_jee_2025/stock-ms:3.0 .
```

- System Status:** 42:87 CRLF UTF-8 2 spaces Schema: cl.json
- System Date/Time:** 4:33 PM 1/26/2025



gitlab.com/ci-cd4410646/backend/microservice_jee_2025/container_registry/8251607

ci-cd / backend / microservice_jee_2025 / Container Registry / stock-ms

stock-ms

1 tag 260.36 MiB Cleanup disabled Created Jan 26, 2025 15:59 Last published at Jan 26, 2025 16:36

Filter results Published

1 tag

3.0 260.37 MiB Published 1 minute ago Digest: a10fe6a

7. Creation de HELM chart

```
Administrator: Windows Pow > + \v
PS C:\Users\Administrator> helm create stock-ms
Creating stock-ms
PS C:\Users\Administrator> |
```

```
C:\USERS\ADMINISTRATOR\STOCK-MS
└── charts
└── templates
    └── tests
PS C:\Users\Administrator> |
```

```
Administrator: Windows Pow C:\Users\Administrator\stock-ms\templates> ls

Directory: C:\Users\Administrator\stock-ms\templates

Mode                LastWriteTime       Length Name
----                -----        ---- 
d----
```

Mode	LastWriteTime	Length	Name
d----	1/26/2025 4:39 PM		tests
-a----	1/26/2025 4:39 PM	2385	deployment.yaml
-a----	1/26/2025 4:39 PM	994	hpa.yaml
-a----	1/26/2025 4:39 PM	1091	ingress.yaml
-a----	1/26/2025 4:39 PM	1748	NOTES.txt
-a----	1/26/2025 4:39 PM	364	service.yaml
-a----	1/26/2025 4:39 PM	391	serviceaccount.yaml
-a----	1/26/2025 4:39 PM	1792	_helpers.tpl

```
PS C:\Users\Administrator\stock-ms\templates> |
```

```
Administrator: Windows Pow C:\Users\Administrator> helm install stock-ms-app ./stock-ms
NAME: stock-ms-app
LAST DEPLOYED: Sun Jan 26 17:35:39 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
PS C:\Users\Administrator> |
```

A screenshot of a code editor window titled "templates [Administrator]". The left sidebar shows a tree view with "TEMPLATES" expanded, containing "deployment.yaml", "secret.yaml", and "service.yaml". The main editor area displays the "deployment.yaml" file:

```
! deployment.yaml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   labels:
5     app: stock-ms
6     name: stock-ms
7     namespace: stock
8 spec:
9   replicas: 3
10  selector:
11    matchLabels:
12      app: stock-ms
13  template:
14    metadata:
15      labels:
16        app: stock-ms
17    spec:
18      containers:
19        - image: registry.gitlab.com/ci-cd4410646/backend/microservice_jee_2025/stock-ms:3.0
20          imagePullPolicy: Always
21          name: stock-ms
22          ports:
23            - containerPort: 8080
24          imagePullSecrets:
25            - name: stock-ms-docker-secret
26
```

The status bar at the bottom right shows "Ln 19, Col 91" and "YAML".

A screenshot of a code editor window titled "templates [Administrator]". The left sidebar shows a tree view with "TEMPLATES" expanded, containing "deployment.yaml", "secret.yaml", and "service.yaml". The main editor area displays the "secret.yaml" file:

```
! secret.yaml
1 apiVersion: v1
2 data:
3   .dockerconfigjson: eyJhdXRocI6eyJyZldpc3RyeS5naXRsyWluyZ9tIjp7InVzzXOuYW1IjoibmFib3Vzb3VtIiwk
4 kind: Secret
5 metadata:
6   creationTimestamp: "2025-01-31T19:51:49Z"
7   name: stock-ms-docker-secret
8   namespace: stock
9   resourceVersion: "95515"
10  uid: 2e2d1370-911e-467f-b595-2420c8632224
11 type: kubernetes.io/dockerconfigjson
```

The status bar at the bottom right shows "Ln 7, Col 31 (2 selected)" and "YAML".

```
! service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: stock-ms
5    labels:
6      app: stock-ms
7  spec:
8    type: NodePort
9    ports:
10      - port: 8080
11        targetPort: 8080
12        protocol: TCP
13        name: http
14    selector:
15      app: stock-ms
16
```

```
PS C:\Users\Administrator> kubectl get pods
NAME                               READY   STATUS            RESTARTS   AGE
sante-app-deployment-595cdc69c4-q4mt4   1/1    Running          1 (120m ago) 6d20h
stock-ms-859b789588-c7vrd               0/1    InvalidImageName 0           46s
stock-ms-859b789588-rvxxv               0/1    InvalidImageName 0           46s
PS C:\Users\Administrator> kubectl get deploy
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
sante-app-deployment  1/1     1          1          6d20h
stock-ms      0/2     2          0          71s
PS C:\Users\Administrator>
```

```
PS C:\Users\Administrator> kubectl get svc -n stock
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
stock-ms   NodePort   10.107.79.183   <none>        8080:32042/TCP  9m42s
PS C:\Users\Administrator>
```

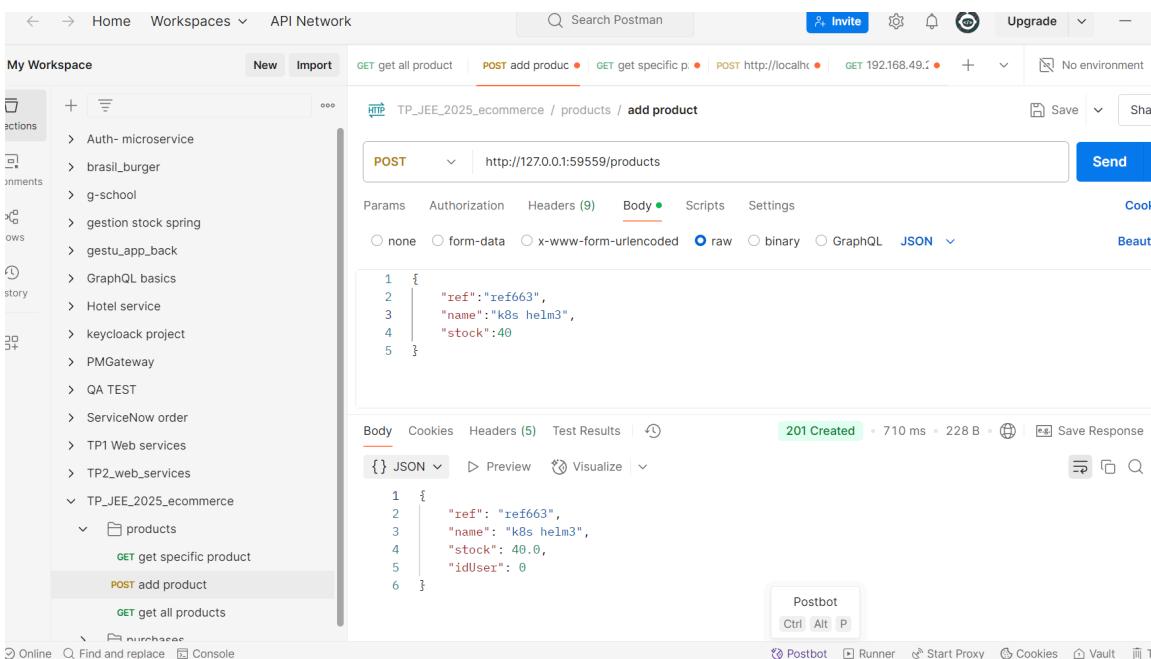
```
NAME      READY   STATUS    RESTARTS   AGE
stock-ms-7d7f975b59-b5tk5   1/1    Running    0          9m13s
stock-ms-7d7f975b59-bm4k7   1/1    Running    0          9m13s
stock-ms-7d7f975b59-srfkk   1/1    Running    0          9m13s
PS C:\Users\Administrator>
```

```
Administrator: Windows Pow PS C:\Users\Administrator> kubectl get services
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)        AGE
kubernetes     ClusterIP  10.96.0.1    <none>       443/TCP       7d4h
sancte-app-service  NodePort   10.96.227.233 <none>       80:30001/TCP  7d2h
stock-ms       ClusterIP  10.106.237.50  <none>       8080/TCP      72s
PS C:\Users\Administrator>
```

```
PS C:\Users\Administrator> kubectl expose deployment stock-ms --type=NodePort --name=stock-ms-service
service/stock-ms-service exposed
PS C:\Users\Administrator>
```

minikube service stock-ms--url

```
PS C:\Users\Administrator> minikube service stock-ms -n stock --url
http://127.0.0.1:59559
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```



The screenshot shows the Postman application interface. On the left, there's a sidebar titled "My Workspace" with a list of projects and environments. The "products" environment is selected. In the main workspace, there's a collection named "TP_JEE_2025_ecommerce" which contains a "products" folder. Inside this folder, there are three items: "GET get specific product", "POST add product" (which is highlighted in grey), and "GET get all products". The "POST add product" item is selected. The "Body" tab is active, showing a JSON payload:

```
{ "ref": "ref663", "name": "k8s helm3", "stock": 40 }
```

Below the body, the response status is shown as "201 Created" with a response time of "710 ms". The response body is also displayed in JSON format:

```
{ "ref": "ref663", "name": "k8s helm3", "stock": 40, "idUser": 0 }
```

docker pull registry.gitlab.com/ci-cd4410646/backend/microservice_jee_2025/stock-ms:3.0