

Sommaire

PARTIE I : Installation.....	1
PARTIE II : Test de Spark avec Spark-Shell.....	Erreur ! Signet non défini.
PARTIE III : L'API de Spark	Erreur ! Signet non défini.
PARTIE IV : Spark Batch en Java	Erreur ! Signet non défini.
PARTIE V : Spark Streaming	Erreur ! Signet non défini.

TP DOCKER K8S

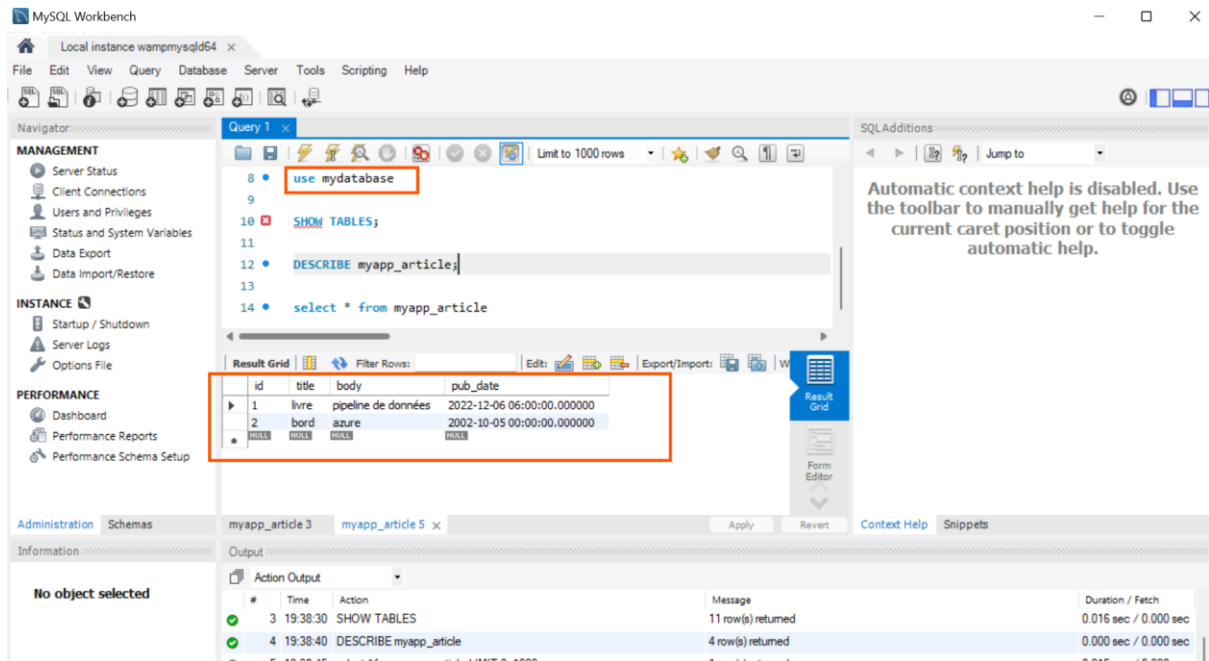
INTRODUCTION

Pipeline:

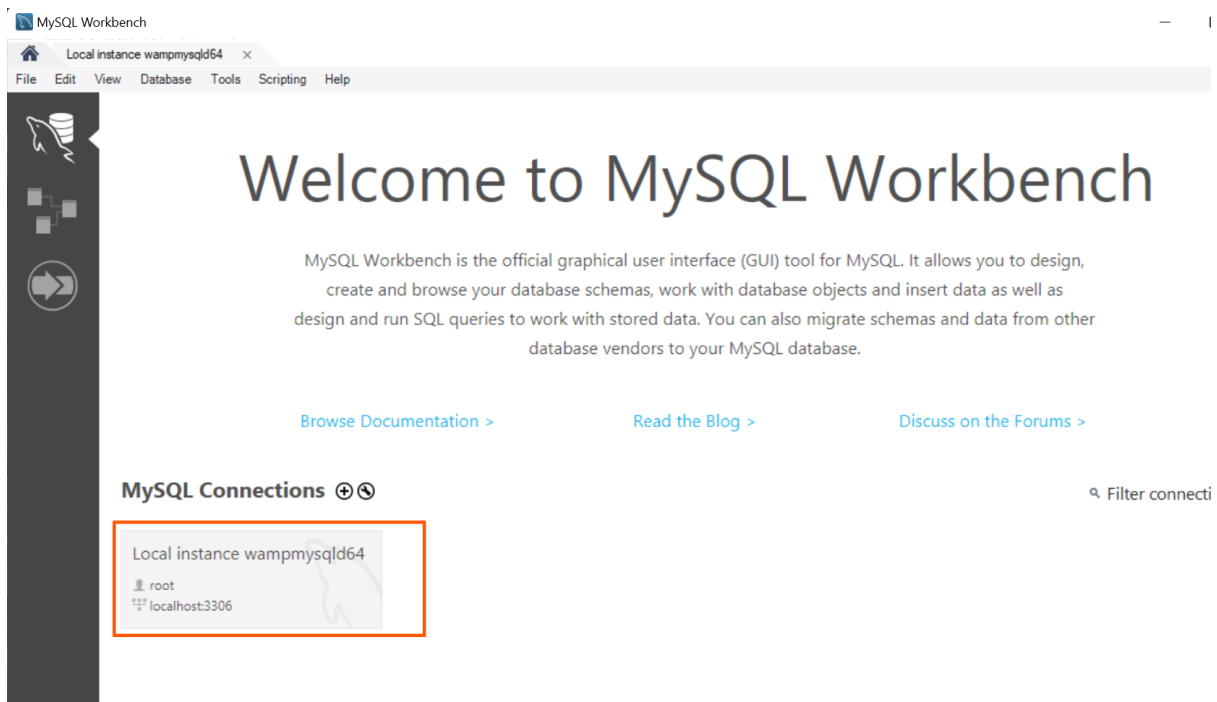
1. **Flume** collecte les données de la source et les envoie à **Kafka**.
2. **Spark Streaming** lit les données de Kafka, les transforme (par exemple, agrégation, filtrage) et les écrit dans **Cassandra**.
3. **Tableau** ou **Power BI** se connecte à Cassandra pour créer des visualisations interactives.

PARTIE I : Étape 1 : Déployer une base de données MySQL comme un Pod

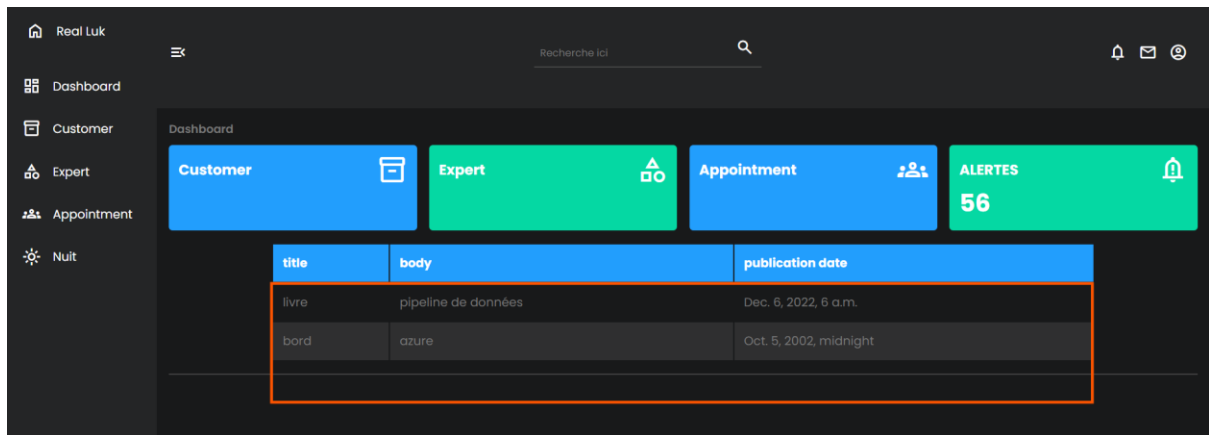
1. creer la bd sur mysqlworkbench



2. le root



3. actualisation sur l'app



1. Créer un fichier YAML pour le déploiement de MySQLion

Preequis : activer kubernestes sur docker

1. Kubernes

```

ations ! https://aka.ms/PSWindows
minikube version
v1.34.0
PS C:\WINDOWS\system32> minikube version
minikube version: v1.34.0
commit: 218b148df93a80eb872ecbe7e35281b3c582c61
PS C:\WINDOWS\system32> kubectl version --client
error: unknown command "version--client" for "kubectl"
PS C:\WINDOWS\system32> kubectl version --client
Client Version: v1.31.0
Kustomize Version: v5.4.2
PS C:\WINDOWS\system32> minikube start
* minikube v1.34.0 sur Microsoft Windows 11 Pro 10.0.22631.4391 Build 22631.4391
* Utilisation du pilote hyperv basé sur le profil existant
* Démarrage du nœud "minikube" primary control-plane dans le cluster "minikube"
* hyperv "minikube" VM est manquant, il va être recréé.
* Création de VM hyperv (CPUs=2, Mémoire=4000MB, Disque=20000MB)...
! Failing to connect to https://registry.k8s.io/ from inside the minikube VM
* Pour extraire de nouvelles images externes, vous devrez peut-être configurer un proxy : https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Préparation de Kubernetes v1.31.0 sur Docker 27.2.0...
  - Génération des certificats et des clés
  - Démarrage du plan de contrôle ...
  - Configuration des règles RBAC ...
* Configuration de bridge CNI (Container Networking Interface)...
* Vérification des composants Kubernetes...
  - Utilisation de l'image gcr.io/k8s-minikube/storage-provisioner:v5
  - Utilisation de l'image docker.io/kubernetes/metrics-scraper:v1.0.8
  - Utilisation de l'image docker.io/kubernetes/dashboard:v2.7.0
* Certaines fonctionnalités du tableau de bord nécessitent le module complémentaire metrics-server. Pour activer toutes les fonctionnalités, veuillez exécuter :
    minikube addons enable metrics-server
* Modules activés: storage-provisioner, dashboard, default-storageclass
* Terminé ! kubectl est maintenant configuré pour utiliser "minikube" cluster et espace de noms "default" par défaut.
PS C:\WINDOWS\system32> kubectl get nodes
NAME       STATUS   ROLES    AGE   VERSION
minikube   Ready    control-plane  33s   v1.31.0
PS C:\WINDOWS\system32> minikube dashboard
* Vérification de l'état du tableau de bord...
* Lancement du proxy...
* Vérification de l'état du proxy...
* Ouverture de http://127.0.0.1:58456/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ dans votre navigateur par défaut...

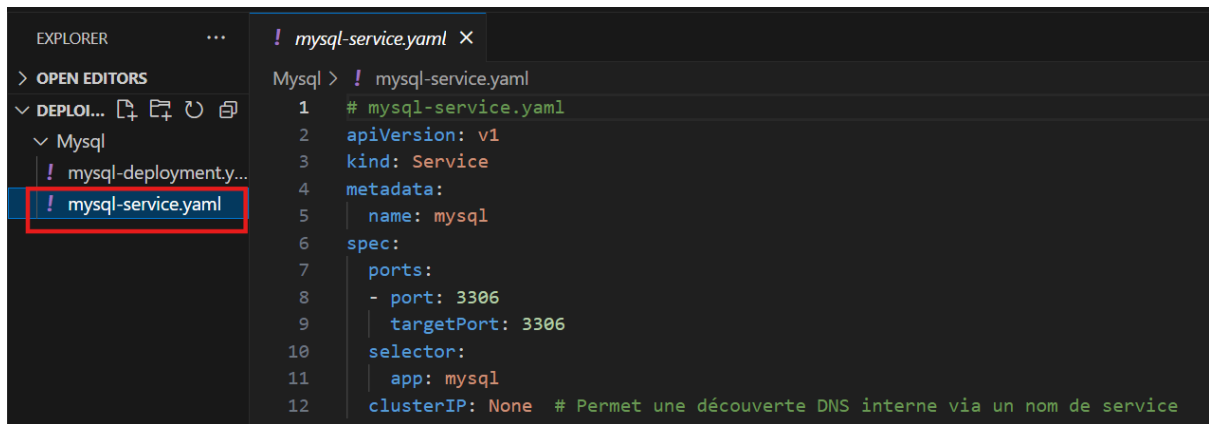
```

2. Créer un fichier YAML pour le déploiement de MySQL

The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORER' sidebar displays a file tree with 'DEPLOYMENTS' expanded, showing 'Mysql' and two files: 'mysql-deployment.y...' (highlighted with a red box) and 'mysql-service.yaml'. The main editor area shows the content of 'mysql-deployment.yaml', which is a Kubernetes Deployment manifest for MySQL. The manifest includes fields for apiVersion, kind, metadata (name: mysql), spec (selector, replicas: 1, template), and a list of environment variables for database credentials. The file is 30 lines long.

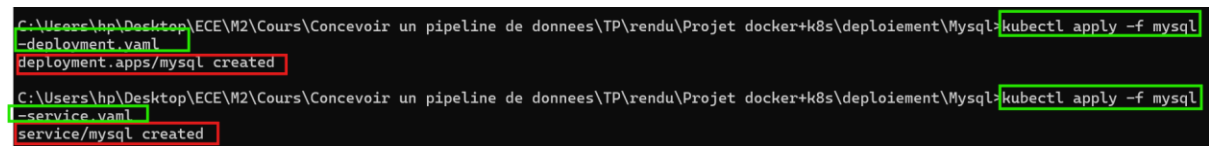
```
1 # mysql-deployment.yaml
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: mysql
6 spec:
7   selector:
8     matchLabels:
9       app: mysql
10  replicas: 1
11  template:
12    metadata:
13      labels:
14        app: mysql
15    spec:
16      containers:
17        - name: mysql
18          image: mysql:5.7
19          env:
20            - name: MYSQL_ROOT_PASSWORD
21              value: "root"
22            - name: MYSQL_DATABASE
23              value: "mydb"
24            - name: MYSQL_USER
25              value: "edson"
26            - name: MYSQL_PASSWORD
27              value: "123"
28          ports:
29            - containerPort: 3306
30
```

3. Créer un fichier YAML pour le service MySQL



```
1 # mysql-service.yaml
2 apiVersion: v1
3 kind: Service
4 metadata:
5   name: mysql
6 spec:
7   ports:
8     - port: 3306
9     targetPort: 3306
10  selector:
11    app: mysql
12  clusterIP: None # Permet une découverte DNS interne via un nom de service
```

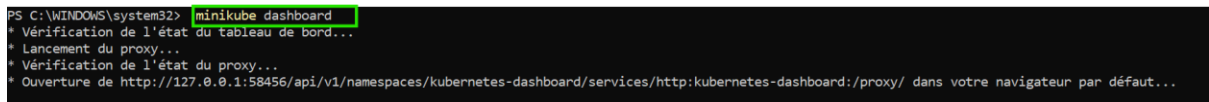
4. Appliquer ces fichiers



```
C:\Users\hp\Desktop\ECE\M2\Cours\Concevoir un pipeline de donnees\TP\rendu\Projet docker+k8s\deploiement\Mysql>kubectl apply -f mysql-deployment.yaml
deployment.apps/mysql created

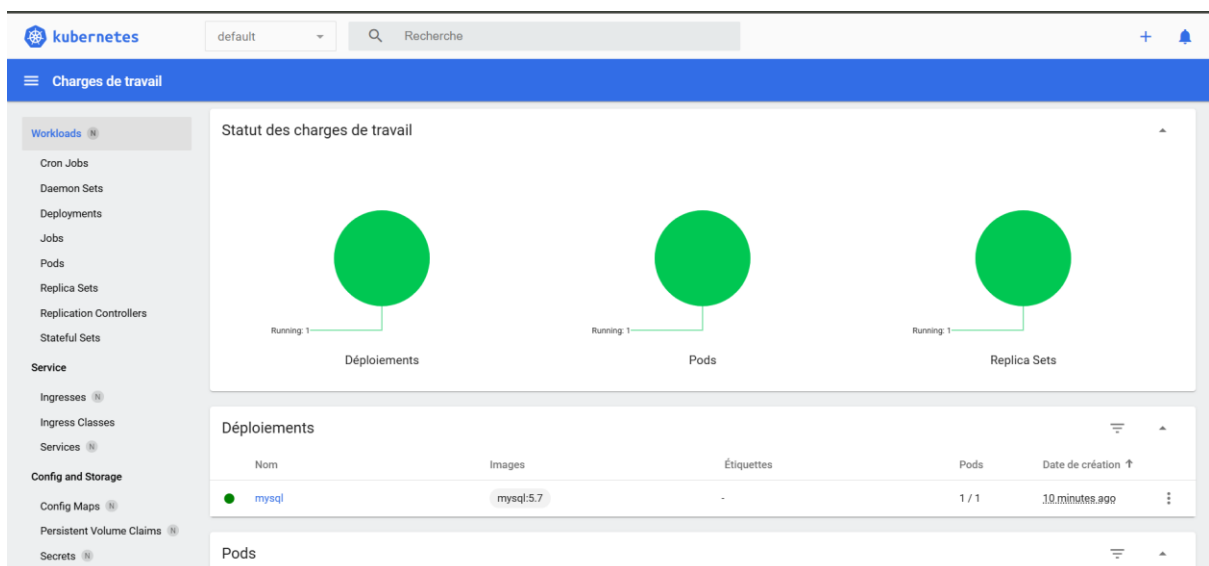
C:\Users\hp\Desktop\ECE\M2\Cours\Concevoir un pipeline de donnees\TP\rendu\Projet docker+k8s\deploiement\Mysql>kubectl apply -f mysql-service.yaml
service/mysql created
```

5. Lancer le dashboard pour vérifier



```
PS C:\WINDOWS\system32> minikube dashboard
* Vérification de l'état du tableau de bord...
* Lancement du proxy...
* Vérification de l'état du proxy...
* Ouverture de http://127.0.0.1:58456/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ dans votre navigateur par défaut...
```

6. dasborad



7. POD

Cron Jobs	Déploiements
Daemon Sets	
Deployments	
Jobs	
Pods	
Replica Sets	
Replication Controllers	
Stateful Sets	
Service	
Ingresses	
Ingress Classes	
Services	
Config and Storage	
Config Maps	
Persistent Volume Claims	
Secrets	

Nom	Images	Étiquettes	Pods	Date de création ↑
mysql	mysql:5.7	-	1 / 1	11 minutes ago

Nom	Images	Étiquettes	Noeud	Statut	Redémar	Utilisation CPU (coeurs)	Utilisation mémoire (octets)	Date de création ↑
mysql-6bd8876d97-zvkzq	mysql:5.7	app: mysql pod-template-hash: 6bd8876d97	minikube	Running	0	-	-	11 minutes ago

Nom	Images	Étiquettes	Pods	Date de création ↑
mysql-6bd8876d97	mysql:5.7	app: mysql pod-template-hash: 6bd8876d97	1 / 1	11 minutes ago

8. Vérifier en ligne de commande les POD

```
C:\Users\hp\Desktop\ECE\M2\Cours\Concevoir un pipeline de donnees\TP\rendu\Projet docker+k8s\deploiement\Mysql>kubectl get pods
NAME                                READY  STATUS   RESTARTS  AGE
mysql-6bd8876d97-zvkzq             1/1    Running  0          2m32s
```

9. Description

```
C:\Users\hp\Desktop\ECE\M2\Cours\Concevoir un pipeline de donnees\TP\rendu\Projet docker+k8s\deploiement\Mysql>kubectl describe pod m
mysql-6bd8876d97-zvkzq
Name:          mysql-6bd8876d97-zvkzq
Namespace:     default
Priority:       0
Service Account: default
Node:          minikube/172.20.109.76
Start Time:    Sun, 27 Oct 2024 14:46:43 +0100
Labels:        app=mysql
               pod-template-hash=6bd8876d97
Annotations:   <none>
Status:        Running
IP:            10.244.0.6
IPs:           IP: 10.244.0.6
Controlled By: ReplicaSet/mysql-6bd8876d97
```

PARTIE II : Étape 2 : L'image vers docker Hub

1. Se connecter

```
PS C:\WINDOWS\system32> docker login
Authenticating with existing credentials...
Login Succeeded
```

2. Vérifier les images

```
PS C:\WINDOWS\system32> docker images
```

REPOSITORY	IMAGE ID	TAG	CREATED	SIZE
projetdjangokubernetes	e9d5d54f35da	latest	37 hours ago	480MB
tutorial-environment-app	0f3ffd336cfc	latest	3 weeks ago	25.1MB
cassandra	d9219aec7f75	latest	4 weeks ago	378MB
monprojetdjango	203e4729dcb6	latest	4 weeks ago	159MB
liliasfaxi/hadoop-cluster	b90f134f6fcd	latest	2 months ago	4.92GB
apache/kafka-native		3.8.0		

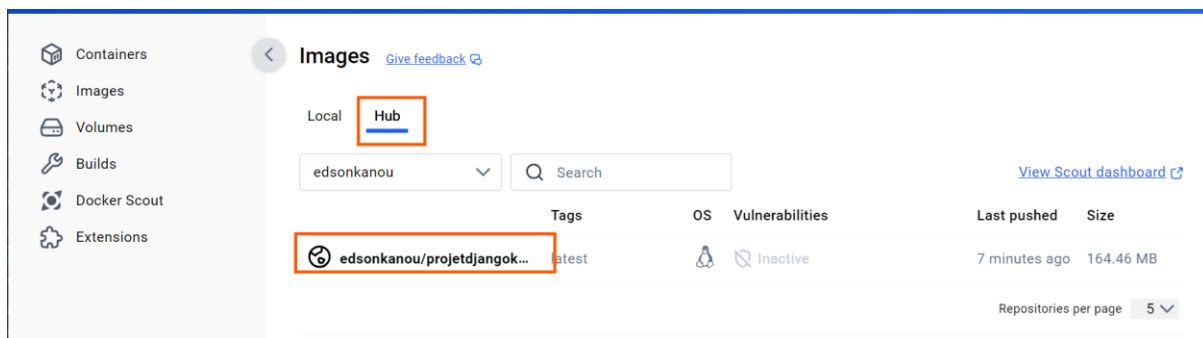
3. Ajouter le tag

```
PS C:\WINDOWS\system32> docker tag projetdjangokubernetes edsonkanou/projetdjangokubernetes:latest
```

4. Push

```
PS C:\WINDOWS\system32> docker push edsonkanou/projetdjangokubernetes:latest
The push refers to repository [docker.io/edsonkanou/projetdjangokubernetes]
39c602b2e001: Pushed
5eb07ca874e5: Pushed
a98e512aeb31: Pushed
4afb9de30d8d: Pushed
05964ac61a9a: Pushed
865cca643298: Mounted from library/python
317c644d5100: Mounted from library/python
835db2ba7b13: Mounted from library/python
98b5f35ea9d3: Mounted from library/python
latest: digest: sha256:1e30ae3fb066b59907b279800881817d3d8690c9161c84fdacd879f3f8808a55 size: 2206
PS C:\WINDOWS\system32>
```

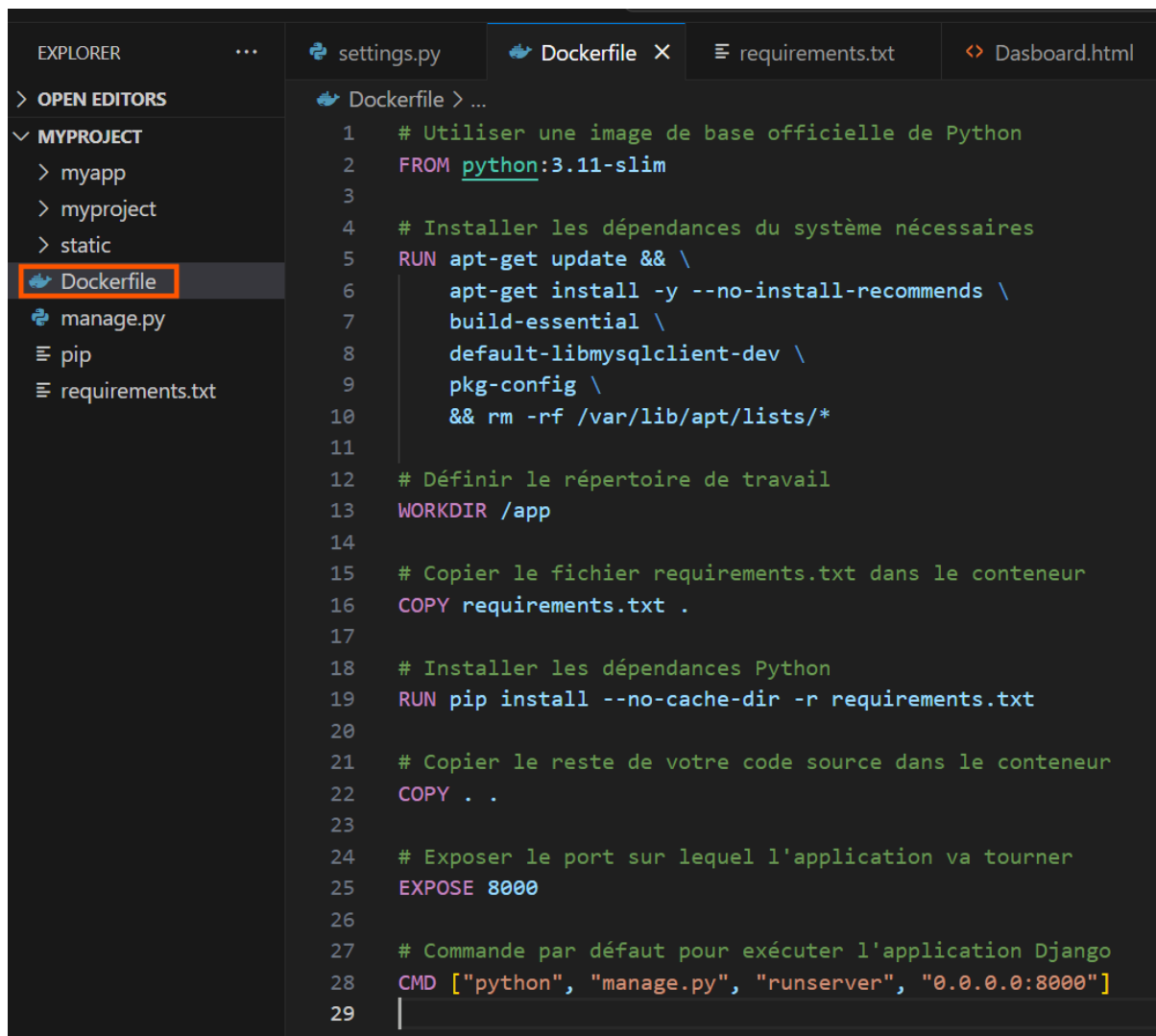
5. Vérifier dans docskeotr desktop si l'operation est ok



PARTIE I : Étape 2 : Déployer une application web Django

L'application Django doit être contenue dans un conteneur Docker avant de pouvoir être déployée dans Kubernetes.

1. Créer Dockerfile pour Django



```
1  # Utiliser une image de base officielle de Python
2  FROM python:3.11-slim
3
4  # Installer les dépendances du système nécessaires
5  RUN apt-get update && \
6      apt-get install -y --no-install-recommends \
7          build-essential \
8          default-libmysqlclient-dev \
9          pkg-config \
10         && rm -rf /var/lib/apt/lists/*
11
12 # Définir le répertoire de travail
13 WORKDIR /app
14
15 # Copier le fichier requirements.txt dans le conteneur
16 COPY requirements.txt .
17
18 # Installer les dépendances Python
19 RUN pip install --no-cache-dir -r requirements.txt
20
21 # Copier le reste de votre code source dans le conteneur
22 COPY . .
23
24 # Exposer le port sur lequel l'application va tourner
25 EXPOSE 8000
26
27 # Commande par défaut pour exécuter l'application Django
28 CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
29
```

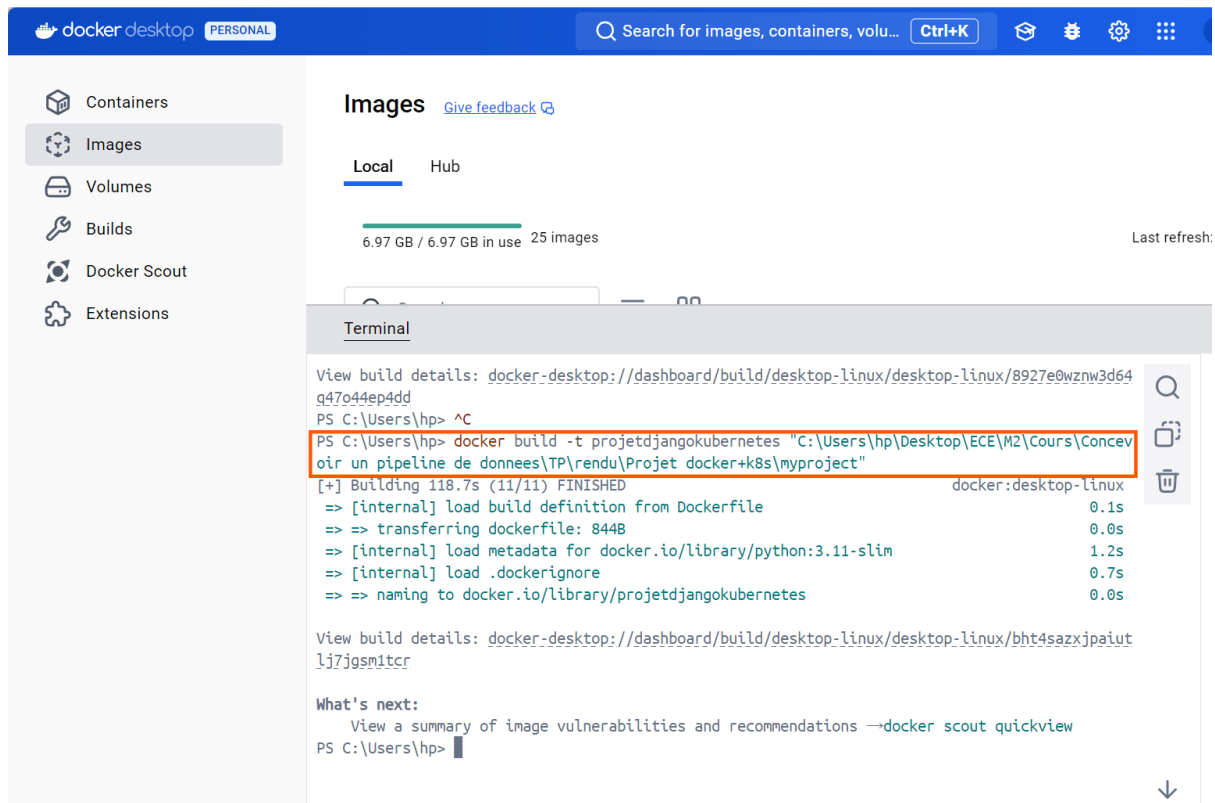
2. Construction de l'installation (build)

Construction de l'image :

- Ouvrez votre interface graphique Docker.
- Utilisez la commande RUN pour exécuter le Dockerfile et créer l'image.
- Spécifiez le chemin vers votre Dockerfile. •

Donnez un nom à votre image (par exemple, projetdjankubernetes).

docker build -t projetdjankubernetes C:\Users\hp\Desktop\ECE\M2\Cours\Concevoir un pipeline de donnees\TP\rendu\Projet docker+k8s\myproject



3. Exécution de l'image dans un conteneur(Run)

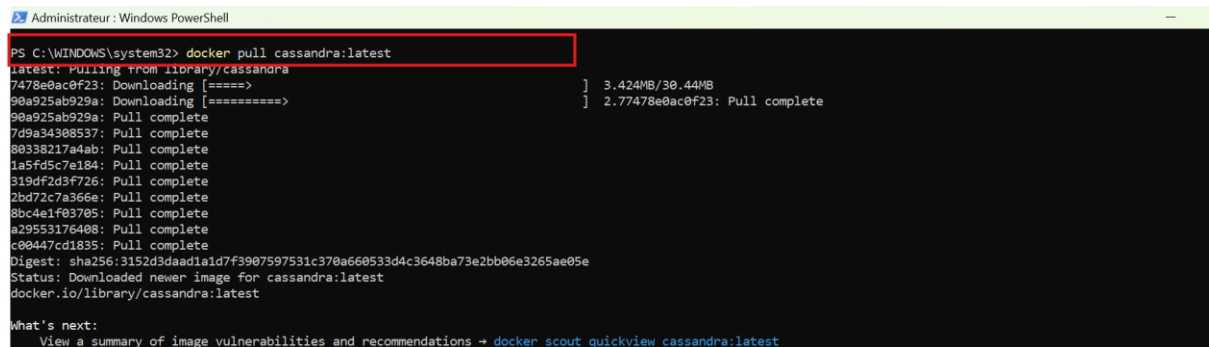
`docker run -p 8000:8000 projetdjankubernetes`

PARTIE I :CASSANDRA

Installation et configurations (docker)

1. Extrayez l'image Docker

Cette docker pull commande obtiendra la dernière version de l'image officielle Docker Apache Cassandra disponible sur [Dockerhub](https://hub.docker.com/_/cassandra/) .



```
Administrateur : Windows PowerShell
PS C:\WINDOWS\system32> docker pull cassandra:latest
latest: Pulling from library/cassandra
7478e0ac0f23: Downloading [====>] 3.424MB/30.44MB
90a925ab929a: Downloading [=====>] 2.77478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
319df2d3f726: Pull complete
2bd72c7a366e: Pull complete
8bc4e1f03705: Pull complete
a29553176408: Pull complete
c00447cd1835: Pull complete
Digest: sha256:3152d3daad1a1d7f3907597531c370a660533d4c3648ba73e2bb06e3265ae05e
Status: Downloaded newer image for cassandra:latest
docker.io/library/cassandra:latest

What's next:
View a summary of image vulnerabilities and recommendations -> docker scout quickview cassandra:latest
```

2. Démarrez Cassandra

```

Administrateur: Windows PowerShell
PS C:\WINDOWS\system32> docker run --name cass_cluster cassandra:latest
CompileCommand: dontinline org/apache/cassandra/db/Columns$Serializer.deserializeLargeSubset(Lorg/apache/cassandra/io/util/DataInputPlus;Lorg/apache/cassandra/db/Columns;I)
org/apache/cassandra/db/Columns; bool dontinline = true
CompileCommand: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeSubset(Ljava/util/Collection;Lorg/apache/cassandra/db/Columns;I)org/apache/cassandra/io
/util/DataOutputPlus;V bool dontinline = true
CompileCommand: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeSubsetSize(Ljava/util/Collection;Lorg/apache/cassandra/db/Columns;I)I bool dontinline =
true
CompileCommand: dontinline org/apache/cassandra/db/commitlog/AbstractCommitLogSegmentManager.advanceAllocatingFrom(Lorg/apache/cassandra/db/commitlog/CommitLogSegment;V) bo
ol dontinline = true
CompileCommand: dontinline org/apache/cassandra/db/transform/BaseIterator.tryGetMoreContents()Z bool dontinline = true
CompileCommand: dontinline org/apache/cassandra/db/transform/StoppingTransformation.stop()V bool dontinline = true
CompileCommand: dontinline org/apache/cassandra/db/transform/StoppingTransformation.stopInPartition()V bool dontinline = true
CompileCommand: dontinline org/apache/cassandra/io/util/BufferedDataOutputStreamPlus.doFlush(I)V bool dontinline = true
CompileCommand: dontinline org/apache/cassandra/io/util/BufferedDataOutputStreamPlus.writeSlow(JI)V bool dontinline = true
CompileCommand: dontinline org/apache/cassandra/io/util/RebufferingInputStream.readPrimitiveSlowly(I)J bool dontinline = true
CompileCommand: exclude org/apache/cassandra/unsafe/UnsafeStabilityInspector.forceHeapSpaceOomMaybe(Ljava/lang/OutOfMemoryError;V) bool exclude = true
CompileCommand: inline org/apache/cassandra/db/NativeDecoratedKey.address()J bool inline = true
CompileCommand: inline org/apache/cassandra/db/NativeDecoratedKey.length()I bool inline = true
CompileCommand: inline org/apache/cassandra/db/rows/UnfilteredSerializer.serializeRowBody(Lorg/apache/cassandra/db/rows/Row;Lorg/apache/cassandra/db/rows/SerializationHelp
er;Lorg/apache/cassandra/io/util/DataOutputPlus;V) bool inline = true
CompileCommand: inline org/apache/cassandra/io/util/Memory.checkBounds(I)V bool inline = true

```

3. Démarrez le shell CQL cqlsh

```

PS C:\WINDOWS\system32> docker exec -it cass_cluster cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.2.0 | Cassandra 5.0.1 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
cqlsh>

```

DOCKER BON DOC

```

PS C:\Users\hp\Desktop\ECE\M2\Cours\Conteneurisation et docker\TP\rendu\DOCKER_KUBERNETES\Django\myproject> docker build -t django-app
p:latest .
[+] Building 7.0s (11/11) FINISHED                                docker:desktop-linux
=> [internal] load build definition from dockerfile                0.3s
=> => transferring dockerfile: 205B                                0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim 2.5s
=> [auth] library/python:pull token for registry-1.docker.io      0.0s
=> [internal] load .dockerignore                                  0.4s
=> => transferring context: 2B                                       0.0s
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:6250eb7983c08b3cf5a7db9309f8630d3ca03dd152158fa37a3f8daaf397085d 0.3s
=> => resolve docker.io/library/python:3.9-slim@sha256:6250eb7983c08b3cf5a7db9309f8630d3ca03dd152158fa37a3f8daaf397085d 0.3s
=> [internal] load build context                                  0.3s
=> => transferring context: 5.66kB                                    0.0s
=> CACHED [2/5] WORKDIR /app                                       0.0s
=> CACHED [3/5] COPY requirements.txt .                            0.0s
=> CACHED [4/5] RUN pip install -r requirements.txt               0.0s
=> [5/5] COPY . .                                                 1.3s
=> exporting to image                                             0.7s
=> => exporting layers                                              0.1s
=> => writing image sha256:f2ec6638c7166f12fc95fd7f092f489abc7173619cb760dafa435ce4119a457a 0.5s
=> => transferring context: 2B                                       0.0s
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:6250eb7983c08b3cf5a7db9309f8630d3ca03dd152158fa37a3f8daaf397085d 0.3s
=> => resolve docker.io/library/python:3.9-slim@sha256:6250eb7983c08b3cf5a7db9309f8630d3ca03dd152158fa37a3f8daaf397085d 0.3s

```

```

PS C:\Users\hp\Desktop\ECE\M2\Cours\Conteneurisation et docker\TP\rendu\DOCKER_KUBERNETES\Django\myproject> docker tag django-app:latest edsonkanou/django-app:latest

```

```

PS C:\Users\hp\Desktop\ECE\M2\Cours\Conteneurisation et docker\TP\rendu\DOCKER_KUBERNETES\Django\myproject> docker push edsonkanou/dj
ango-app:latest
The push refers to repository [docker.io/edsonkanou/django-app]
f5ee6c1add1b: Pushed
e4d3d5675ac3: Pushed
84837b5a7236: Pushed
ed05802f9aca: Pushed
aacba17e24d9: Mounted from library/python
f751ad7c65c4: Mounted from library/python
7822e749b484: Mounted from library/python

```

New :

```

C:\Users\hp> minikube start --driver=docker
* minikube v1.34.0 sur Microsoft Windows 11 Pro 10.0.22631.4391 Build 22631.4391
* Utilisation du pilote docker basé sur le profil existant
* Démarrage du nœud "minikube" primary control-plane dans le cluster "minikube"
* Extraction de l'image de base v0.0.45...
* Redémarrage du docker container existant pour "minikube" ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* Pour extraire de nouvelles images externes, vous devrez peut-être configurer un proxy : https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Préparation de Kubernetes v1.31.0 sur Docker 27.2.0...
* Vérification des composants Kubernetes...
  - Utilisation de l'image gcr.io/k8s-minikube/storage-provisioner:v5
  - Utilisation de l'image docker.io/kubernetesui/dashboard:v2.7.0
  - Utilisation de l'image docker.io/kubernetesui/metrics-scraper:v1.0.8
* Certaines fonctionnalités du tableau de bord nécessitent le module complémentaire metrics-server. Pour activer toutes les fonctionnalités, veuillez exécuter :

    minikube addons enable metrics-server

* Modules activés: storage-provisioner, dashboard, default-storageclass
* Terminé ! kubectl est maintenant configuré pour utiliser "minikube" cluster et espace de noms "default" par défaut.

```

```

C:\Users\hp> minikube dashboard
* Vérification de l'état du tableau de bord...
* Lancement du proxy...
* Vérification de l'état du proxy...
* Ouverture de http://127.0.0.1:57218/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ dans votre navigateur par défaut...

```

```

C:\Users\hp\Desktop\ECE\M2\Cours\Conteneurisation et docker\TP\rendu\DOCKER_KUBERNETES\Deploiements\Postgres> kubectl apply -f postgres-pod.yaml
pod/postgres created

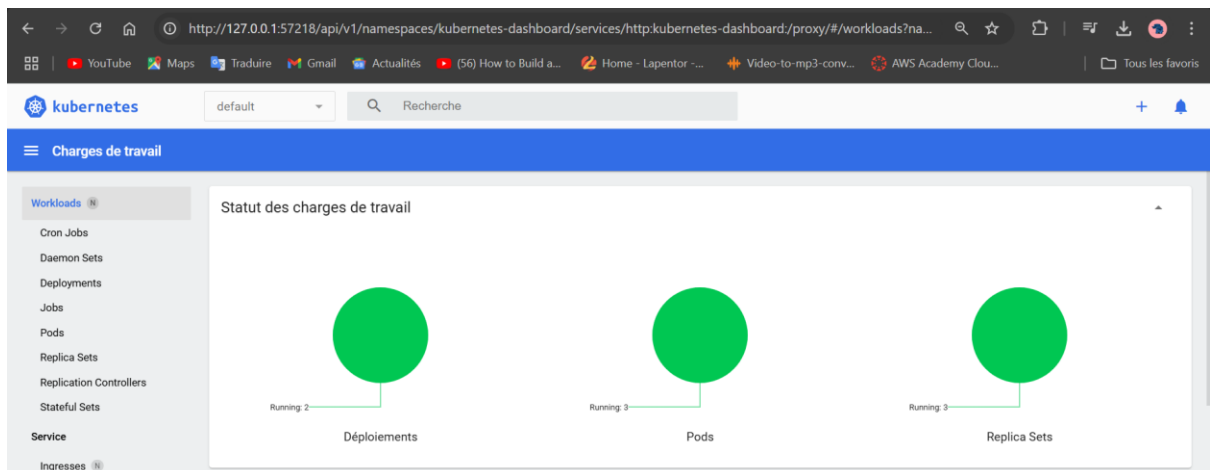
```

```
C:\Users\hp\Desktop\ECE\M2\Cours\Conteneurisation et docker\TP\rendu\DOCKER_KUBERNETES\Deploiements\Postgres>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
django-app-5455dffc6b-7vpgs        1/1     Running   2 (3h38m ago)  3d20h
django-app-5455dffc6b-sc2j         1/1     Running   2 (3h38m ago)  3d20h
mysql-6d9dbb585b-pm8z7             1/1     Running   2 (3h38m ago)  3d20h
postgres                           0/1     ContainerCreating 0          44s
```

```
C:\Users\hp\Desktop\ECE\M2\Cours\Conteneurisation et docker\TP\rendu\DOCKER_KUBERNETES\Deploiements\Django>kubectl apply -f django-deployment.yaml
deployment.apps/django-app configured
```

```
C:\Users\hp\Desktop\ECE\M2\Cours\Conteneurisation et docker\TP\rendu\DOCKER_KUBERNETES\Deploiements\Django>kubectl get deployments
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
django-app  1/1     1            1           3d21h
mysql    1/1     1            1           3d21h
```

```
C:\Users\hp\Desktop\ECE\M2\Cours\Conteneurisation et docker\TP\rendu\DOCKER_KUBERNETES\Deploiements\Django>kubectl apply -f django-service.yaml
service/django-service configured
```



Nom	Images	Étiquettes	Noeud	Statut	Redéman	Utilisation CPU (coeurs)	Utilisation mémoire (octets)	Date de création
django-app-7b7d76bd46-wp4bt	edsonkanou/django-app:latest	app: django pod-template-hash: 7b7d76bd46	minikube	Running	0	-	-	54 minutes ago
postgres	postgres:15	app: postgres	minikube	Running	0	-	-	2 hours ago
mysql-6d9dbb585b-pm8z7	mysql:8.0	app: mysql pod-template-hash: 6d9dbb585b	minikube	Running	2	-	-	3 days ago

```
C:\Users\hp>kubectl get services
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
django-service      NodePort    10.108.78.151 <none>         80:30007/TCP     3d23h
kubernetes           ClusterIP   10.96.0.1     <none>         443/TCP          3d23h
mysql                ClusterIP   10.97.5.202   <none>         3306/TCP         3d23h
postgres-service    ClusterIP   10.105.41.164 <none>         5432/TCP         108m
```

```
C:\Users\hp>kubectl get pods
NAME                READY    STATUS    RESTARTS    AGE
django-app-7b7d76bd46-wg4bt 1/1      Running   0            56m
mysql-6d9d9bb585b-pm8z7      1/1      Running   2 (5h46m ago) 3d23h
postgres              1/1      Running   0            129m
```

Activer Windows

```
C:\Users\hp>kubectl exec -it django-app-7b7d76bd46-wg4bt -- /bin/bash
```

```
C:\Users\hp>kubectl exec -it django-app-7b7d76bd46-lkn4m -- /bin/bash
root@django-app-7b7d76bd46-lkn4m:/app# python manage.py migrate
System check identified some issues:

WARNINGS:
?: (urls.W002) Your URL pattern '/details' [name='details'] has a route beginning with a '/'. Remove this slash as it is unnecessary.
If this pattern is targeted in an include(), ensure the include() pattern has a trailing '/'.
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, shop
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
  Applying shop.0001_initial... OK
root@django-app-7b7d76bd46-lkn4m:/app# python manage.py createsuperuser
System check identified some issues:
```

Activer Windows

```
root@django-app-7b7d76bd46-lkn4m:/app# python manage.py createsuperuser
System check identified some issues:

WARNINGS:
?: (urls.W002) Your URL pattern '/details' [name='details'] has a route beginning with a '/'. Remove this slash as it is unnecessary.
If this pattern is targeted in an include(), ensure the include() pattern has a trailing '/'.
Username (leave blank to use 'root'): edson
Email address:
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
root@django-app-7b7d76bd46-lkn4m:/app# python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:
```

Vérifier les erreurs

```
C:\Users\hp>kubectl logs django-app-7b7d76bd46-wg4bt
Watching for file changes with StatReloader
```

```
C:\Users\hp>kubectl port-forward pod/diango-app-7b7d76bd46-wg4bt 8000:8000
Forwarding from 127.0.0.1:8000 -> 8000
Forwarding from [::1]:8000 -> 8000
Handling connection for 8000
Handling connection for 8000
Handling connection for 8000
Handling connection for 8000
```

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

SHOP

Categories + Add

Products + Add

Add product

Titre:

MacBook Pro

Prix:

1000

Description:

New MacBook Pro

Category:

PC

My Photo:

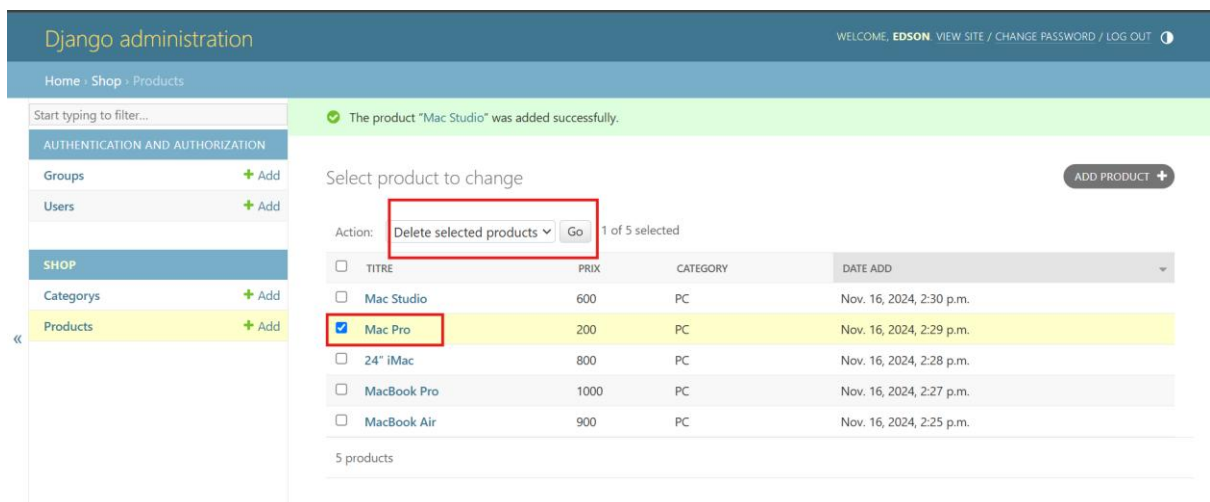
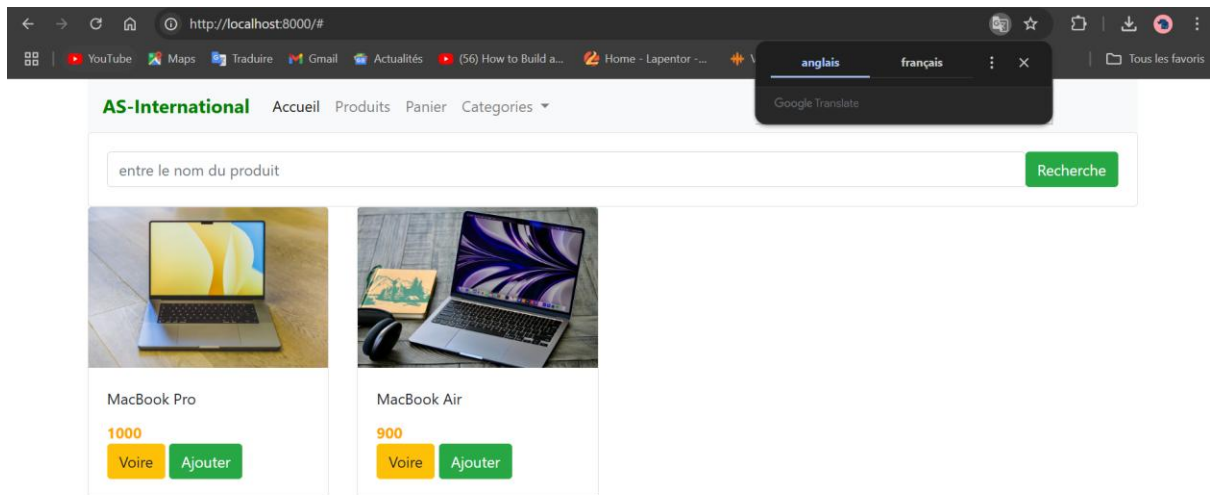
Choisir un fichier

Image:

https://macexpertguide.com/wp-content/upl...

Activer Windows

Activer Windows



entre le nom du produit

Recherche



Mac Studio

600

Voire

Ajouter



24" iMac

800

Voire

Ajouter



MacBook Pro

1000

Voire

Ajouter



MacBook Air

900

Voire

Ajouter

Activer Windows

pou