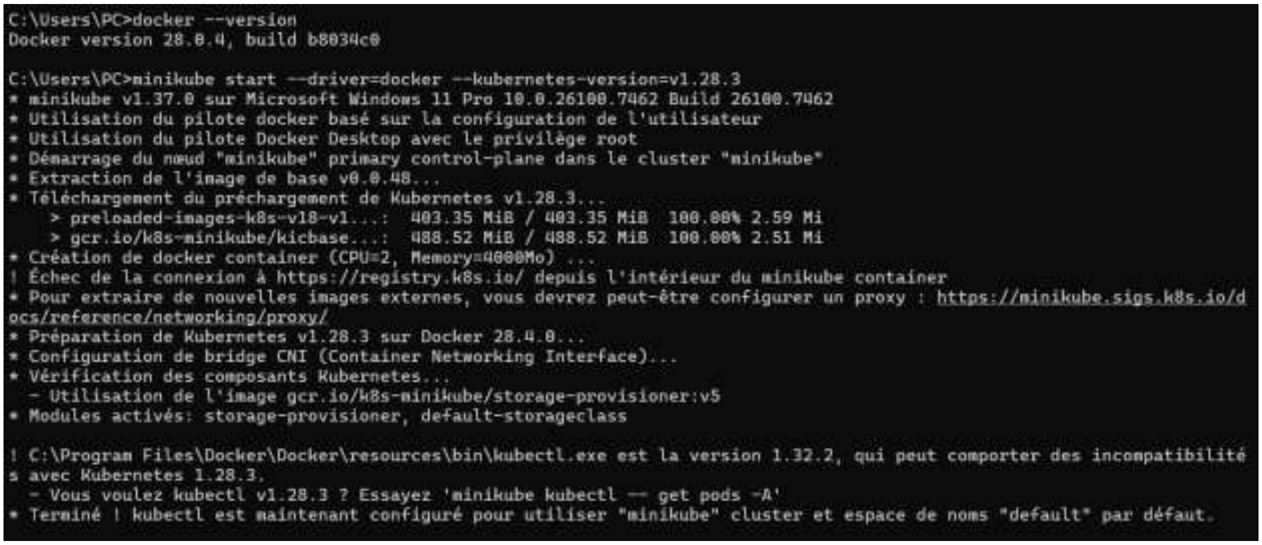
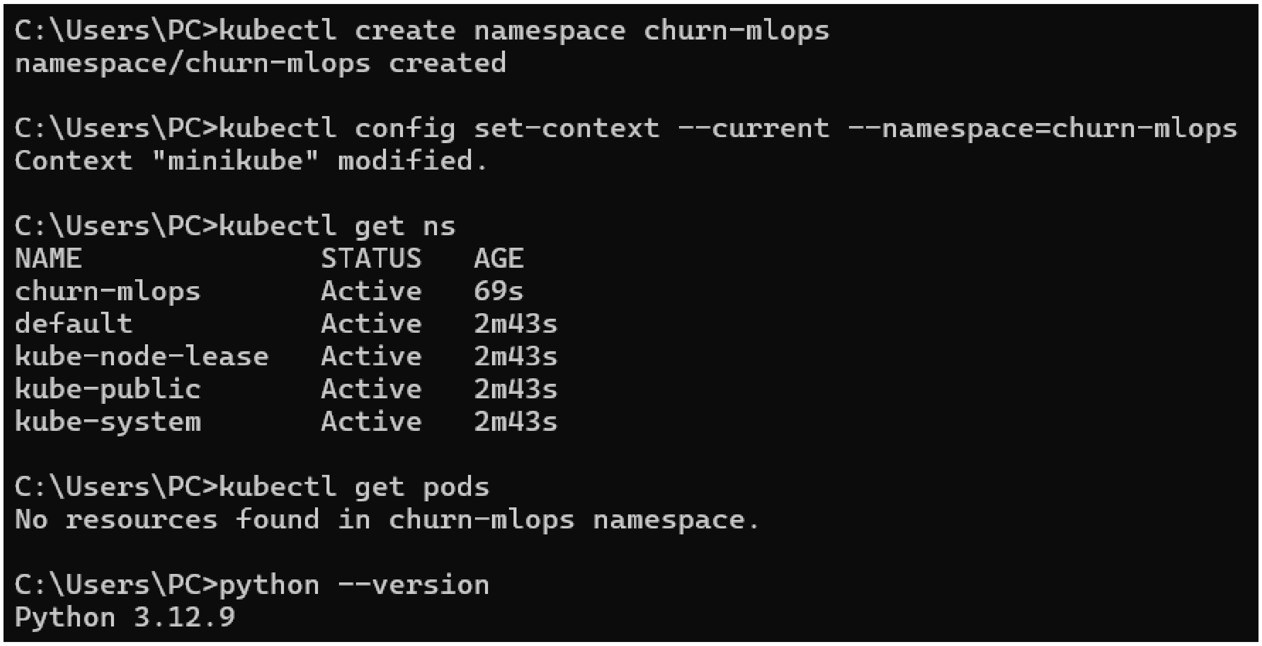
Lab 6 : Déploiement K8s d’un système MLOps Churn

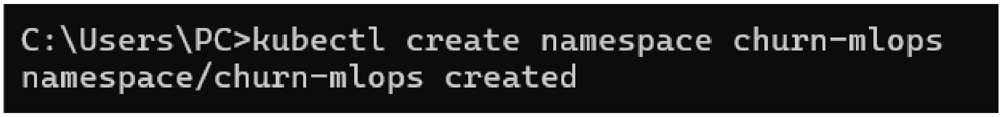
Étape 1 : Préparer l’environnement Kubernetes



```
C:\Users\PC>docker --version
Docker version 28.0.4, build b8034c0

C:\Users\PC>minikube start --driver=docker --kubernetes-version=v1.28.3
* minikube v1.37.0 sur Microsoft Windows 11 Pro 10.0.26100.7462 Build 26100.7462
* Utilisation du pilote docker basé sur la configuration de l'utilisateur
* Utilisation du pilote Docker Desktop avec le privilège root
* Démarrage du nœud "minikube" primary control-plane dans le cluster "minikube"
* Extraction de l'image de base v0.0.48...
* Téléchargement du préchargement de Kubernetes v1.28.3...
    > preloaded-images-k8s-v18-v1...:  403.35 MiB / 403.35 MiB  100.00% 2.59 Mi
    > gcr.io/k8s-minikube/kicbase...:  488.52 MiB / 488.52 MiB  100.00% 2.51 Mi
* Création de docker container (CPU=2, Memory=4000Mo) ...
! Échec de la connexion à https://registry.k8s.io/ depuis l'intérieur du minikube container
* Pour extraire de nouvelles images externes, vous devrez peut-être configurer un proxy : https://minikube.sigs.k8s.io/d
ocs/reference/networking/proxy/
* Préparation de Kubernetes v1.28.3 sur Docker 28.4.0...
* Configuration de bridge CNI (Container Networking Interface)...
* Vérification des composants Kubernetes...
  - Utilisation de l'image gcr.io/k8s-minikube/storage-provisioner:v5
* Modules activés: storage-provisioner, default-storageclass

! C:\Program Files\Docker\Docker\resources\bin\kubectl.exe est la version 1.32.2, qui peut comporter des incompatibilité
s avec Kubernetes 1.28.3.
  - Vous voulez kubectl v1.28.3 ? Essayez 'minikube kubectl -- get pods -A'
* Terminé ! kubectl est maintenant configuré pour utiliser "minikube" cluster et espace de noms "default" par défaut.
```

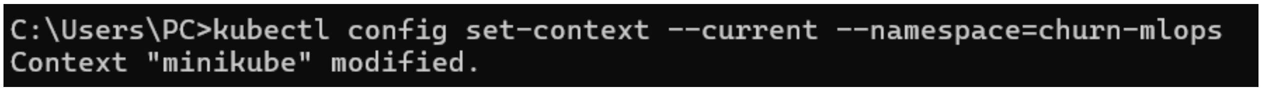


```
C:\Users\PC>kubectl create namespace churn-mlops
namespace/churn-mlops created

C:\Users\PC>kubectl config set-context --current --namespace=churn-mlops
Context "minikube" modified.

C:\Users\PC>kubectl get ns
NAME              STATUS   AGE
churn-mlops       Active   69s
default           Active   2m43s
kube-node-lease   Active   2m43s
kube-public       Active   2m43s
kube-system       Active   2m43s

C:\Users\PC>kubectl get pods
No resources found in churn-mlops namespace.

C:\Users\PC>python --version
Python 3.12.9
```

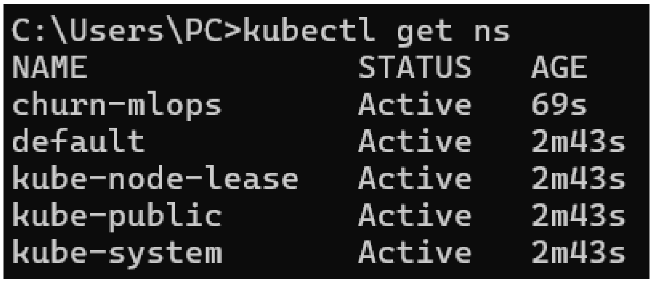


```
C:\Users\PC>kubectl create namespace churn-mlops
namespace/churn-mlops created
```

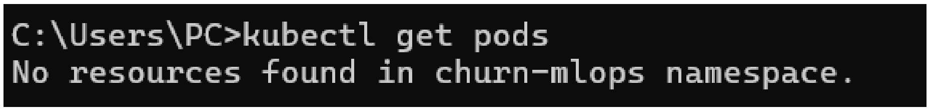
Namespace: clustervirtuel



```
C:\Users\PC>kubectl config set-context --current --namespace=churn-mlops
Context "minikube" modified.
```

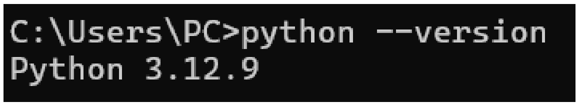


```
C:\Users\PC>kubectl get ns
NAME              STATUS   AGE
churn-mlops       Active   69s
default           Active   2m43s
kube-node-lease   Active   2m43s
kube-public       Active   2m43s
kube-system       Active   2m43s
```

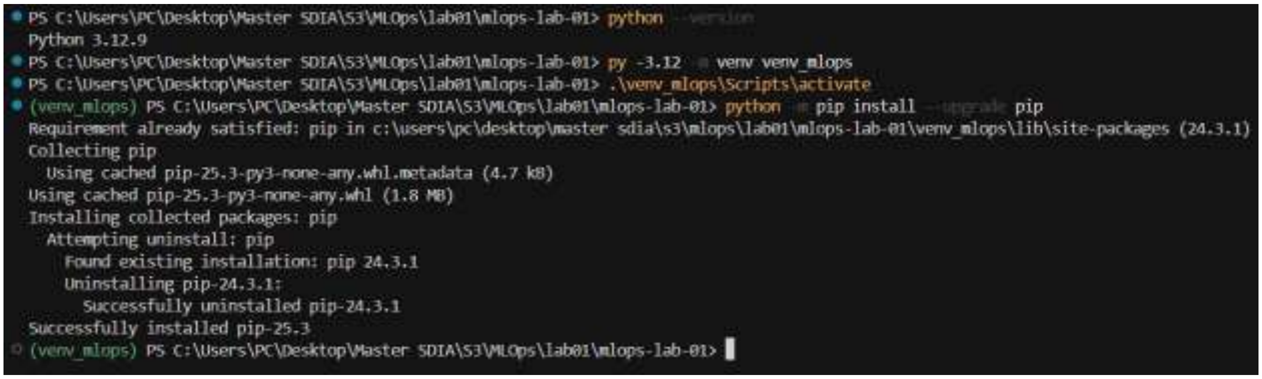


```
C:\Users\PC>kubectl get pods
No resources found in churn-mlops namespace.
```

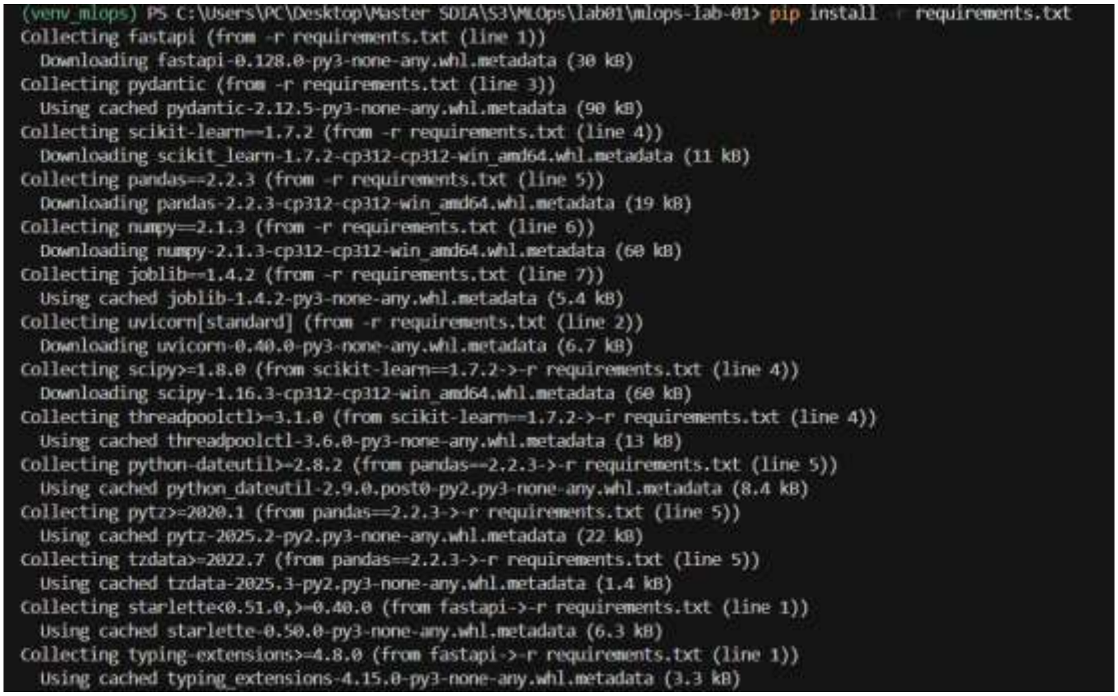
Étape 2 : Préparer l’image Docker de l’API churn



```
C:\Users\PC>python --version
Python 3.12.9
```

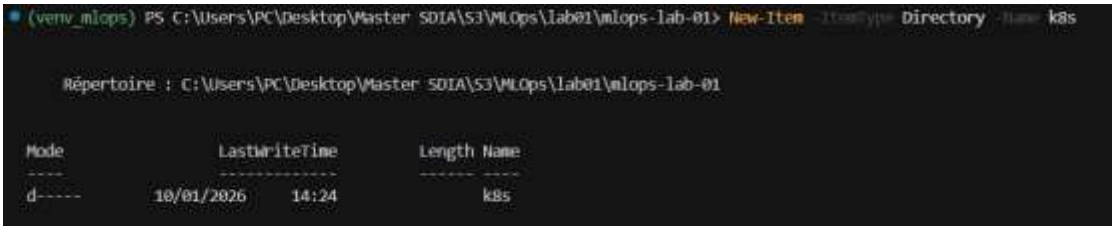


```
● PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> python --version
  Python 3.12.9
● PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> py -3.12 -m venv venv_mlops
● PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> .\venv_mlops\Scripts\activate
● (venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> python -m pip install --upgrade pip
  Requirement already satisfied: pip in c:\users\pc\desktop\master sdia\s3\mlops\lab01\mlops-lab-01\venv_mlops\lib\site-packages (24.3.1)
  Collecting pip
    Using cached pip-25.3-py3-none-any.whl.metadata (4.7 kB)
  Using cached pip-25.3-py3-none-any.whl (1.8 MB)
  Installing collected packages: pip
    Attempting uninstall: pip
      Found existing installation: pip 24.3.1
      Uninstalling pip-24.3.1:
        Successfully uninstalled pip-24.3.1
  Successfully installed pip-25.3
○ (venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> █
```

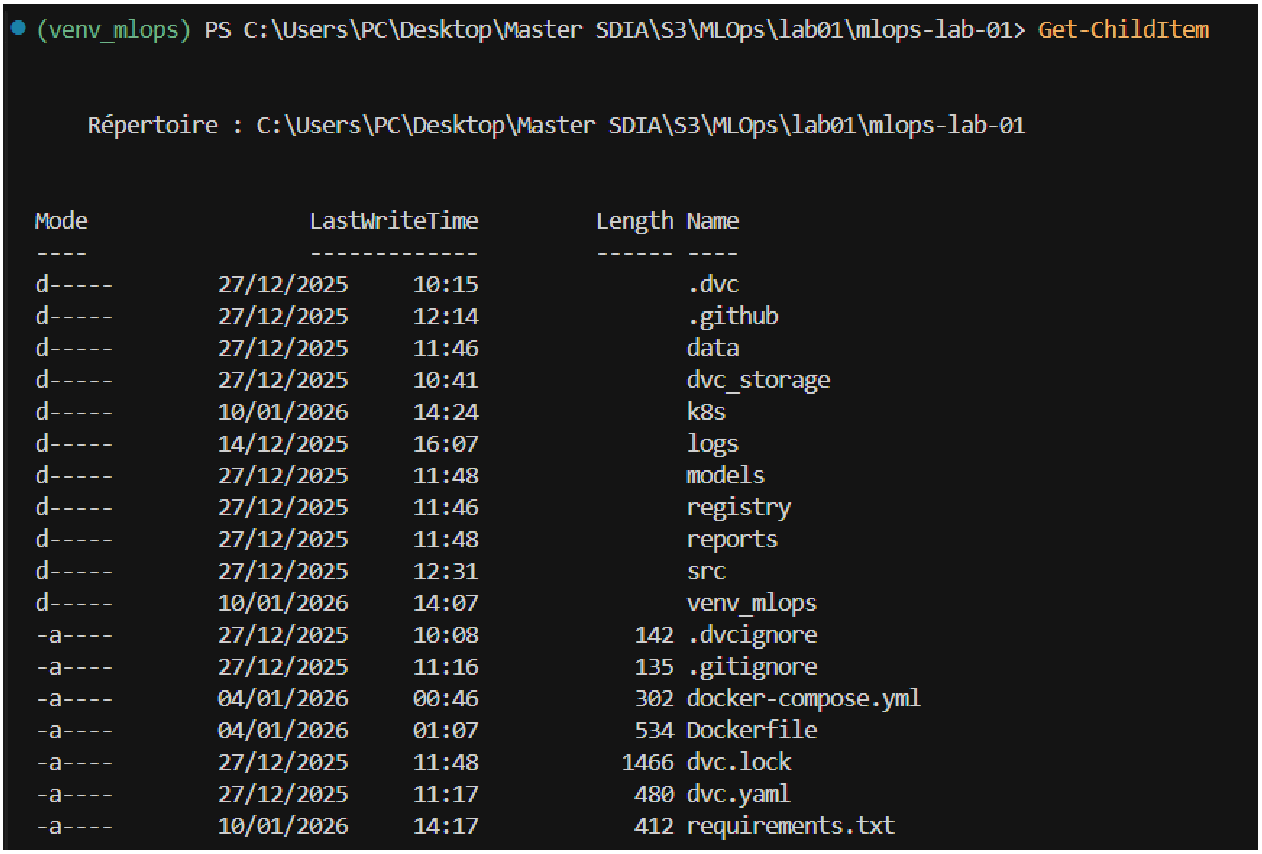


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> pip install -r requirements.txt
Collecting fastapi (from -r requirements.txt (line 1))
  Downloading fastapi-0.128.0-py3-none-any.whl.metadata (30 kB)
Collecting pydantic (from -r requirements.txt (line 3))
  Using cached pydantic-2.12.5-py3-none-any.whl.metadata (90 kB)
Collecting scikit-learn==1.7.2 (from -r requirements.txt (line 4))
  Downloading scikit_learn-1.7.2-cp312-cp312-win_amd64.whl.metadata (11 kB)
Collecting pandas==2.2.3 (from -r requirements.txt (line 5))
  Downloading pandas-2.2.3-cp312-cp312-win_amd64.whl.metadata (19 kB)
Collecting numpy==2.1.3 (from -r requirements.txt (line 6))
  Downloading numpy-2.1.3-cp312-cp312-win_amd64.whl.metadata (60 kB)
Collecting joblib==1.4.2 (from -r requirements.txt (line 7))
  Using cached joblib-1.4.2-py3-none-any.whl.metadata (5.4 kB)
Collecting uvicorn[standard] (from -r requirements.txt (line 2))
  Downloading uvicorn-0.40.0-py3-none-any.whl.metadata (6.7 kB)
Collecting scipy>=1.8.0 (from scikit-learn==1.7.2->-r requirements.txt (line 4))
  Downloading scipy-1.16.3-cp312-cp312-win_amd64.whl.metadata (60 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn==1.7.2->-r requirements.txt (line 4))
  Using cached threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
Collecting python-dateutil>=2.8.2 (from pandas==2.2.3->-r requirements.txt (line 5))
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas==2.2.3->-r requirements.txt (line 5))
  Using cached pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas==2.2.3->-r requirements.txt (line 5))
  Using cached tzdata-2025.3-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting starlette<0.51.0,>=0.40.0 (from fastapi->-r requirements.txt (line 1))
  Using cached starlette-0.50.0-py3-none-any.whl.metadata (6.3 kB)
Collecting typing-extensions>=4.8.0 (from fastapi->-r requirements.txt (line 1))
  Using cached typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
```

Étape 3 : Créer le dossier des manifests Kubernetes

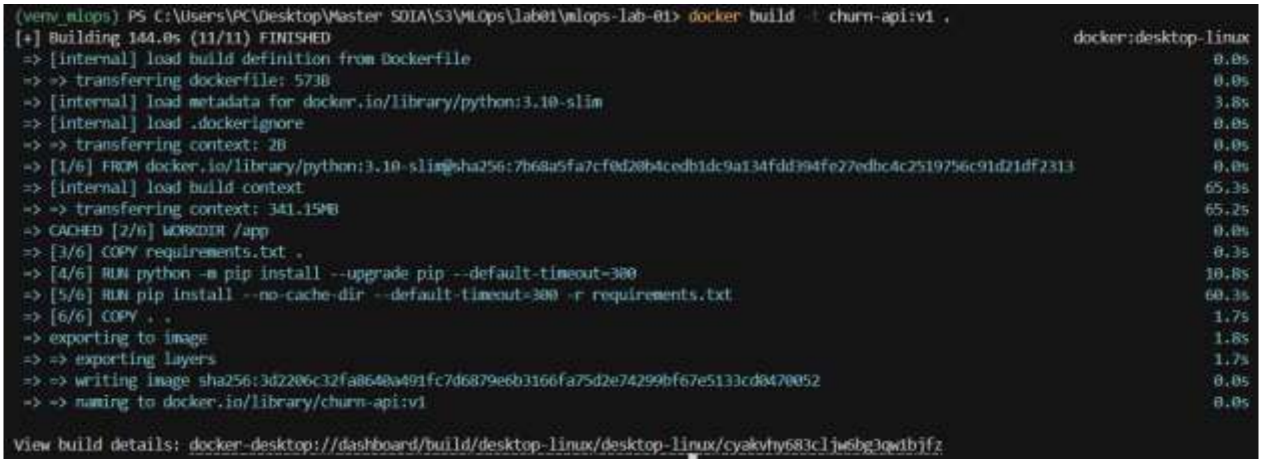


```
● (venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> New-Item -ItemType Directory -Name k8s


    Répertoire : C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/01/2026     14:24                k8s
```

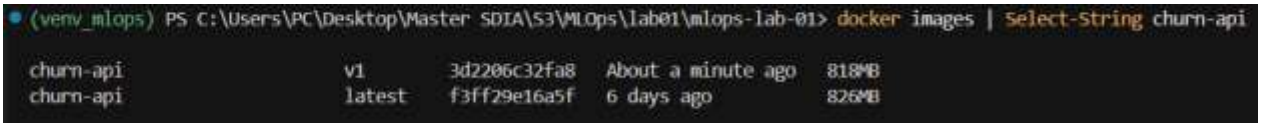


```
● (venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> Get-ChildItem


    Répertoire : C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        27/12/2025     10:15                .dvc
d-----        27/12/2025     12:14                .github
d-----        27/12/2025     11:46                data
d-----        27/12/2025     10:41                dvc_storage
d-----        10/01/2026     14:24                k8s
d-----        14/12/2025     16:07                logs
d-----        27/12/2025     11:48                models
d-----        27/12/2025     11:46                registry
d-----        27/12/2025     11:48                reports
d-----        27/12/2025     12:31                src
d-----        10/01/2026     14:07                venv_mlops
-a----        27/12/2025     10:08            142 .dvcignore
-a----        27/12/2025     11:16            135 .gitignore
-a----        04/01/2026     00:46            302 docker-compose.yml
-a----        04/01/2026     01:07            534 Dockerfile
-a----        27/12/2025     11:48           1466 dvc.lock
-a----        27/12/2025     11:17            480 dvc.yaml
-a----        10/01/2026     14:17            412 requirements.txt
```

Étape 4 : Construire l'image Docker (tag versionné)

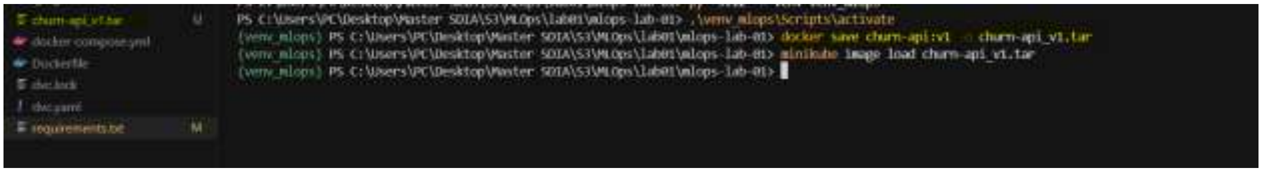


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> docker build -t churn-api:v1 .
[+] Building 144.0s (11/11) FINISHED                                                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                  0.0s
 => => transferring dockerfile: 573B                                                                                  0.0s
 => [internal] load metadata for docker.io/library/python:3.10-slim                                                   3.8s
 => [internal] load .dockerignore                                                                                     0.0s
 => => transferring context: 2B                                                                                       0.0s
 => [1/6] FROM docker.io/library/python:3.10-slim@sha256:7b68a5fa7cf0d20b4cedb1dc9a134fdd394fe27edbc4c2519756c91d21df2313   0.0s
 => [internal] load build context                                                                                    65.3s
 => => transferring context: 341.15MB                                                                                65.2s
 => CACHED [2/6] WORKDIR /app                                                                                         0.0s
 => [3/6] COPY requirements.txt .                                                                                     0.3s
 => [4/6] RUN python -m pip install --upgrade pip --default-timeout=300                                              10.8s
 => [5/6] RUN pip install --no-cache-dir --default-timeout=300 -r requirements.txt                                   60.3s
 => [6/6] COPY . .                                                                                                    1.7s
 => exporting to image                                                                                                1.8s
 => => exporting layers                                                                                               1.7s
 => => writing image sha256:3d2206c32fa8640a491fc7d6879e6b3166fa75d2e74299bf67e5133cd0470052                          0.0s
 => => naming to docker.io/library/churn-api:v1                                                                       0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/cyakvhy683cljw6bg3qw1bjfz
```

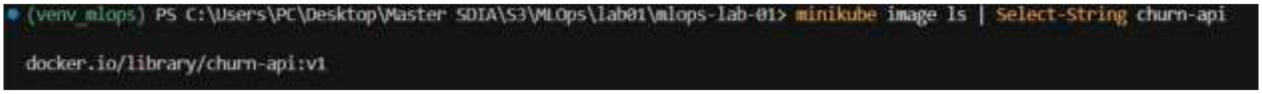


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> docker images | Select-String churn-api

  churn-api                    v1        3d2206c32fa8   About a minute ago   818MB
  churn-api                    latest    f3ff29e16a5f   6 days ago           826MB
```

Étape 5 : Charger explicitement l’image dans Minikube

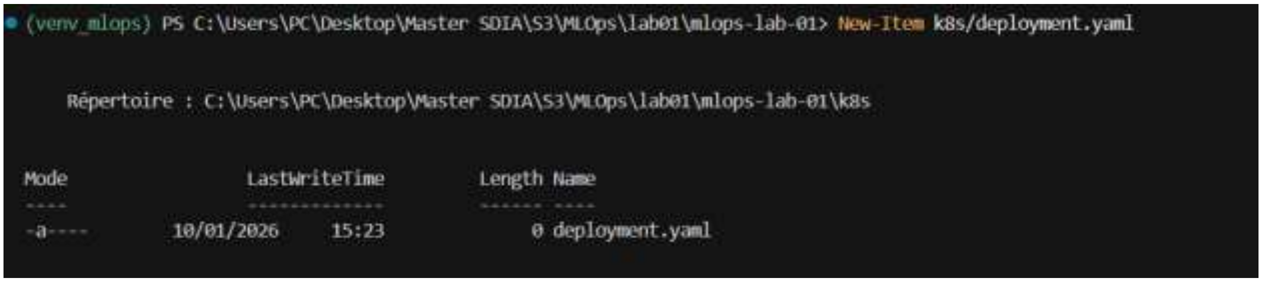


```
PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> .\venv_mlops\Scripts\activate
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> docker save churn-api:v1 -o churn-api_v1.tar
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> minikube image load churn-api_v1.tar
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01>
```

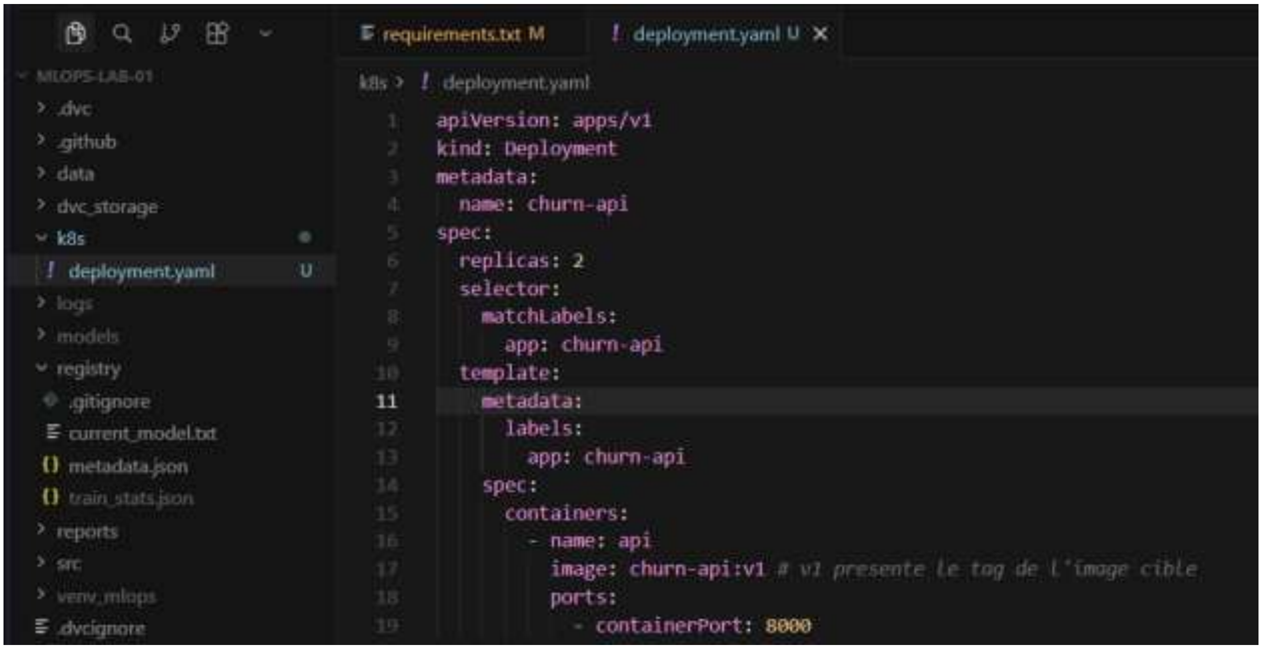


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> minikube image ls | Select-String churn-api

docker.io/library/churn-api:v1
```

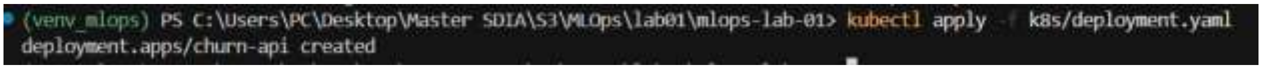
Étape 6 : Deployment Kubernetes pour l’API churn



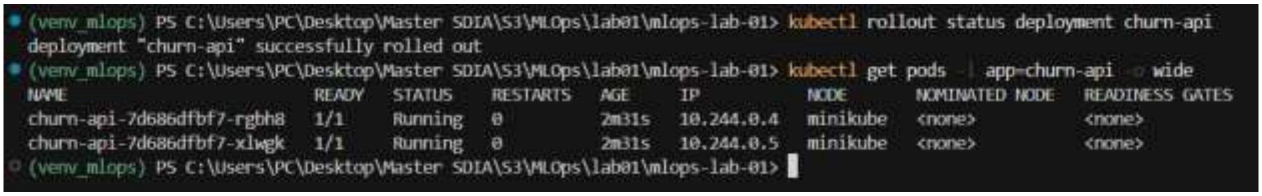
```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> New-Item k8s/deployment.yaml


    Répertoire : C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01\k8s


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/01/2026     15:23              0 deployment.yaml
```



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: churn-api
spec:
  replicas: 2
  selector:
    matchLabels:
      app: churn-api
  template:
    metadata:
      labels:
        app: churn-api
    spec:
      containers:
        - name: api
          image: churn-api:v1 # v1 presente le tag de l'image cible
          ports:
            - containerPort: 8000
```

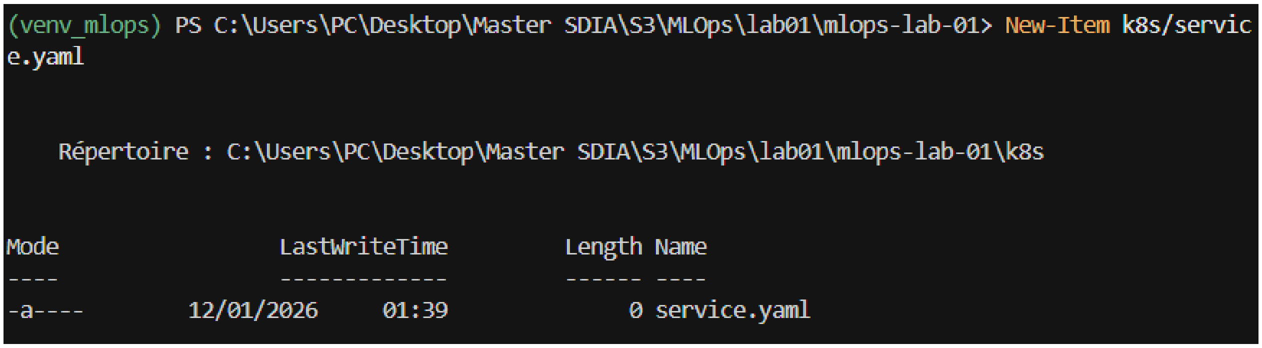


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/deployment.yaml
deployment.apps/churn-api created
```

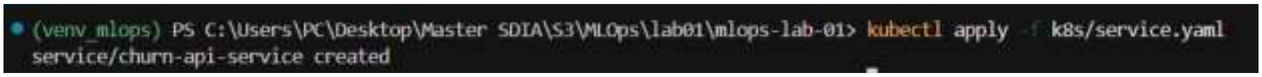


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl rollout status deployment churn-api
deployment "churn-api" successfully rolled out
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get pods -l app=churn-api -o wide
NAME                         READY   STATUS    RESTARTS   AGE     IP           NODE       NOMINATED NODE   READINESS GATES
churn-api-7d686dfbf7-rgbh8   1/1     Running   0          2m31s   10.244.0.4   minikube   <none>           <none>
churn-api-7d686dfbf7-xlwgk   1/1     Running   0          2m31s   10.244.0.5   minikube   <none>           <none>
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01>
```

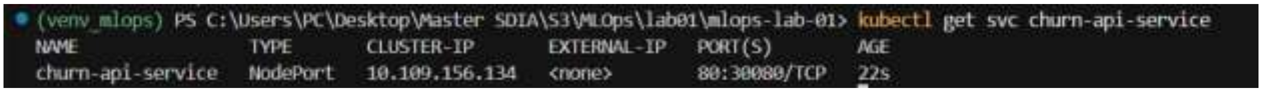
Étape 7 : Exposer l’API via un Service NodePort



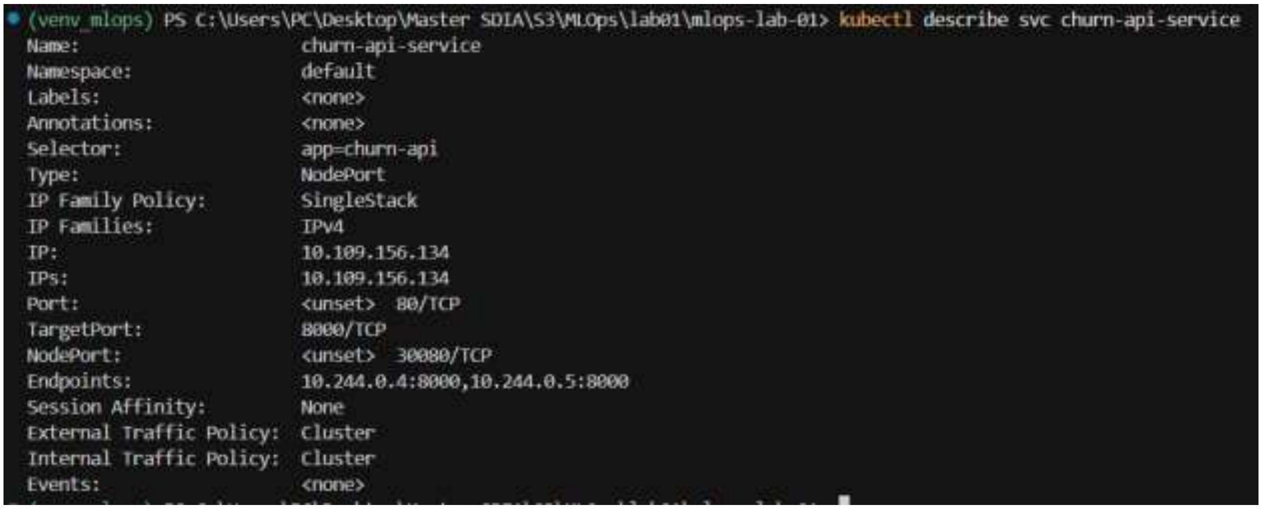
```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> New-Item k8s/servic
e.yaml


    Répertoire : C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01\k8s


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        12/01/2026     01:39              0 service.yaml
```



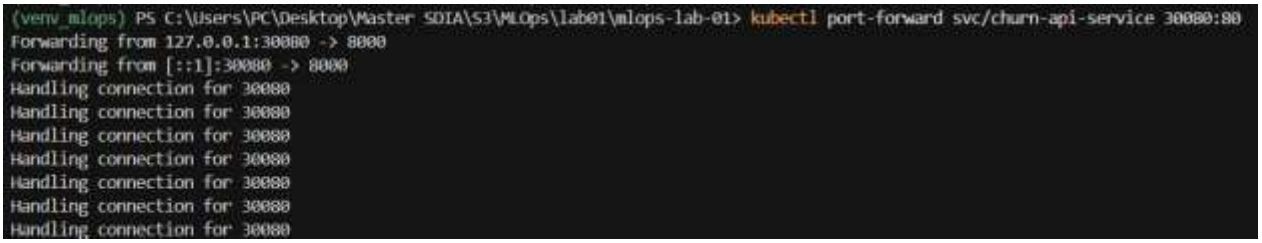
```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/service.yaml
service/churn-api-service created
```



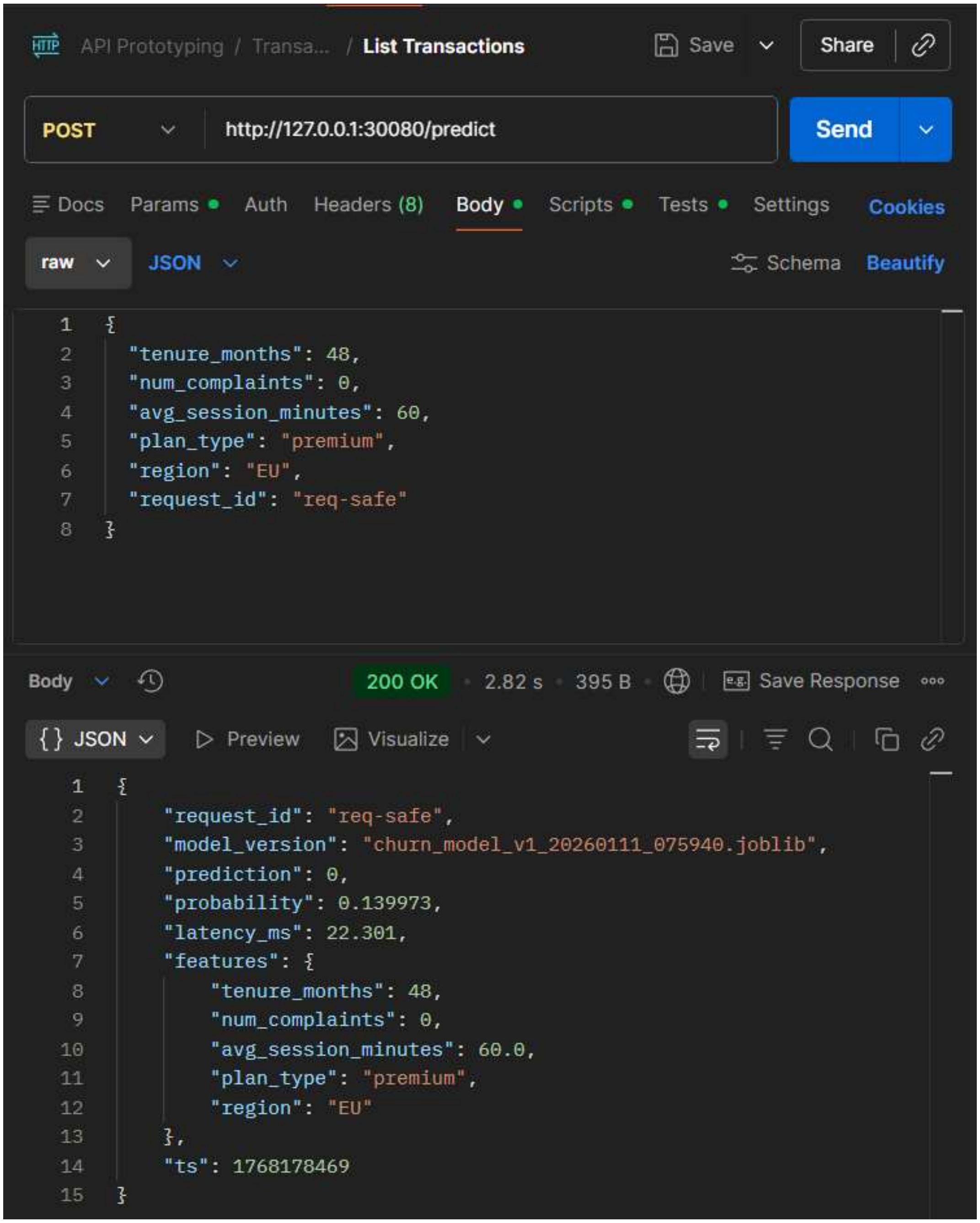
```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get svc churn-api-service
NAME                TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
churn-api-service   NodePort   10.109.156.134   <none>        80:30080/TCP   22s
```



```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl describe svc churn-api-service
Name:                     churn-api-service
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 app=churn-api
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.109.156.134
IPs:                      10.109.156.134
Port:                     <unset>  80/TCP
TargetPort:               8000/TCP
NodePort:                 <unset>  30080/TCP
Endpoints:                10.244.0.4:8000,10.244.0.5:8000
Session Affinity:         None
External Traffic Policy:  Cluster
Internal Traffic Policy:  Cluster
Events:                   <none>
```



```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl port-forward svc/churn-api-service 30080:80
Forwarding from 127.0.0.1:30080 -> 8000
Forwarding from [::1]:30080 -> 8000
Handling connection for 30080
Handling connection for 30080
Handling connection for 30080
Handling connection for 30080
Handling connection for 30080
Handling connection for 30080
Handling connection for 30080
```



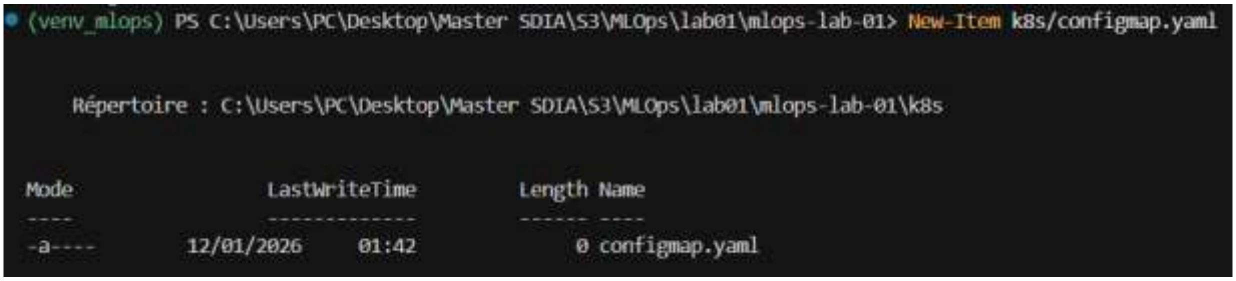
POST http://127.0.0.1:30080/predict

```
{
  "tenure_months": 48,
  "num_complaints": 0,
  "avg_session_minutes": 60,
  "plan_type": "premium",
  "region": "EU",
  "request_id": "req-safe"
}
```

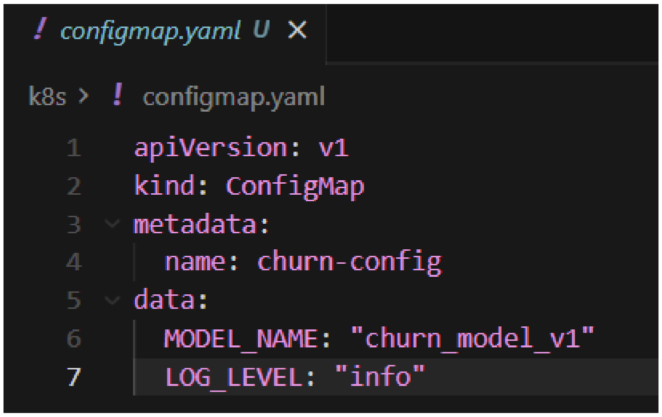
200 OK • 2.82 s • 395 B

```
{
    "request_id": "req-safe",
    "model_version": "churn_model_v1_20260111_075940.joblib",
    "prediction": 0,
    "probability": 0.139973,
    "latency_ms": 22.301,
    "features": {
        "tenure_months": 48,
        "num_complaints": 0,
        "avg_session_minutes": 60.0,
        "plan_type": "premium",
        "region": "EU"
    },
    "ts": 1768178469
}
```

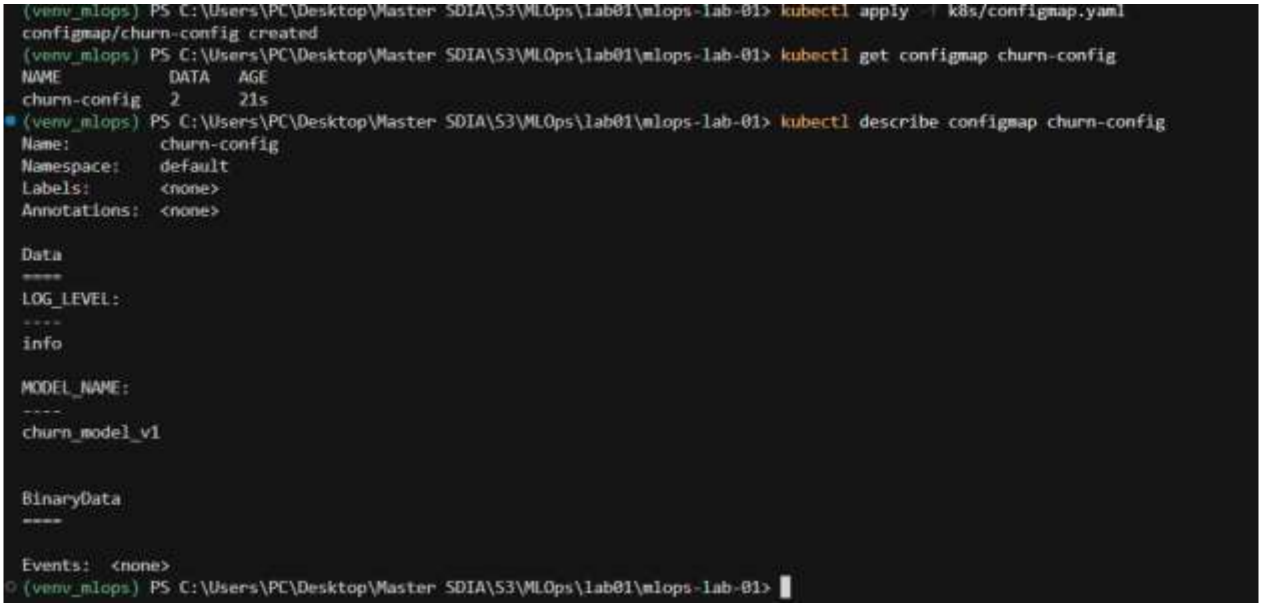
Étape 8 : Injecter la configuration MLOps via ConfigMap



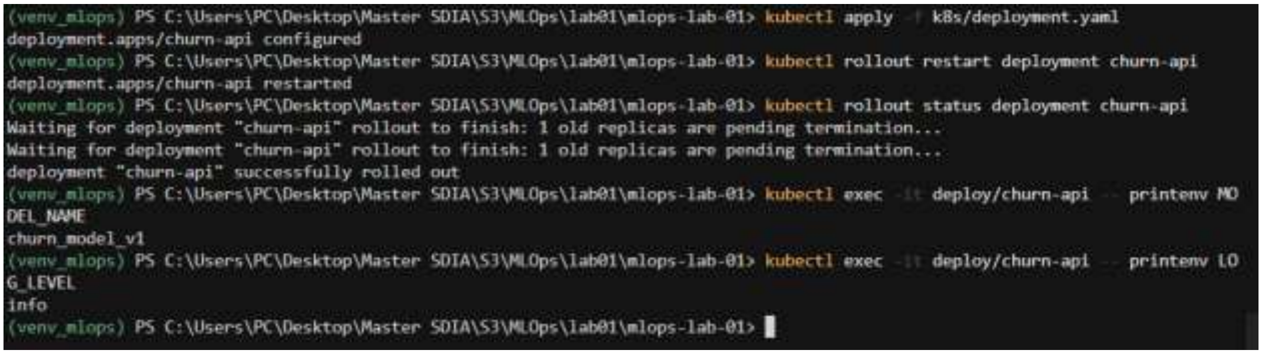
```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> New-Item k8s/configmap.yaml


    Répertoire : C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01\k8s


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        12/01/2026     01:42              0 configmap.yaml
```



```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: churn-config
data:
  MODEL_NAME: "churn_model_v1"
  LOG_LEVEL: "info"
```

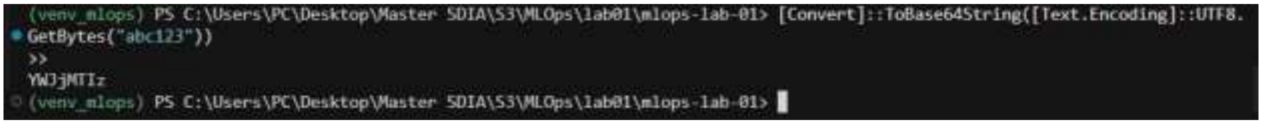


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/configmap.yaml
configmap/churn-config created
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get configmap churn-config
NAME           DATA   AGE
churn-config   2      21s
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl describe configmap churn-config
Name:         churn-config
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
LOG_LEVEL:
----
info

MODEL_NAME:
----
churn_model_v1


BinaryData
====

Events:  <none>
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> 
```

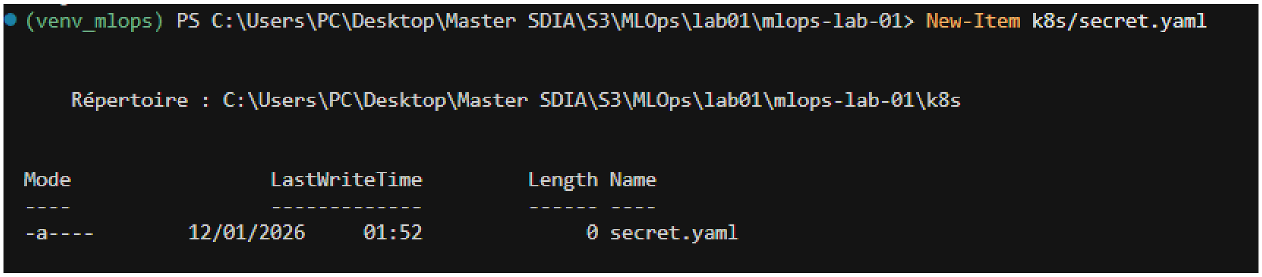


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/deployment.yaml
deployment.apps/churn-api configured
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl rollout restart deployment churn-api
deployment.apps/churn-api restarted
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl rollout status deployment churn-api
Waiting for deployment "churn-api" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "churn-api" rollout to finish: 1 old replicas are pending termination...
deployment "churn-api" successfully rolled out
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl exec -it deploy/churn-api -- printenv MO
DEL_NAME
churn_model_v1
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl exec -it deploy/churn-api -- printenv LO
G_LEVEL
info
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> 
```

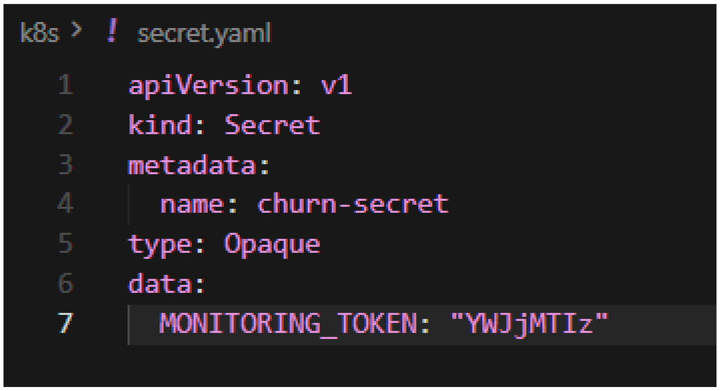
Étape 9 : Gérer les secrets (MONITORING_TOKEN)



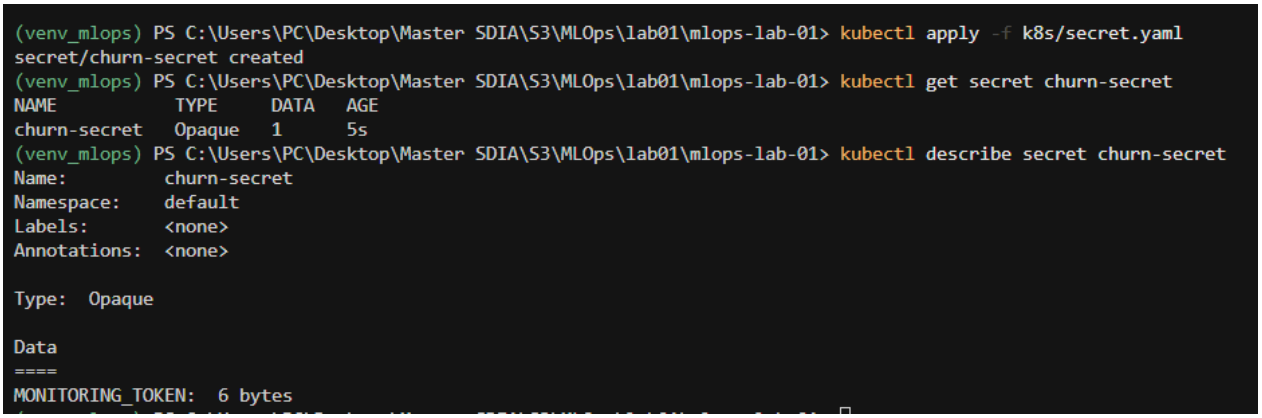
```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> [Convert]::ToBase64String([Text.Encoding]::UTF8.
GetBytes("abc123"))
>>
YWJjMTIz
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> 
```



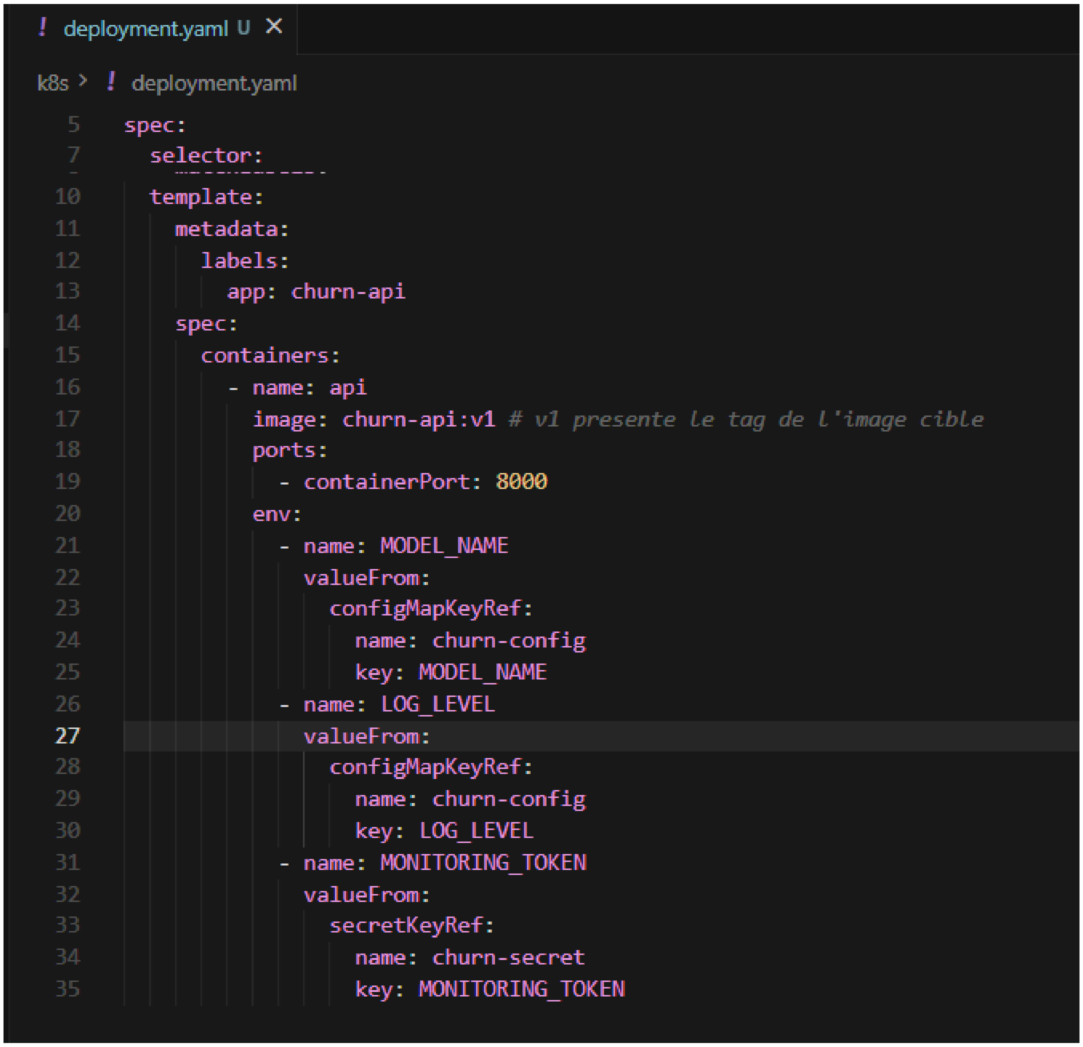
```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> New-Item k8s/secret.yaml


    Répertoire : C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01\k8s


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        12/01/2026     01:52              0 secret.yaml
```



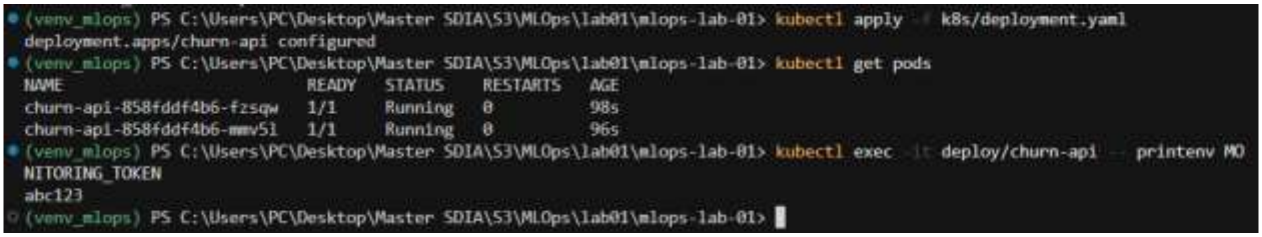
```yaml
apiVersion: v1
kind: Secret
metadata:
  name: churn-secret
type: Opaque
data:
  MONITORING_TOKEN: "YWJjMTIz"
```



```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/secret.yaml
secret/churn-secret created
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get secret churn-secret
NAME           TYPE     DATA   AGE
churn-secret   Opaque   1      5s
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl describe secret churn-secret
Name:         churn-secret
Namespace:    default
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
MONITORING_TOKEN:  6 bytes
```

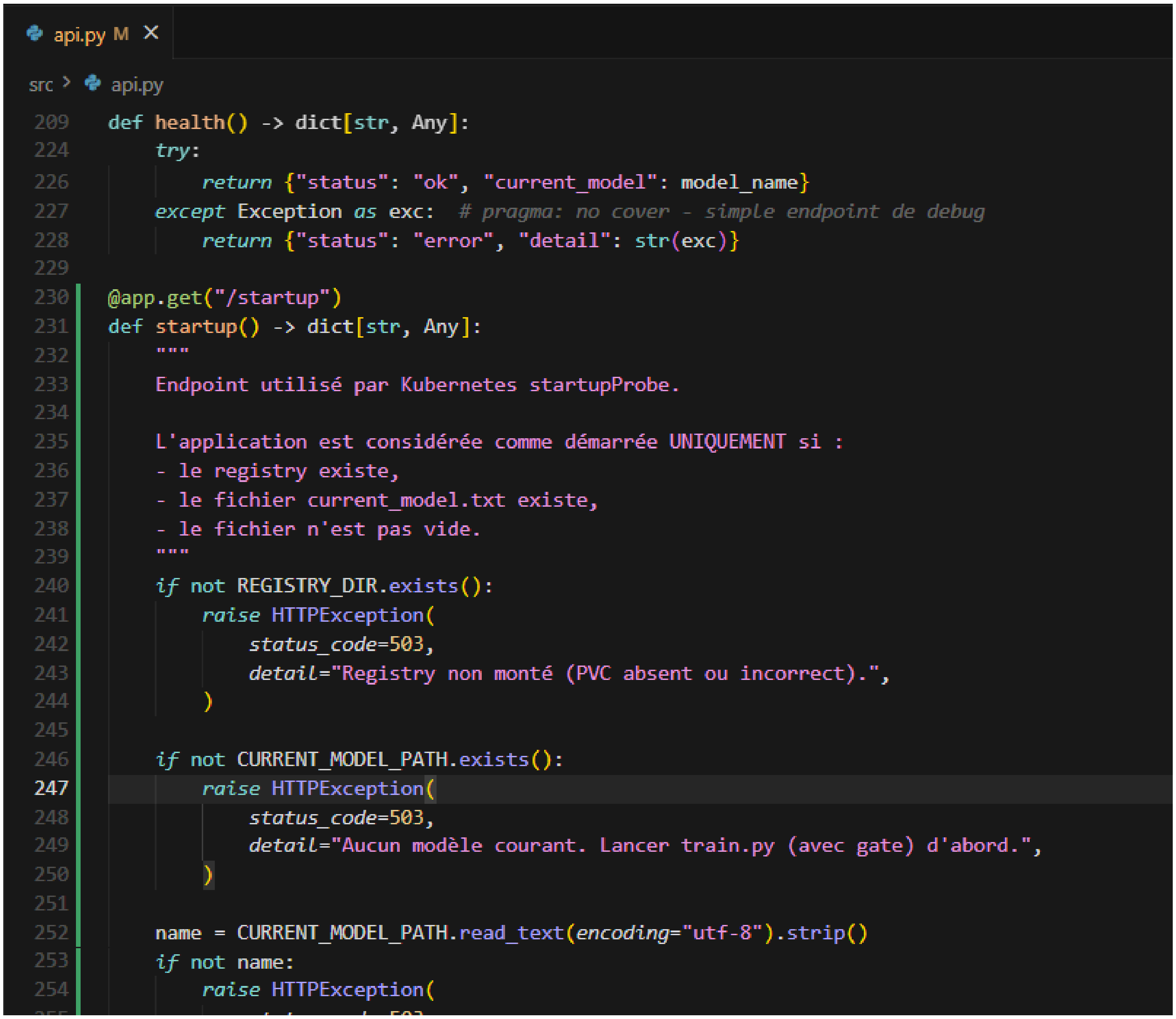


```yaml
spec:
  selector:
  template:
    metadata:
      labels:
        app: churn-api
    spec:
      containers:
        - name: api
          image: churn-api:v1 # v1 presente le tag de l'image cible
          ports:
            - containerPort: 8000
          env:
            - name: MODEL_NAME
              valueFrom:
                configMapKeyRef:
                  name: churn-config
                  key: MODEL_NAME
            - name: LOG_LEVEL
              valueFrom:
                configMapKeyRef:
                  name: churn-config
                  key: LOG_LEVEL
            - name: MONITORING_TOKEN
              valueFrom:
                secretKeyRef:
                  name: churn-secret
                  key: MONITORING_TOKEN
```

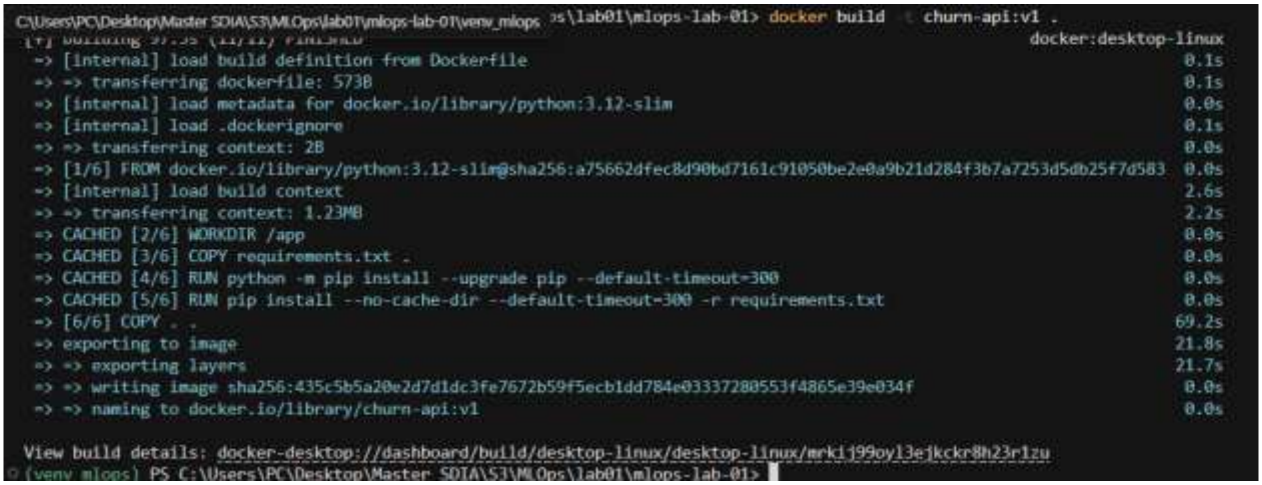


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/deployment.yaml
deployment.apps/churn-api configured
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get pods
NAME                        READY   STATUS    RESTARTS   AGE
churn-api-858fddf4b6-fzsqw   1/1     Running   0          98s
churn-api-858fddf4b6-mmv51   1/1     Running   0          96s
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl exec -it deploy/churn-api -- printenv MO
NITORING_TOKEN
abc123
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01>
```

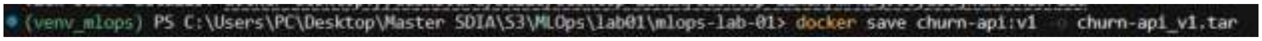
Étape 10 : Mise en place des endpoints de santé et des probes Kubernetes pour l’API Churn



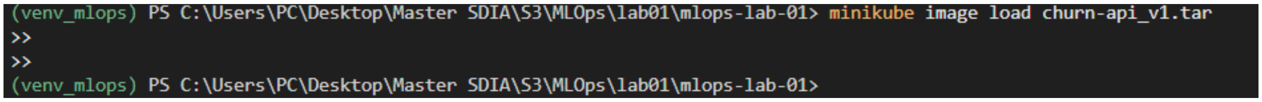
```python
def health() -> dict[str, Any]:
    try:
        return {"status": "ok", "current_model": model_name}
    except Exception as exc:  # pragma: no cover - simple endpoint de debug
        return {"status": "error", "detail": str(exc)}

@app.get("/startup")
def startup() -> dict[str, Any]:
    """
    Endpoint utilisé par Kubernetes startupProbe.

    L'application est considérée comme démarrée UNIQUEMENT si :
    - le registry existe,
    - le fichier current_model.txt existe,
    - le fichier n'est pas vide.
    """
    if not REGISTRY_DIR.exists():
        raise HTTPException(
            status_code=503,
            detail="Registry non monté (PVC absent ou incorrect).",
        )

    if not CURRENT_MODEL_PATH.exists():
        raise HTTPException(
            status_code=503,
            detail="Aucun modèle courant. Lancer train.py (avec gate) d'abord.",
        )

    name = CURRENT_MODEL_PATH.read_text(encoding="utf-8").strip()
    if not name:
        raise HTTPException(
```



```
C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01\venv_mlops s\lab01\mlops-lab-01> docker build -t churn-api:v1 .
[+] Building 97.3s (11/11) FINISHED                                                                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                        0.1s
 => => transferring dockerfile: 573B                                                                                        0.1s
 => [internal] load metadata for docker.io/library/python:3.12-slim                                                        0.0s
 => [internal] load .dockerignore                                                                                           0.1s
 => => transferring context: 2B                                                                                             0.0s
 => [1/6] FROM docker.io/library/python:3.12-slim@sha256:a75662dfec8d90bd7161c91050be2e0a9b21d284f3b7a7253d5db25f7d583  0.0s
 => [internal] load build context                                                                                           2.6s
 => => transferring context: 1.23MB                                                                                         2.2s
 => CACHED [2/6] WORKDIR /app                                                                                               0.0s
 => CACHED [3/6] COPY requirements.txt .                                                                                    0.0s
 => CACHED [4/6] RUN python -m pip install --upgrade pip --default-timeout=300                                              0.0s
 => CACHED [5/6] RUN pip install --no-cache-dir --default-timeout=300 -r requirements.txt                                   0.0s
 => [6/6] COPY . .                                                                                                         69.2s
 => exporting to image                                                                                                     21.8s
 => => exporting layers                                                                                                    21.7s
 => => writing image sha256:435c5b5a20e2d7d1dc3fe7672b59f5ecb1dd784e03337280553f4865e39e034f                                0.0s
 => => naming to docker.io/library/churn-api:v1                                                                             0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/mrkij99oyl3ejkckr8h23r1zu
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01>
```

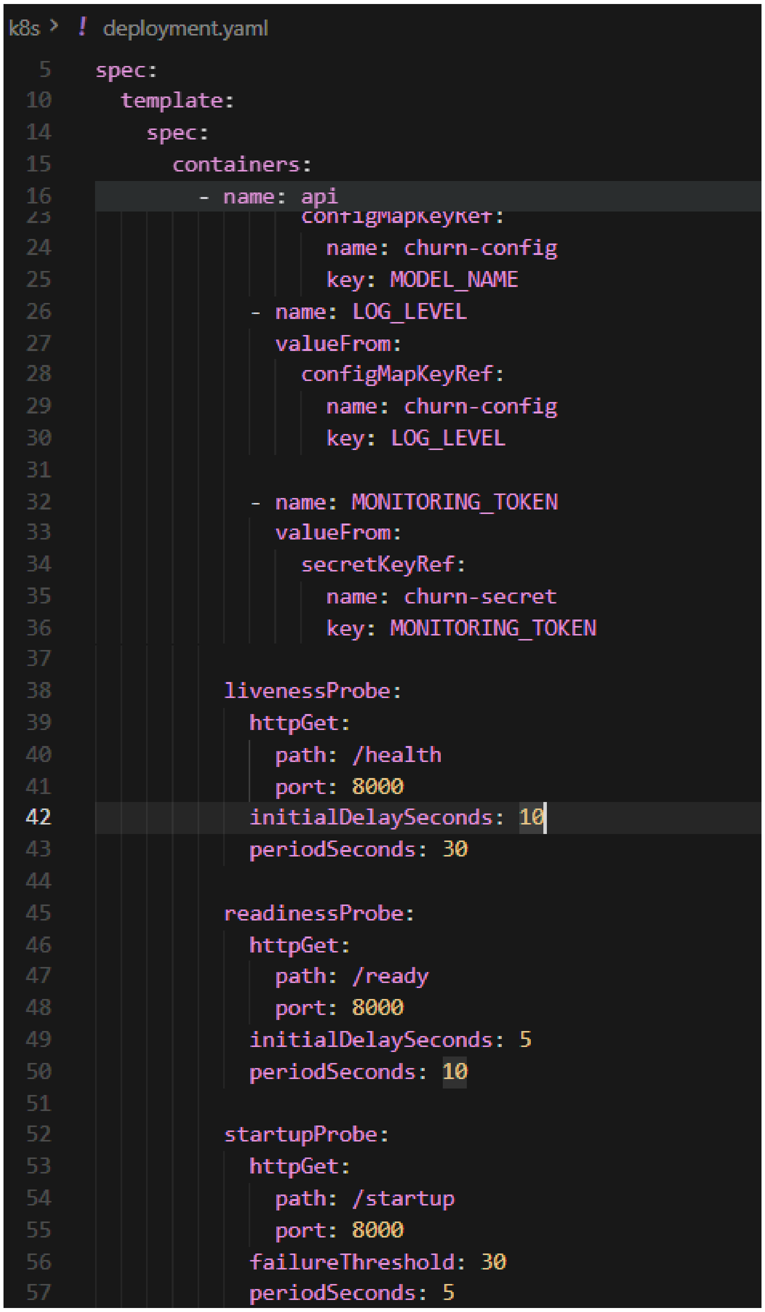


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> docker save churn-api:v1 -o churn-api_v1.tar
```

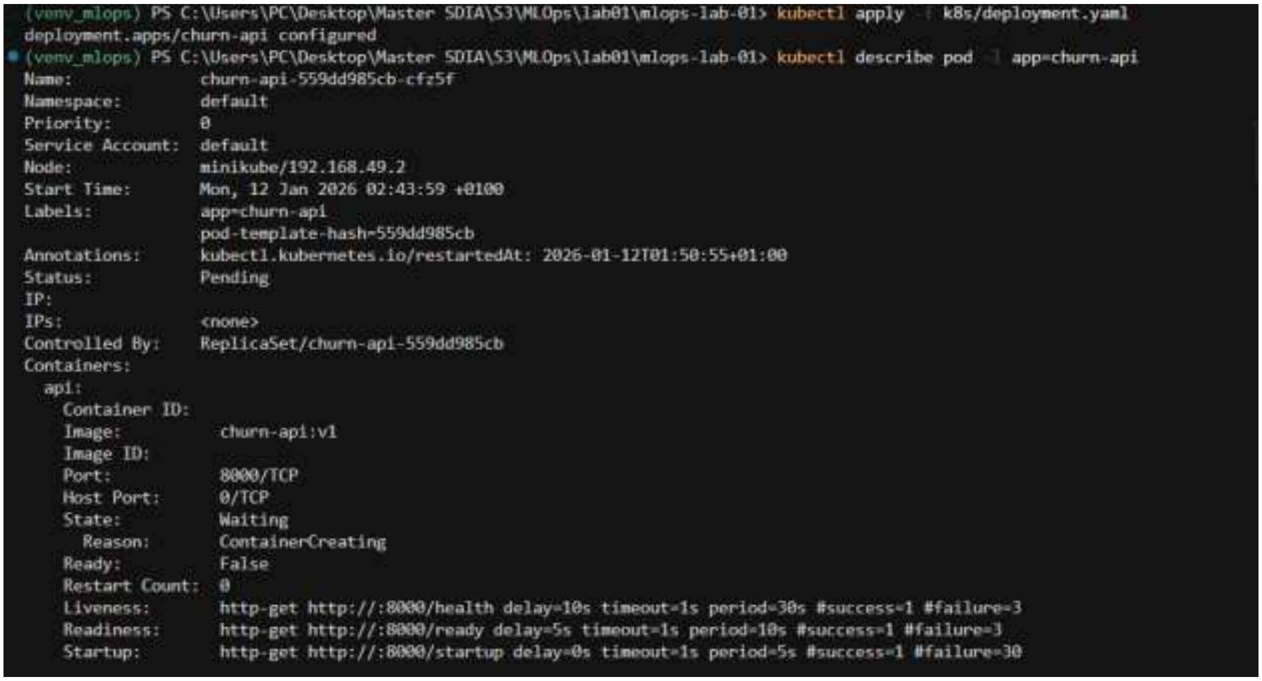


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> minikube image load churn-api_v1.tar
>>
>>
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01>
```

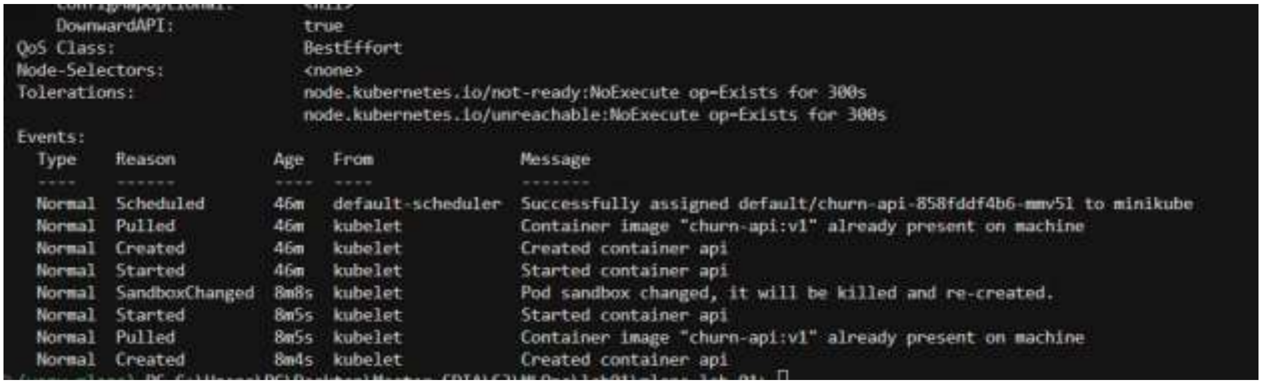
Étape 11 : Ajouter les probes (liveness / readiness / startup)



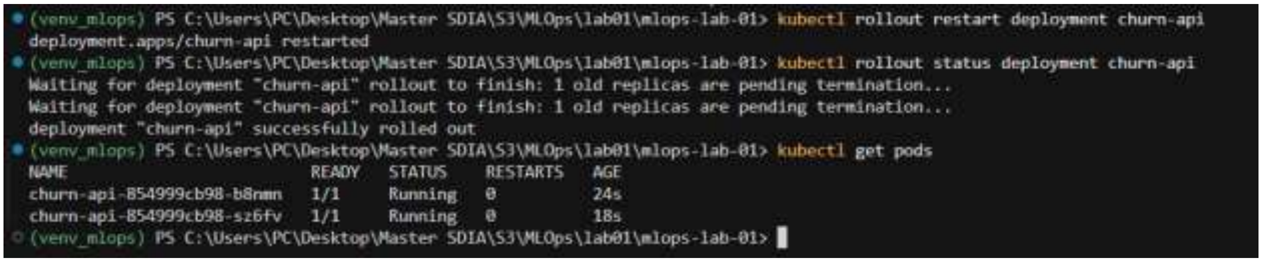
```
spec:
  template:
    spec:
      containers:
        - name: api
              configMapKeyRef:
                name: churn-config
                key: MODEL_NAME
          - name: LOG_LEVEL
            valueFrom:
              configMapKeyRef:
                name: churn-config
                key: LOG_LEVEL

          - name: MONITORING_TOKEN
            valueFrom:
              secretKeyRef:
                name: churn-secret
                key: MONITORING_TOKEN

          livenessProbe:
            httpGet:
              path: /health
              port: 8000
            initialDelaySeconds: 10
            periodSeconds: 30

          readinessProbe:
            httpGet:
              path: /ready
              port: 8000
            initialDelaySeconds: 5
            periodSeconds: 10

          startupProbe:
            httpGet:
              path: /startup
              port: 8000
            failureThreshold: 30
            periodSeconds: 5
```



```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/deployment.yaml
deployment.apps/churn-api configured
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl describe pod -l app=churn-api
Name:             churn-api-559dd985cb-cfz5f
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Mon, 12 Jan 2026 02:43:59 +0100
Labels:           app=churn-api
                  pod-template-hash=559dd985cb
Annotations:      kubectl.kubernetes.io/restartedAt: 2026-01-12T01:50:55+01:00
Status:           Pending
IP:
IPs:              <none>
Controlled By:    ReplicaSet/churn-api-559dd985cb
Containers:
  api:
    Container ID:
    Image:          churn-api:v1
    Image ID:
    Port:           8000/TCP
    Host Port:      0/TCP
    State:          Waiting
      Reason:       ContainerCreating
    Ready:          False
    Restart Count:  0
    Liveness:       http-get http://:8000/health delay=10s timeout=1s period=30s #success=1 #failure=3
    Readiness:      http-get http://:8000/ready delay=5s timeout=1s period=10s #success=1 #failure=3
    Startup:        http-get http://:8000/startup delay=0s timeout=1s period=5s #success=1 #failure=30
```

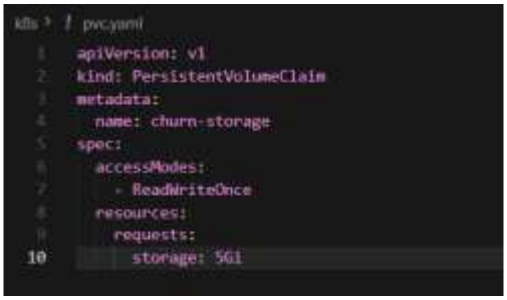


```
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason          Age   From               Message
  ----    ------          ----  ----               -------
  Normal  Scheduled       46m   default-scheduler  Successfully assigned default/churn-api-858fddf4b6-mmv5l to minikube
  Normal  Pulled          46m   kubelet            Container image "churn-api:v1" already present on machine
  Normal  Created         46m   kubelet            Created container api
  Normal  Started         46m   kubelet            Started container api
  Normal  SandboxChanged  8m8s  kubelet            Pod sandbox changed, it will be killed and re-created.
  Normal  Started         8m5s  kubelet            Started container api
  Normal  Pulled          8m5s  kubelet            Container image "churn-api:v1" already present on machine
  Normal  Created         8m4s  kubelet            Created container api
```



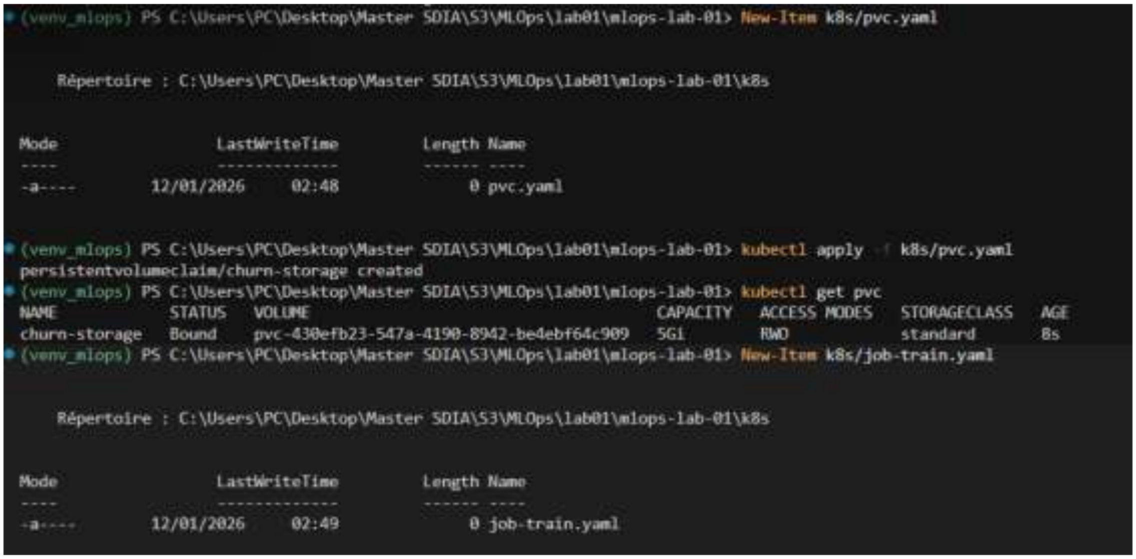
```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl rollout restart deployment churn-api
deployment.apps/churn-api restarted
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl rollout status deployment churn-api
Waiting for deployment "churn-api" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "churn-api" rollout to finish: 1 old replicas are pending termination...
deployment "churn-api" successfully rolled out
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get pods
NAME                         READY   STATUS    RESTARTS   AGE
churn-api-854999cb98-b8nmn   1/1     Running   0          24s
churn-api-854999cb98-sz6fv   1/1     Running   0          18s
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01>
```

Étape 12 : Volume persistant pour registry + logs

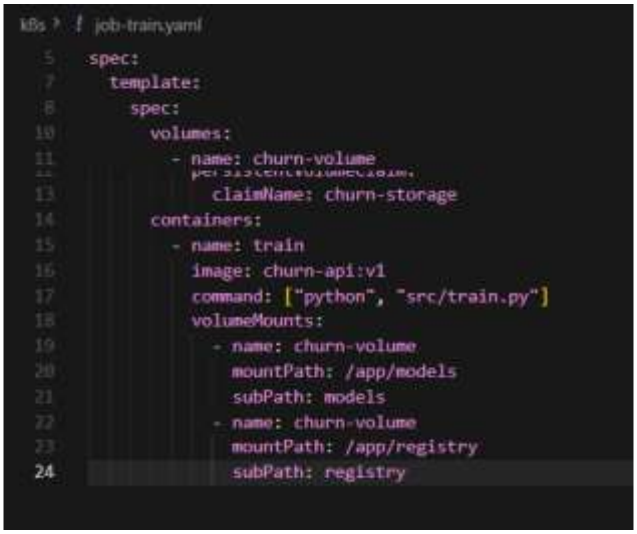


k8s > ! pvc.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: churn-storage
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

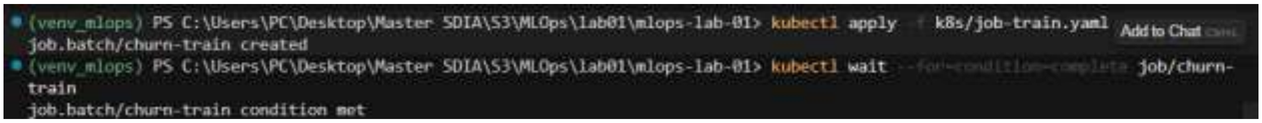


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> New-Item k8s/pvc.yaml


    Répertoire : C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01\k8s


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        12/01/2026     02:48              0 pvc.yaml


(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/pvc.yaml
persistentvolumeclaim/churn-storage created
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get pvc
NAME            STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
churn-storage   Bound    pvc-430efb23-547a-4190-8942-be4ebf64c909   5Gi        RWO            standard       8s
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> New-Item k8s/job-train.yaml


    Répertoire : C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01\k8s


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        12/01/2026     02:49              0 job-train.yaml
```

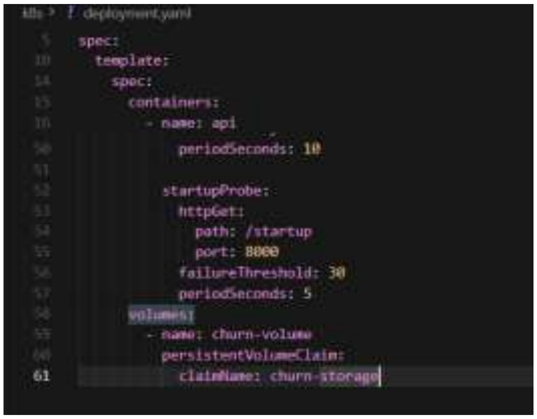


k8s › job-train.yaml

```yaml
spec:
  template:
    spec:
      volumes:
        - name: churn-volume
          persistentVolumeClaim:
            claimName: churn-storage
      containers:
        - name: train
          image: churn-api:v1
          command: ["python", "src/train.py"]
          volumeMounts:
            - name: churn-volume
              mountPath: /app/models
              subPath: models
            - name: churn-volume
              mountPath: /app/registry
              subPath: registry
```



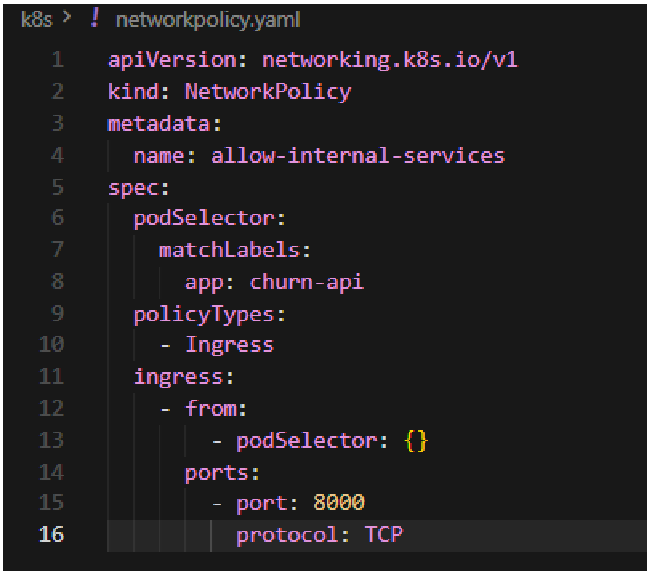
```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/job-train.yaml
job.batch/churn-train created
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl wait --for=condition=complete job/churn-
train
job.batch/churn-train condition met
```



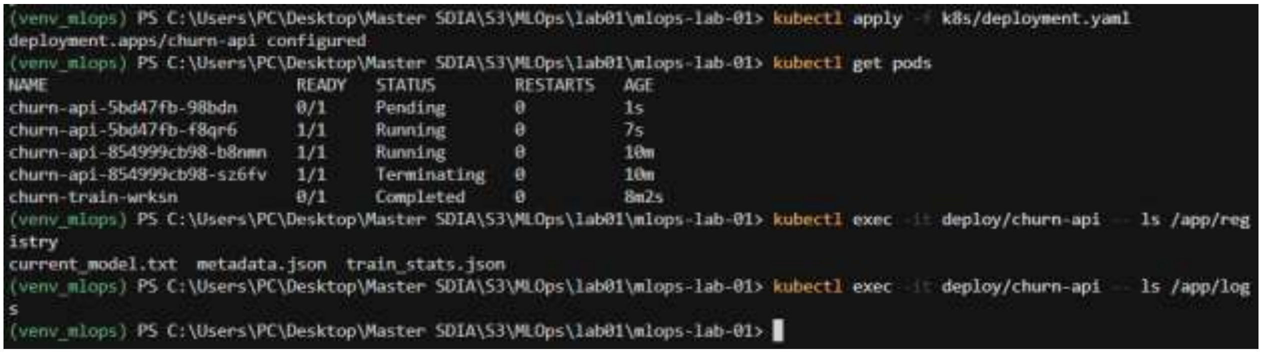
k8s › deployment.yaml

```yaml
spec:
  template:
    spec:
      containers:
        - name: api
            periodSeconds: 10

          startupProbe:
            httpGet:
              path: /startup
              port: 8000
            failureThreshold: 30
            periodSeconds: 5
      volumes:
        - name: churn-volume
          persistentVolumeClaim:
            claimName: churn-storage
```

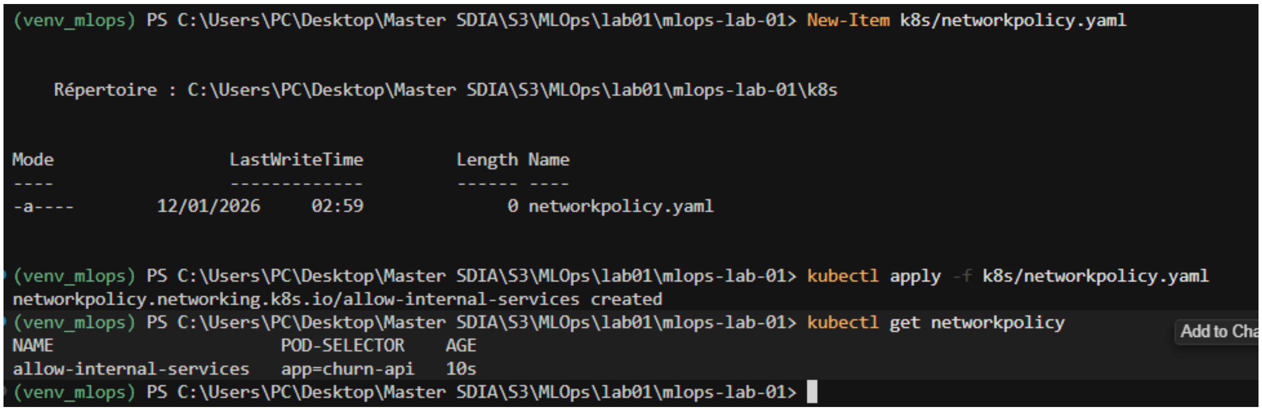
Étape 13 : NetworkPolicy



```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-internal-services
spec:
  podSelector:
    matchLabels:
      app: churn-api
  policyTypes:
    - Ingress
  ingress:
    - from:
        - podSelector: {}
      ports:
        - port: 8000
          protocol: TCP
```

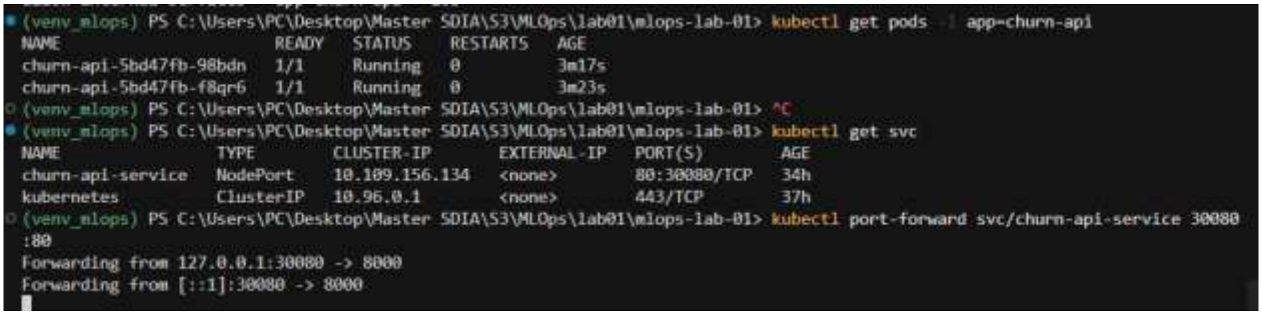


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/deployment.yaml
deployment.apps/churn-api configured
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get pods
NAME                         READY   STATUS        RESTARTS   AGE
churn-api-5bd47fb-98bdn      0/1     Pending       0          1s
churn-api-5bd47fb-f8qr6      1/1     Running       0          7s
churn-api-854999cb98-b8nmn   1/1     Running       0          10m
churn-api-854999cb98-sz6fv   1/1     Terminating   0          10m
churn-train-wrksn            0/1     Completed     0          8m2s
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl exec -it deploy/churn-api -- ls /app/reg
istry
current_model.txt  metadata.json  train_stats.json
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl exec -it deploy/churn-api -- ls /app/log
s
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> 
```

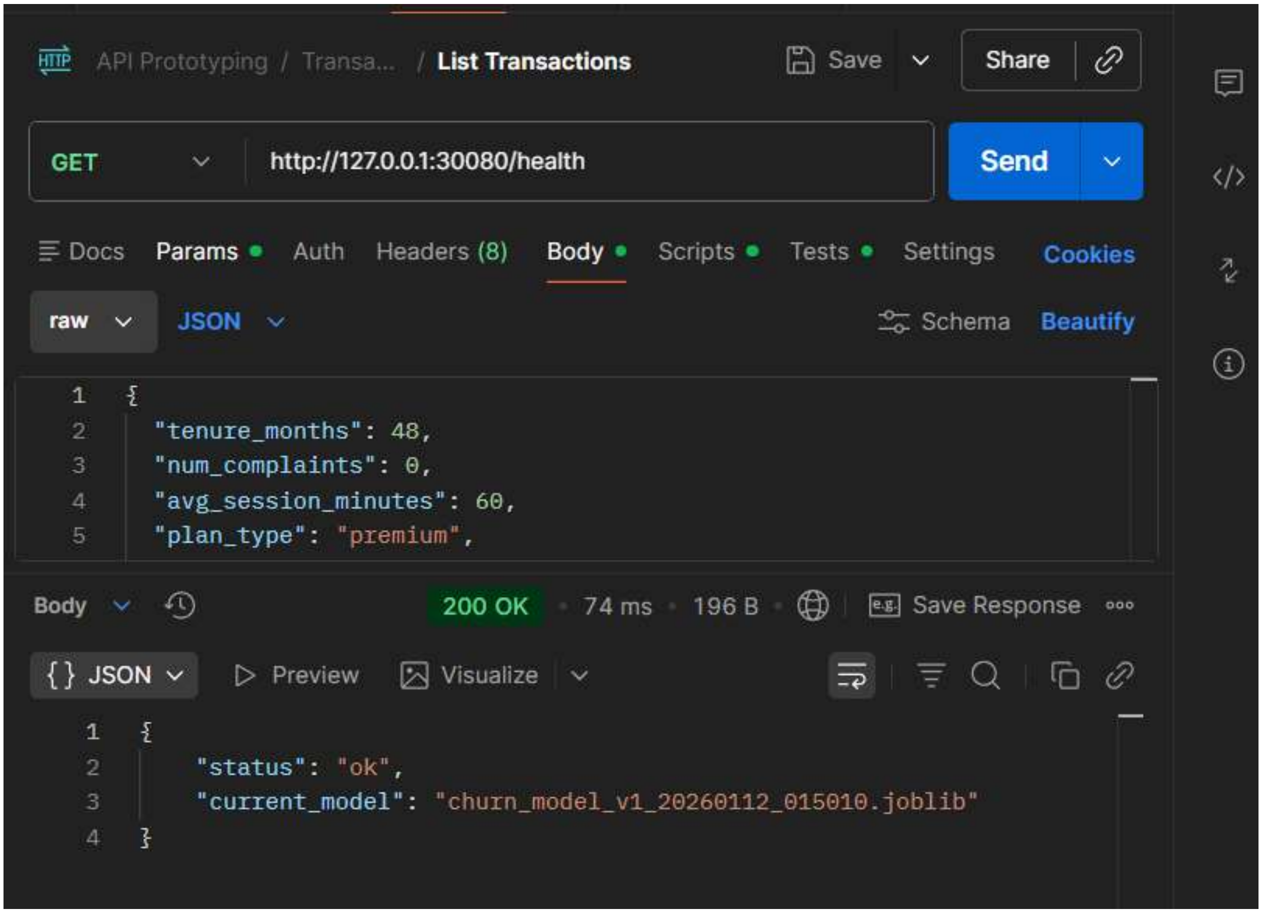


```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> New-Item k8s/networkpolicy.yaml


    Répertoire : C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01\k8s


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        12/01/2026     02:59              0 networkpolicy.yaml


(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl apply -f k8s/networkpolicy.yaml
networkpolicy.networking.k8s.io/allow-internal-services created
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get networkpolicy
NAME                      POD-SELECTOR    AGE
allow-internal-services   app=churn-api   10s
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> 
```

Étape 14 : Vérifications finales



```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get pods -l app=churn-api
NAME                        READY   STATUS    RESTARTS   AGE
churn-api-5bd47fb-98bdn     1/1     Running   0          3m17s
churn-api-5bd47fb-f8qr6     1/1     Running   0          3m23s
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> ^C
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get svc
NAME                TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
churn-api-service   NodePort    10.109.156.134   <none>        80:30080/TCP   34h
kubernetes          ClusterIP   10.96.0.1        <none>        443/TCP        37h
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl port-forward svc/churn-api-service 30080
:80
Forwarding from 127.0.0.1:30080 -> 8000
Forwarding from [::1]:30080 -> 8000
```

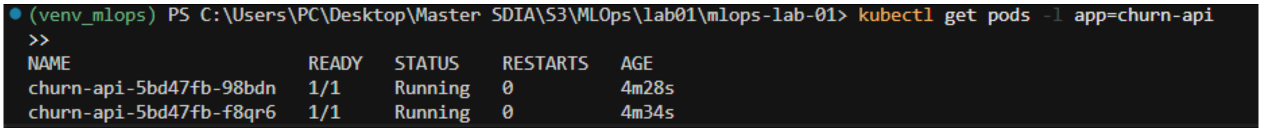


GET http://127.0.0.1:30080/health

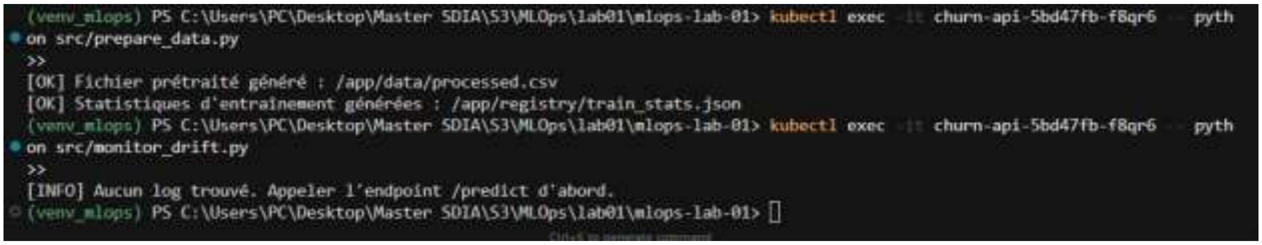
```
{
  "tenure_months": 48,
  "num_complaints": 0,
  "avg_session_minutes": 60,
  "plan_type": "premium",
```

Body · 200 OK · 74 ms · 196 B

```
{
    "status": "ok",
    "current_model": "churn_model_v1_20260112_015010.joblib"
}
```



```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl get pods -l app=churn-api
>>
NAME                      READY   STATUS    RESTARTS   AGE
churn-api-5bd47fb-98bdn   1/1     Running   0          4m28s
churn-api-5bd47fb-f8qr6   1/1     Running   0          4m34s
```



```
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl exec -it churn-api-5bd47fb-f8qr6 -- pyth
on src/prepare_data.py
>>
[OK] Fichier prétraité généré : /app/data/processed.csv
[OK] Statistiques d'entrainement générées : /app/registry/train_stats.json
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> kubectl exec -it churn-api-5bd47fb-f8qr6 -- pyth
on src/monitor_drift.py
>>
[INFO] Aucun log trouvé. Appeler l'endpoint /predict d'abord.
(venv_mlops) PS C:\Users\PC\Desktop\Master SDIA\S3\MLOps\lab01\mlops-lab-01> 
```