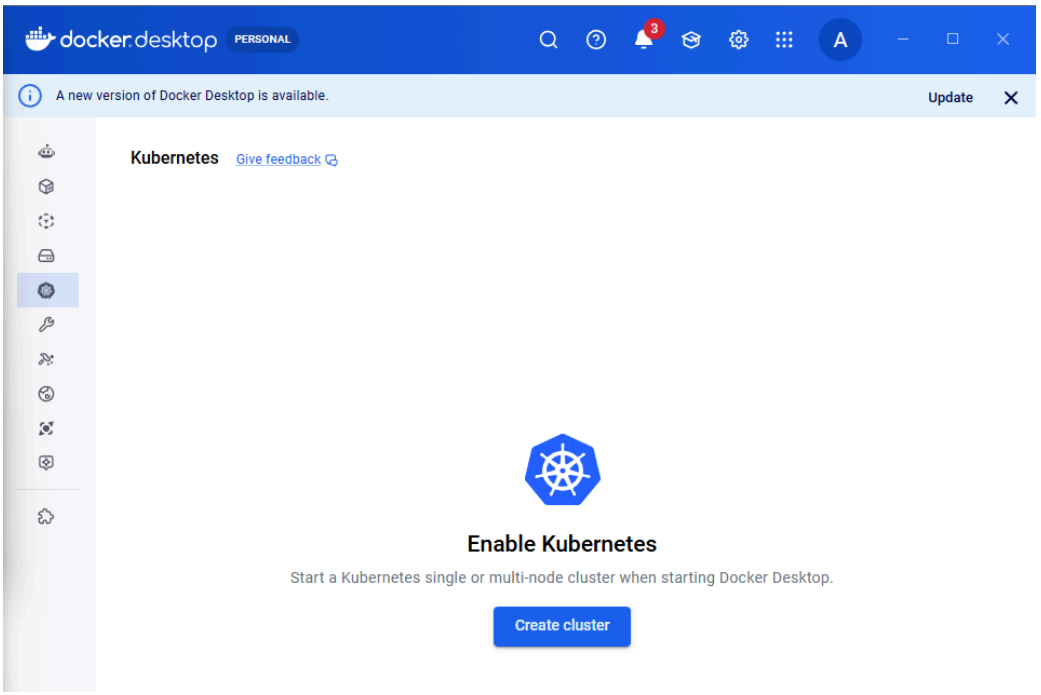
Chapitre 5 – Kubernetes : plateforme de déploiement des applications conteneurisées

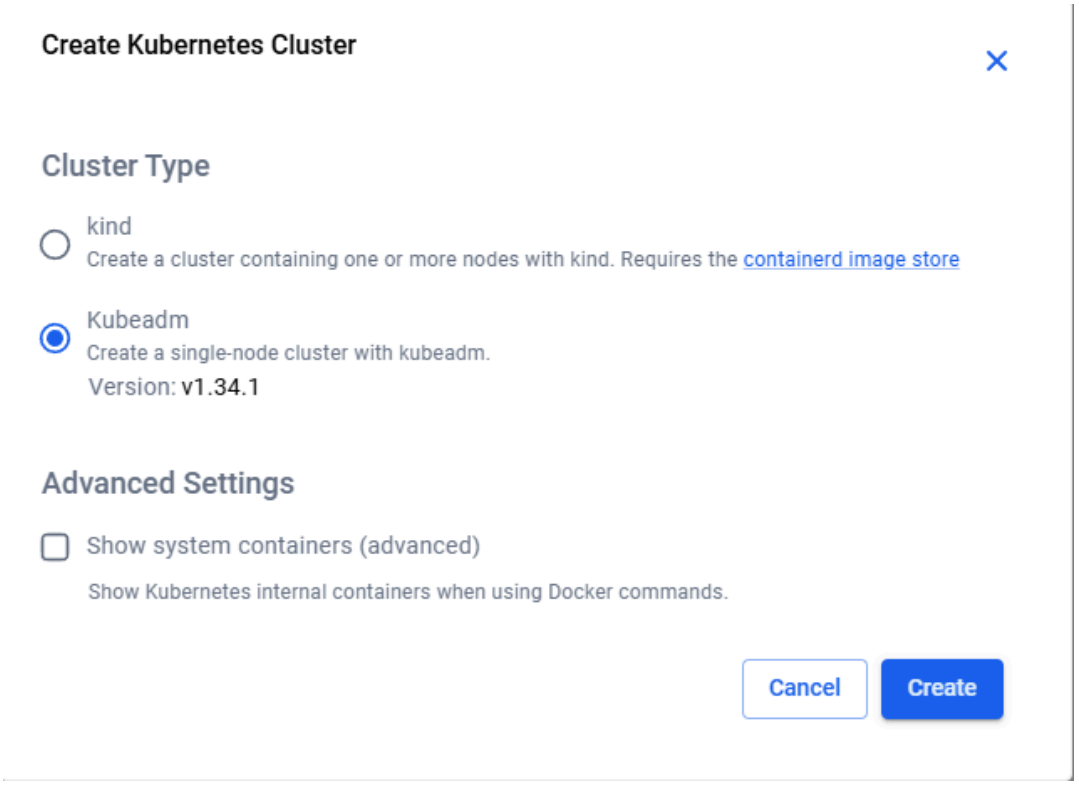
Table des Matières :

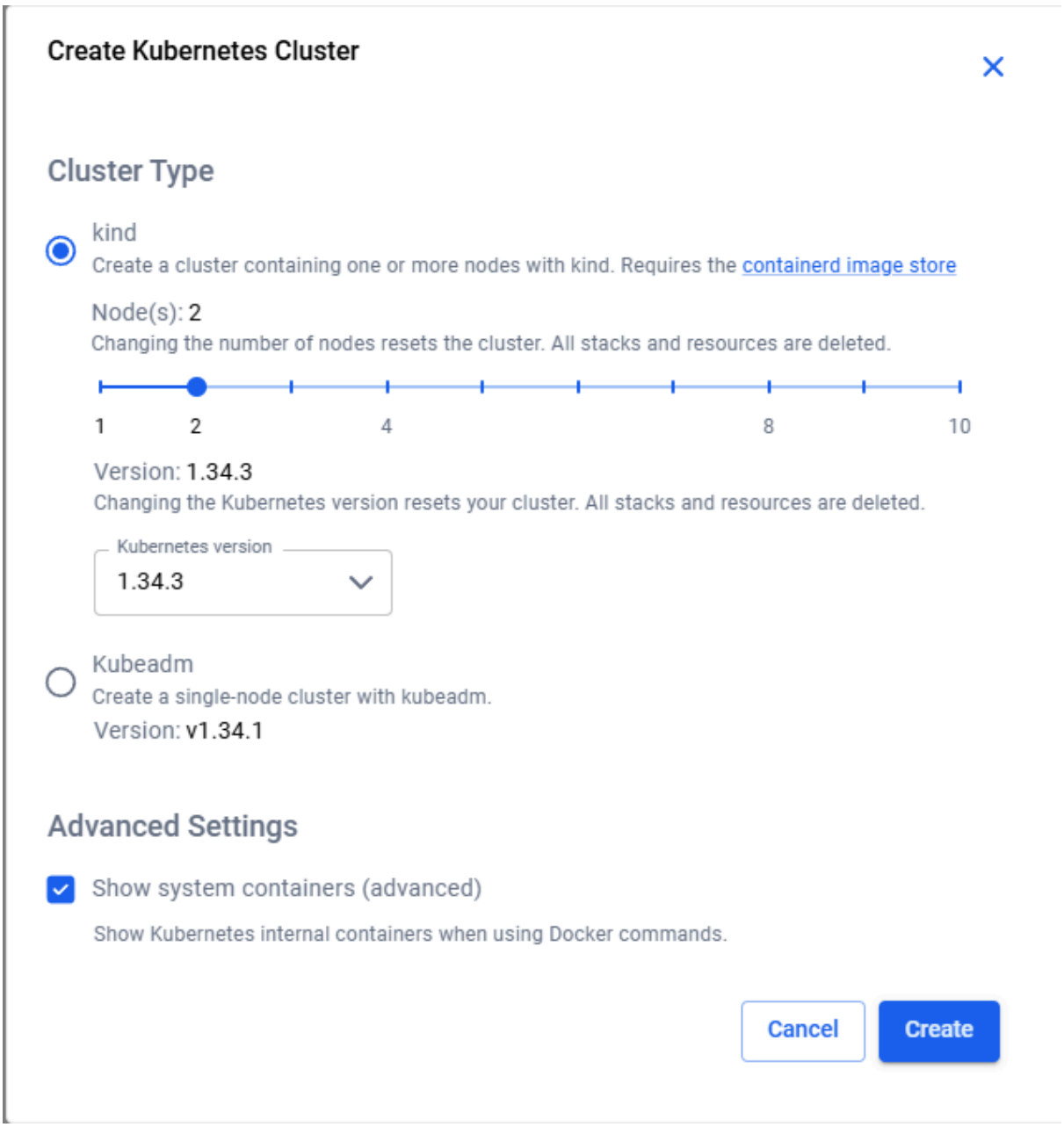
1. Kubernetes avec Docker Desktop.

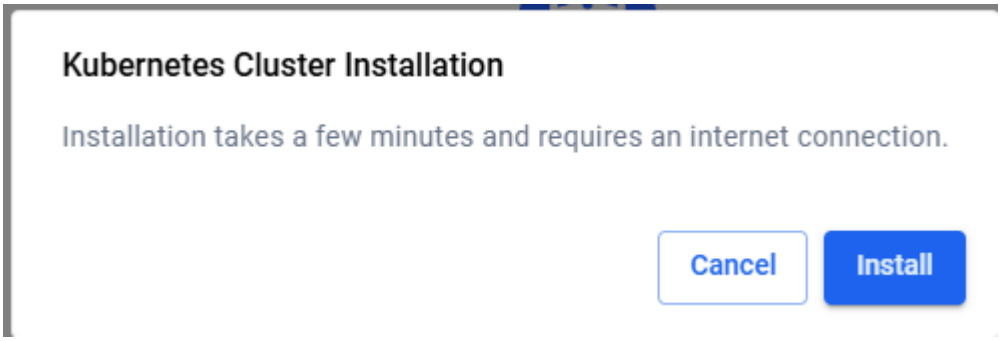
▪ Découverte de Kubernetes sur Docker :

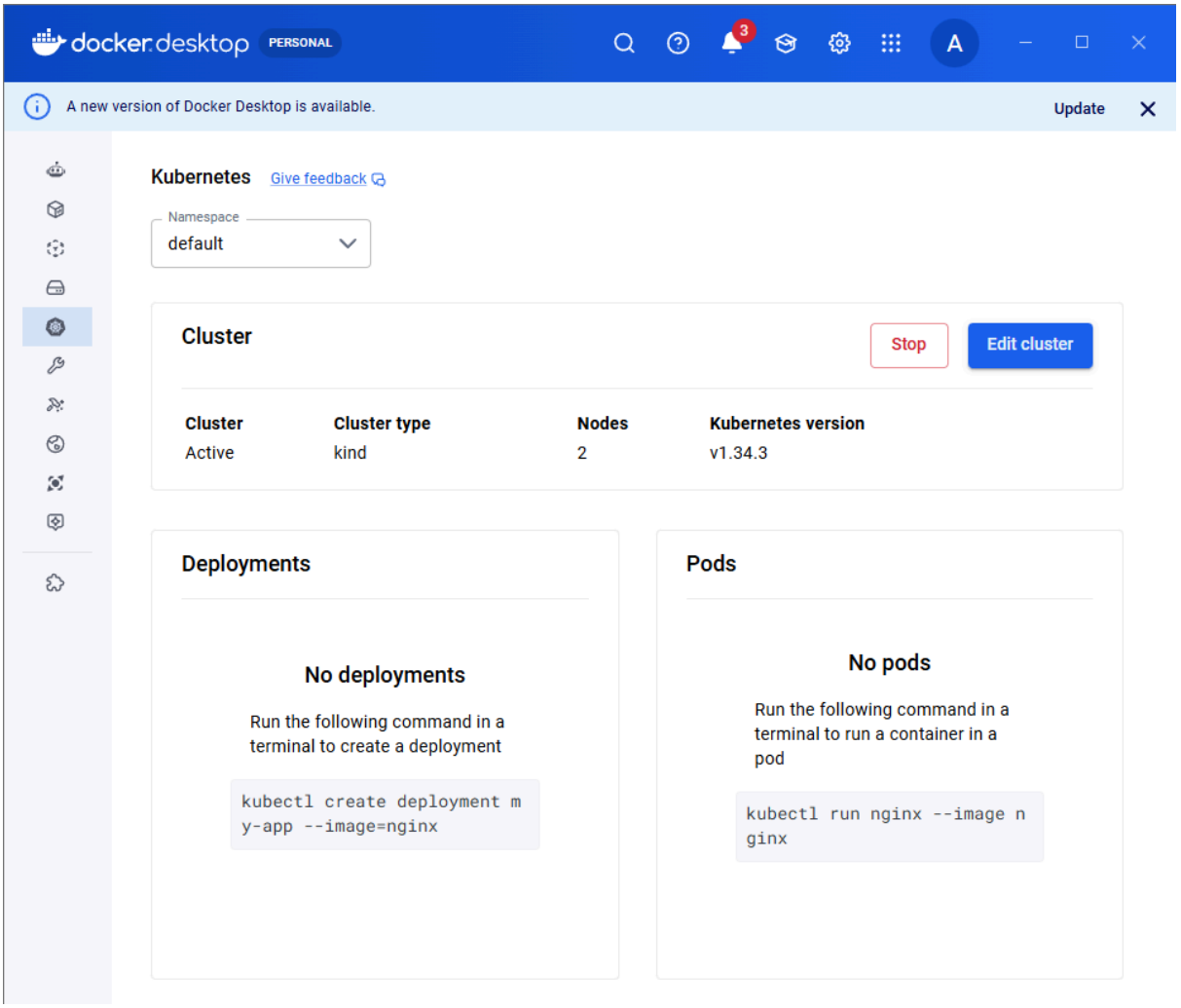


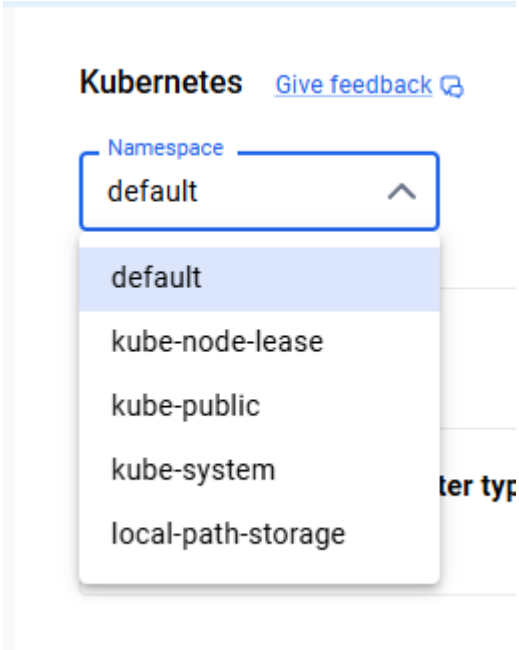
▪ Création d'un cluster :




Create Kubernetes Cluster
Cluster Type
kind
Create a cluster containing one or more nodes with kind. Requires the containerd image store
Node(s): 2
Changing the number of nodes resets the cluster. All stacks and resources are deleted.
1
2
4
8
10
Version: 1.34.3
Changing the Kubernetes version resets your cluster. All stacks and resources are deleted.
Kubernetes version
1.34.3
Kubeadm
Create a single-node cluster with kubeadm.
Version: v1.34.1
Advanced Settings
Show system containers (advanced)
Show Kubernetes internal containers when using Docker commands.
Cancel
Create

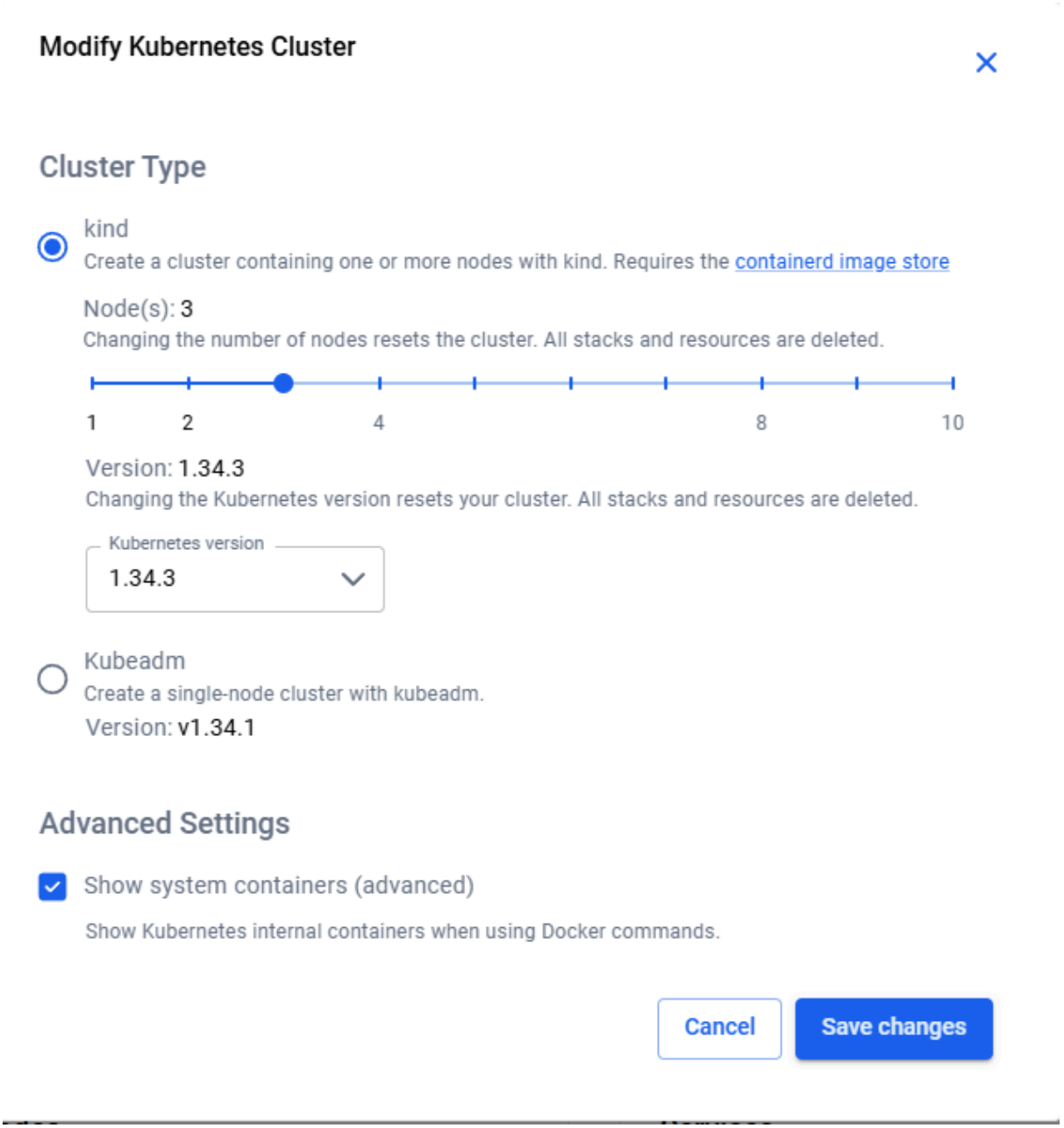

Kubernetes Cluster Installation
Installation takes a few minutes and requires an internet connection.
Cancel
Install

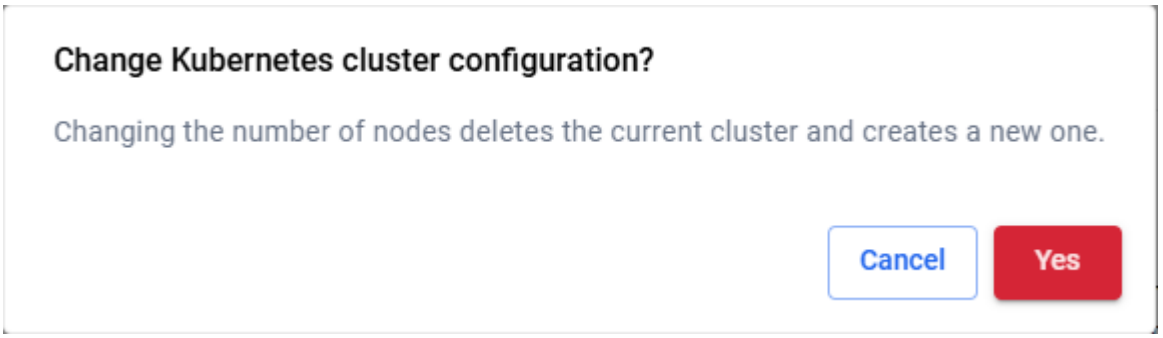

docker desktop
PERSONAL
A new version of Docker Desktop is available.
Update
Kubernetes
Give feedback
Namespace
default
Cluster
Stop
Edit cluster
Cluster
Active
Cluster type
kind
Nodes
2
Kubernetes version
v1.34.3
Deployments
No deployments
Run the following command in a terminal to create a deployment
kubectl create deployment my-app --image=nginx
Pods
No pods
Run the following command in a terminal to run a container in a pod
kubectl run nginx --image nginx

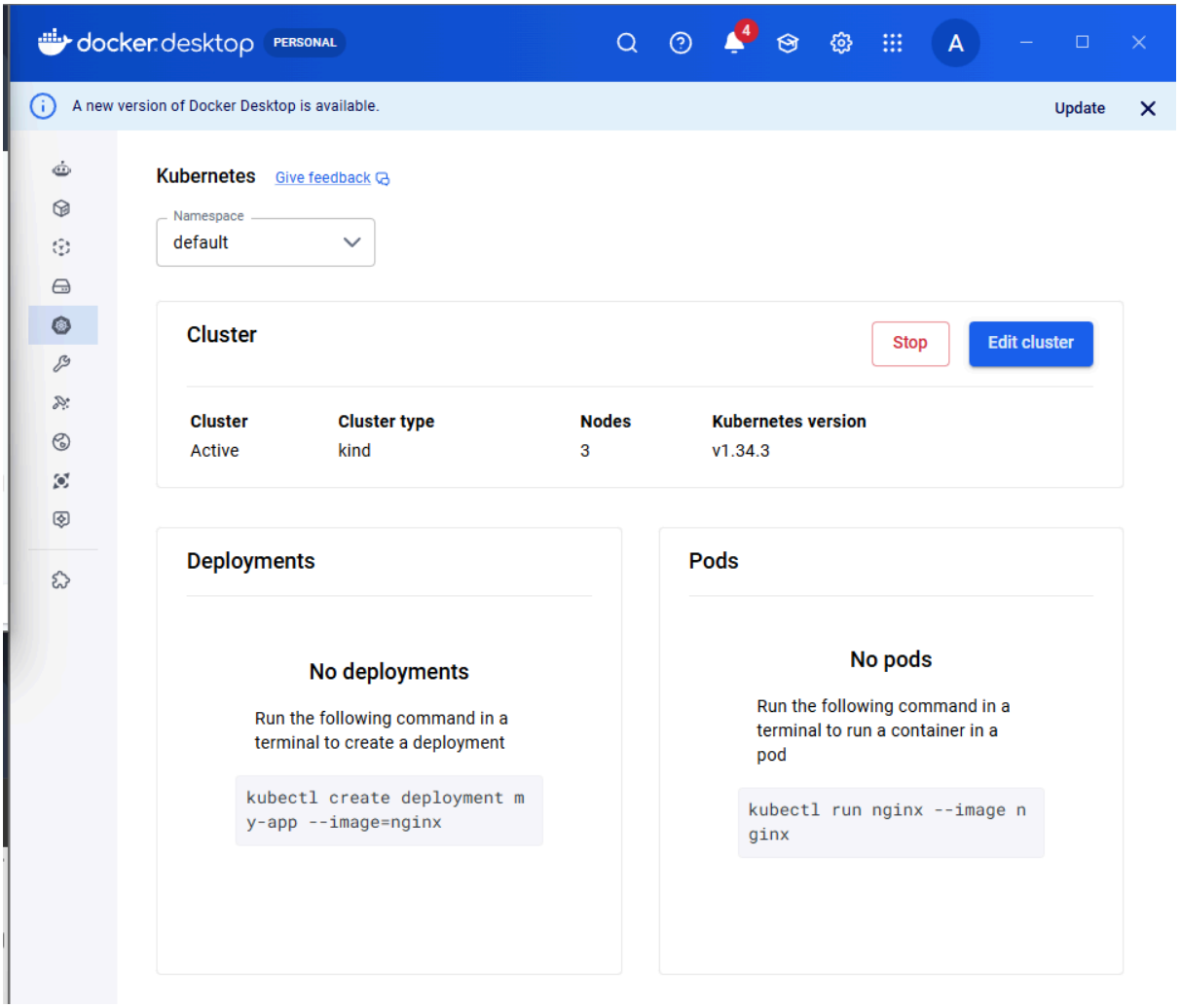

Kubernetes
Give feedback
Namespace
default
default
kube-node-lease
kube-public
kube-system
local-path-storage

▪ Nous constatons l'état des machines

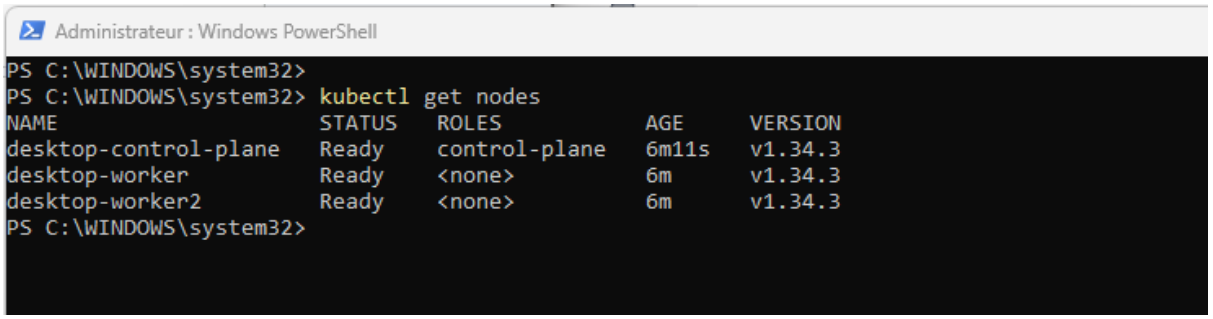
```
Administrateur : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités
s

PS C:\WINDOWS\system32> kubectl get nodes
NAME                   STATUS   ROLES           AGE   VERSION
desktop-control-plane   Ready    control-plane   84s   v1.34.3
desktop-worker          Ready    <none>          73s   v1.34.3
PS C:\WINDOWS\system32>
```

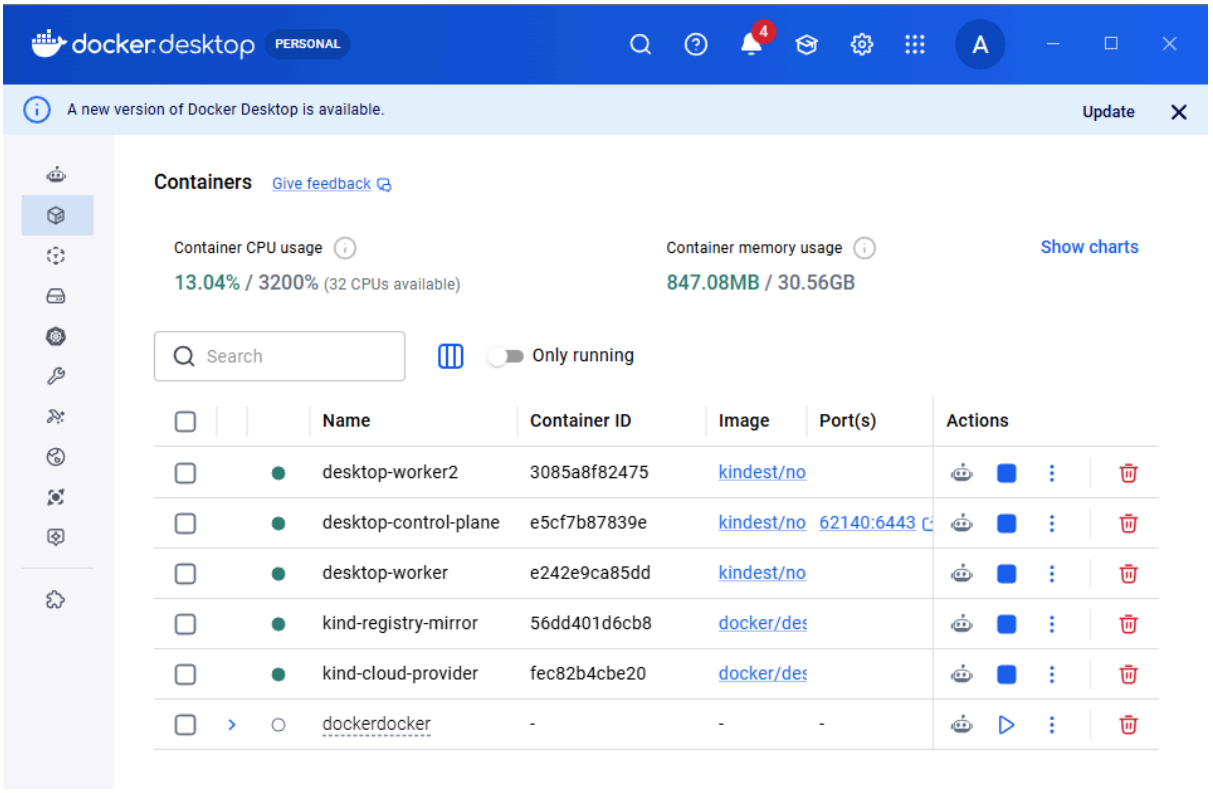



Change Kubernetes cluster configuration?
Changing the number of nodes deletes the current cluster and creates a new one.
Cancel
Yes


docker desktop
PERSONAL
A new version of Docker Desktop is available.
Update
Kubernetes
Give feedback
Namespace
default
Cluster
Stop
Edit cluster
Cluster
Active
Cluster type
kind
Nodes
3
Kubernetes version
v1.34.3
Deployments
No deployments
Run the following command in a terminal to create a deployment
kubectl create deployment my-app --image=nginx
Pods
No pods
Run the following command in a terminal to run a container in a pod
kubectl run nginx --image nginx

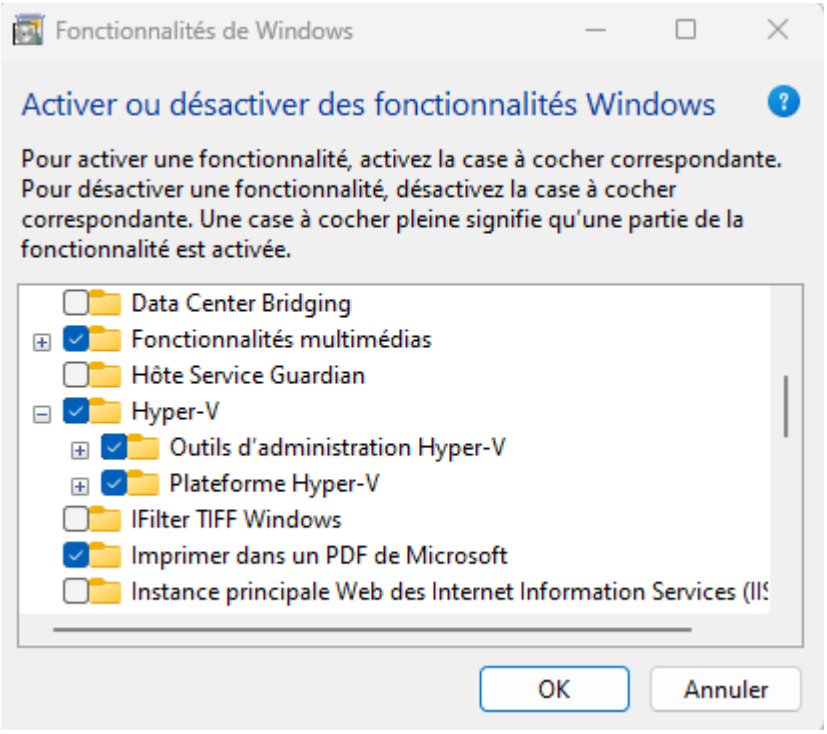
▪ Nous pouvons trouver ci-dessous un maître et deux workers :



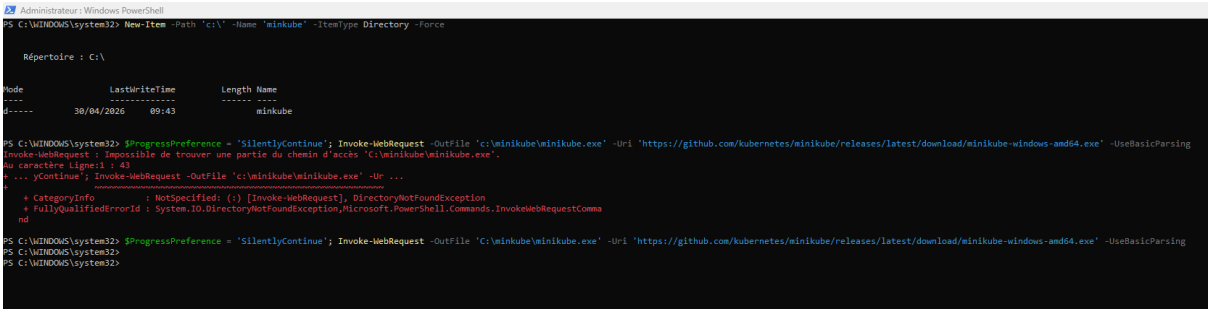


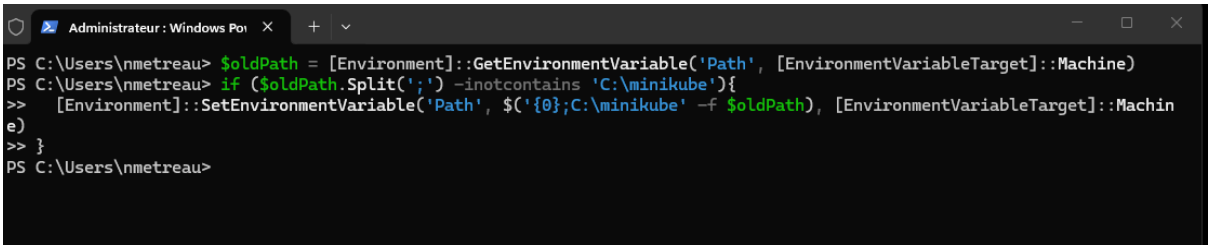
2. Installation de Minikube.

▪ Nous activons la fonctionnalité Windows Hyper-V :



- Nous installons minikube





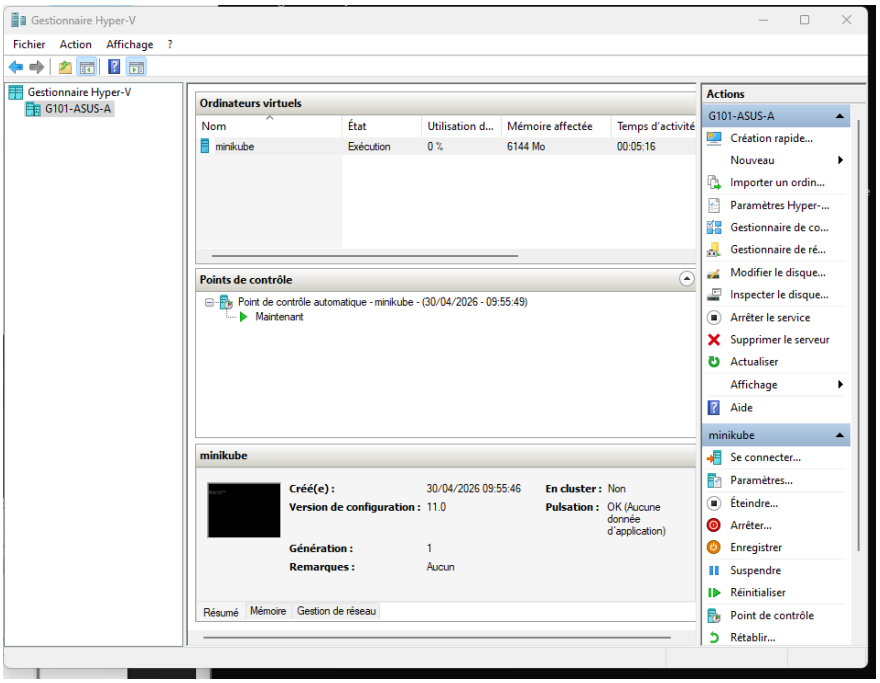
▪ Nous lançons minikube à l'aide de la commande "minikube start" :

```
Administrateur : Invite de commandes
Microsoft Windows [version 10.0.22631.6199]
(c) Microsoft Corporation. Tous droits réservés.

C:\Windows\System32>cd C:\minkube

C:\minkube>minikube start
* minikube v1.38.1 sur Microsoft Windows 11 Pro 23H2
* Choix automatique du pilote hyperv. Autres choix: virtualbox, ssh
! Starting v1.39.0, minikube will default to "containerd" container runtime. See #21973 for more info.
* Téléchargement de l'image de démarrage de la VM...
    > minikube-v1.38.0-amd64.iso....:  65 B / 65 B [---------] 100.00% ? p/s 0s
    > minikube-v1.38.0-amd64.iso:  370.55 MiB / 370.55 MiB  100.00% 10.91 MiB p
* Démarrage du nœud "minikube" primary control-plane dans le cluster "minikube"
* Téléchargement du préchargement de Kubernetes v1.35.1...
    > preloaded-images-k8s-v18-v1...:  272.45 MiB / 272.45 MiB  100.00% 10.76 M
* Création de VM hyperv (CPUs=2, Mémoire=6144MB, Disque=20000MB)...
* Préparation de Kubernetes v1.35.1 sur Docker 28.5.2...
* 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
* Terminé ! kubectl est maintenant configuré pour utiliser "minikube" cluster et espace de noms "default" par défaut.

C:\minkube>
```

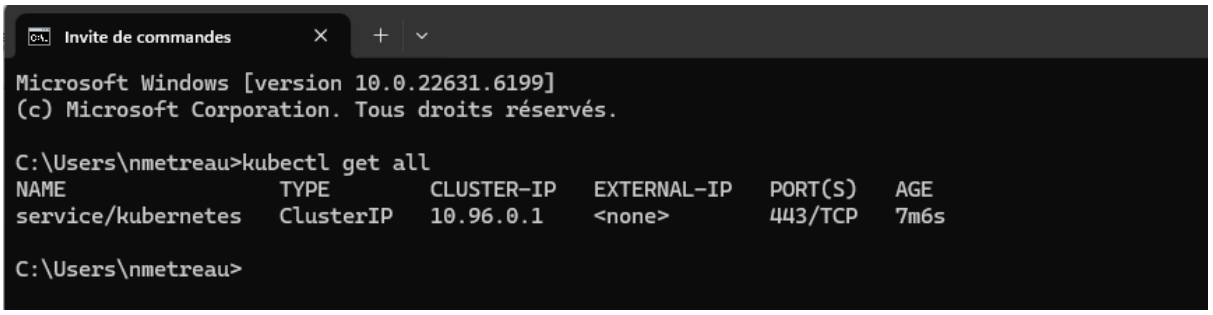
▪ Nous vérifions l'état de la machine :

```
C:\minkube>kubectl get nodes
NAME       STATUS   ROLES           AGE     VERSION
minikube   Ready    control-plane   3m33s   v1.35.1

C:\minkube>
```

▪ Nous constatons la création de la vm dans HyperV



- Commande pour lister les objets :



- Création d'un pod nginx via kubectl :

```
C:\Users\nmetreau>kubectl run monpod --image=nginx:latest
pod/monpod created

C:\Users\nmetreau>
```

```
C:\Users\nmetreau>kubectl get pods
NAME     READY   STATUS    RESTARTS   AGE
monpod   1/1     Running   0          28s

C:\Users\nmetreau>
```

```
C:\Users\nmetreau>kubectl get all
NAME         READY   STATUS    RESTARTS   AGE
pod/monpod   1/1     Running   0          48s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   8m47s

C:\Users\nmetreau>
```

- Nous consultons le log d'un pod :

```
C:\Users\nmetreau>kubectl logs pod/monpod
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2026/04/30 08:04:27 [notice] 1#1: using the "epoll" event method
2026/04/30 08:04:27 [notice] 1#1: nginx/1.29.8
2026/04/30 08:04:27 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2026/04/30 08:04:27 [notice] 1#1: OS: Linux 6.6.95
2026/04/30 08:04:27 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2026/04/30 08:04:27 [notice] 1#1: start worker processes
2026/04/30 08:04:27 [notice] 1#1: start worker process 29
2026/04/30 08:04:27 [notice] 1#1: start worker process 30

C:\Users\nmetreau>
```

- Les espaces de noms dans lesquels sont créés les objets :

```
C:\Users\nmetreau>kubectl get namespace
NAME              STATUS   AGE
default           Active   9m47s
kube-node-lease   Active   9m47s
kube-public       Active   9m47s
kube-system       Active   9m47s

C:\Users\nmetreau>
```

- Les objets : singulier ou pluriel, peu importe

```
C:\Users\nmetreau>kubectl get pods
NAME     READY   STATUS    RESTARTS   AGE
monpod   1/1     Running   0          2m29s

C:\Users\nmetreau>
```

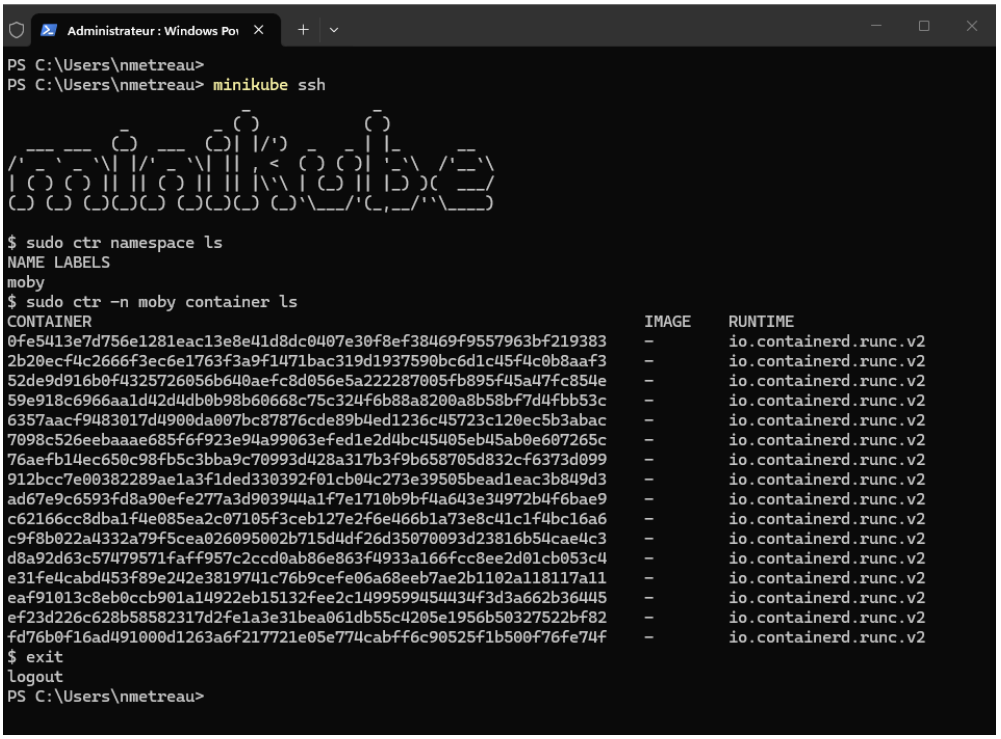
```
C:\Users\nmetreau>kubectl get pod
NAME     READY   STATUS    RESTARTS   AGE
monpod   1/1     Running   0          2m46s

C:\Users\nmetreau>
```

```
C:\Users\nmetreau>kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   11m   v1.35.1

C:\Users\nmetreau>kubectl get node
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   11m   v1.35.1

C:\Users\nmetreau>
```

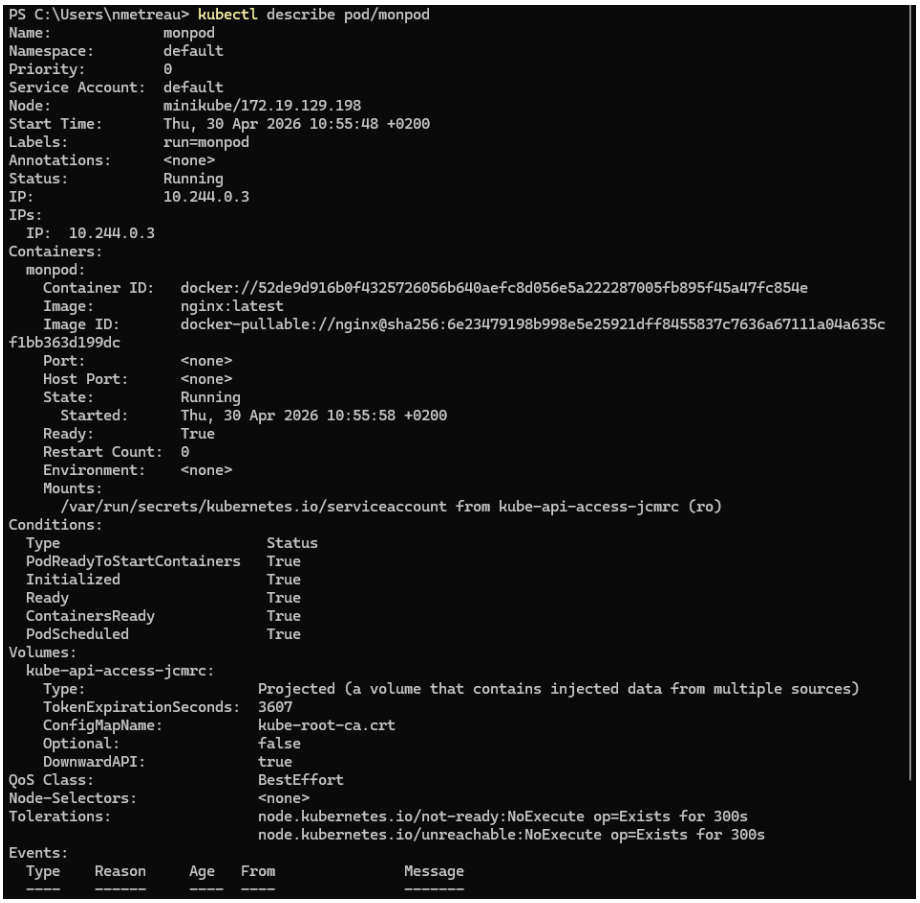
- Champs supplémentaires :

```
C:\Users\nmetreau>kubectl get node -o wide
NAME       STATUS   ROLES           AGE   VERSION   INTERNAL-IP      EXTERNAL-IP   OS-IMAGE
 KERNEL-VERSION   CONTAINER-RUNTIME
minikube   Ready    control-plane   11m   v1.35.1   192.168.220.57   <none>        Buildroot 2025.02
 6.6.95           docker://28.5.2
```

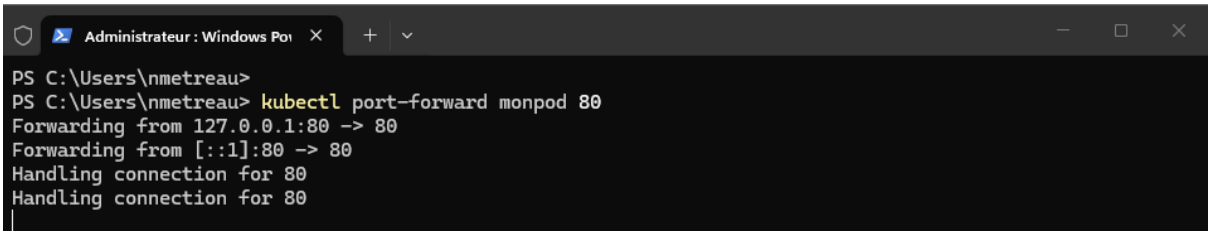
- Nous accédons et interrogeons le démon containerd (commande ctr précédée de sudo afin de récupérer la liste des espaces de noms) :

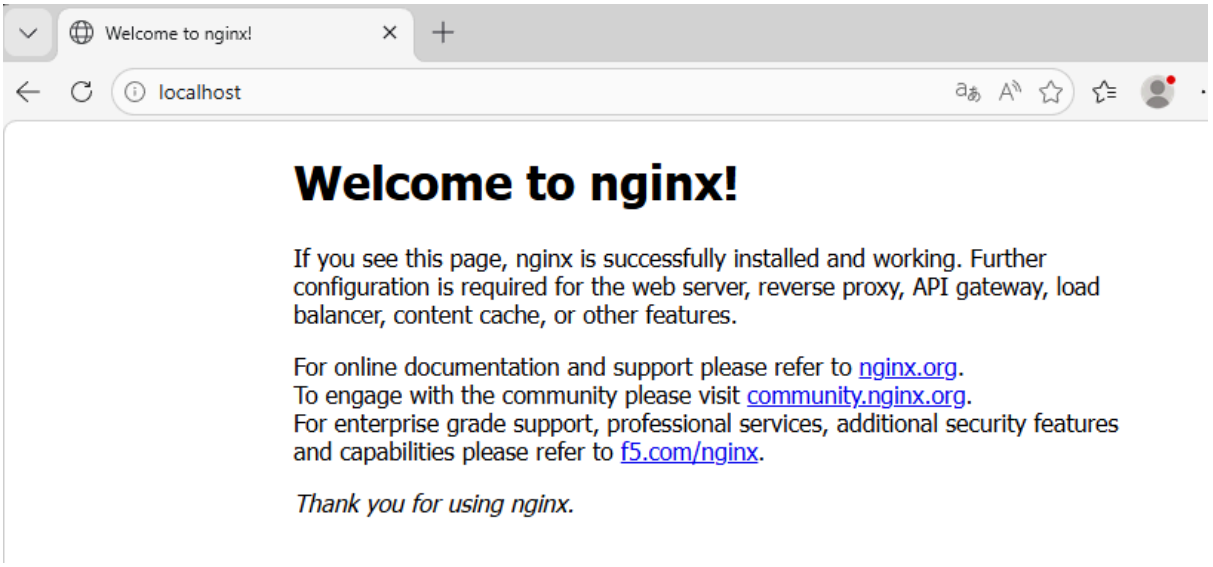


- Description d'un objet :

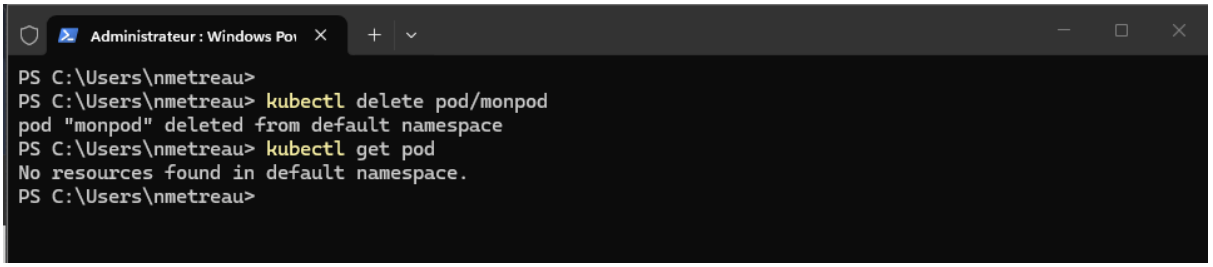


▪ Nous accédons à l'application Nginx depuis l'extérieur du cluster Kubernetes (instruction port-forward) :



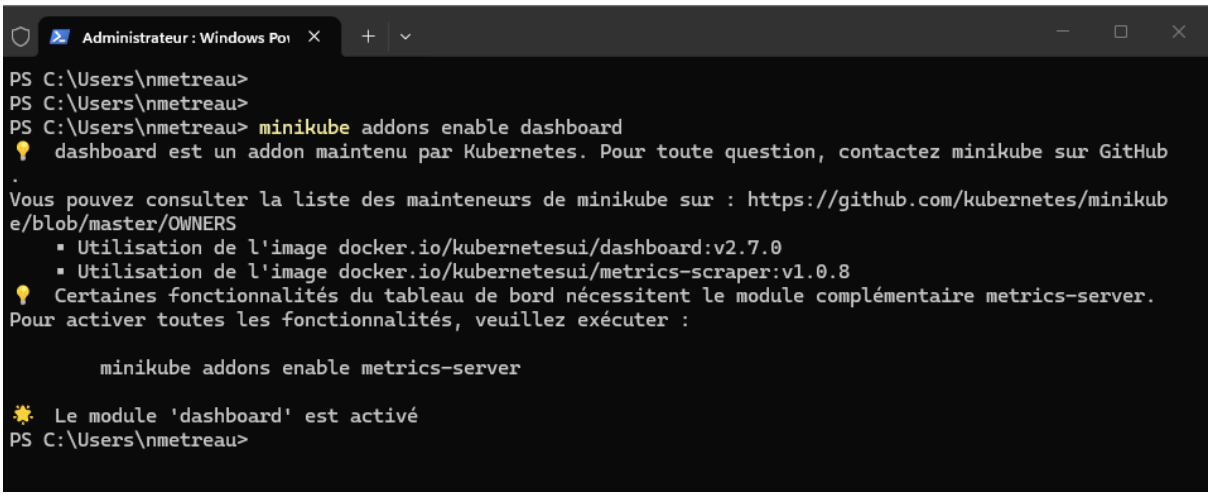


▪ Nous détruisons le pod nginx :

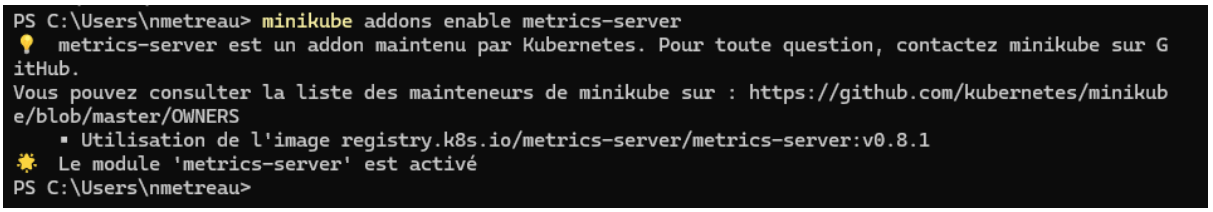


3. Le tableau de bord de Kubernetes.

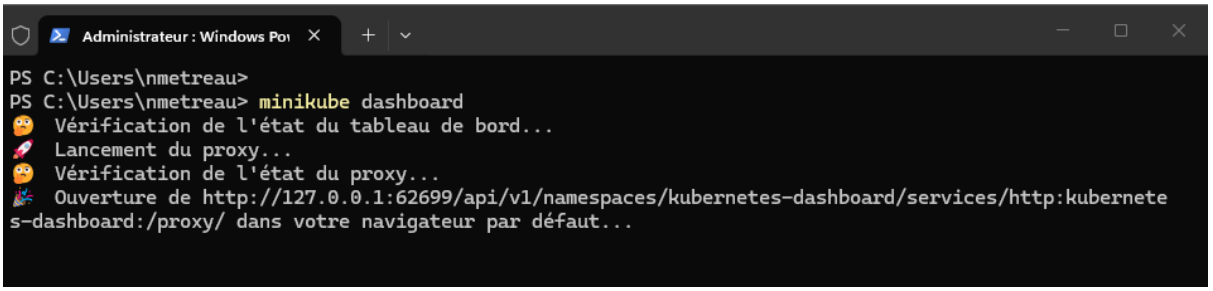
▪ Nous activons le plugin :

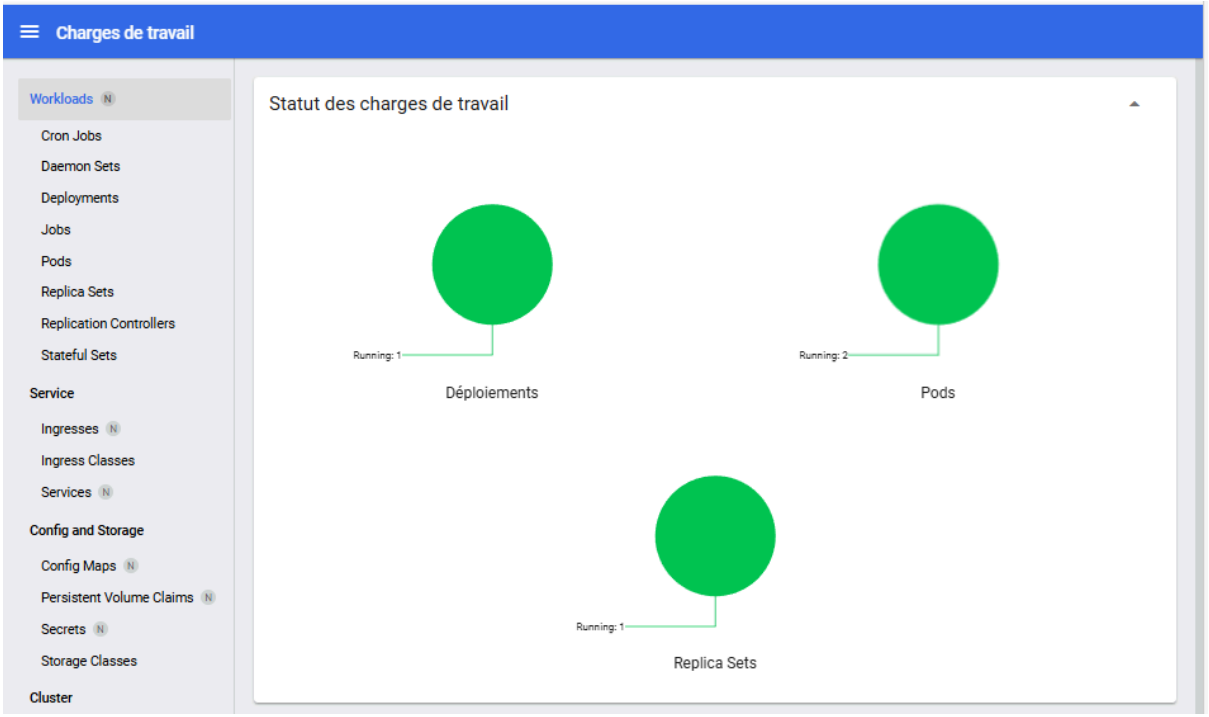


▪ Nous activons le serveur de métriques pour bénéficier de l'ensemble des fonctionnalités du tableau de bord :



▪ Nous accédons au tableau de bord :





Déploiements

Nom	Images	Étiquettes	Pods
mailpit	axllent/mailpit	k8s-app: mailpit	2 / 2

Pods

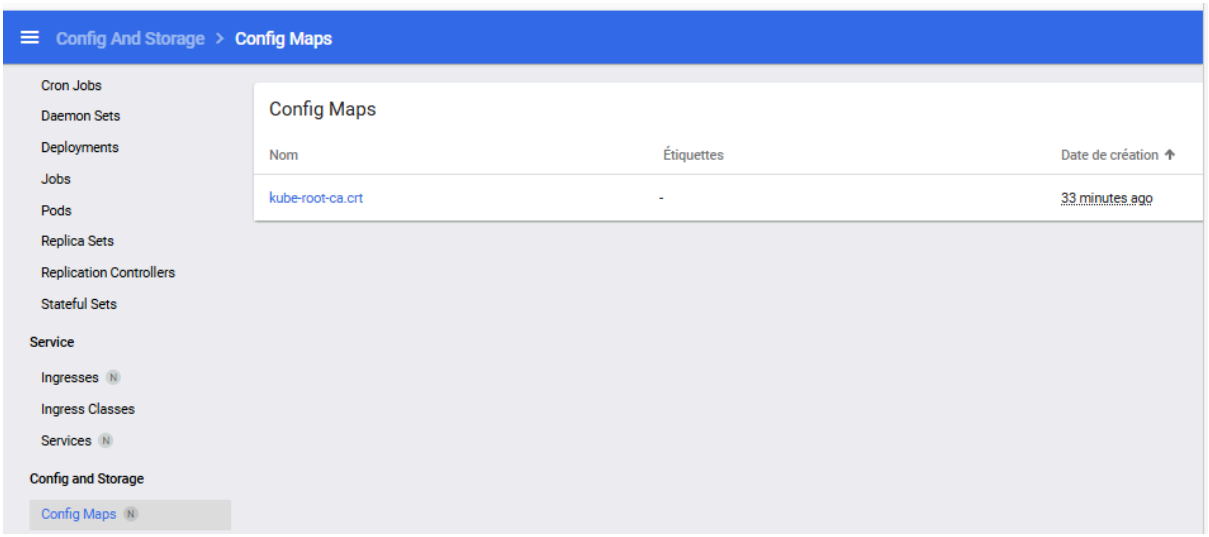
Nom	Images	Étiquettes	Noeud	Statut	Redémarr	Utilisation CPU (coeurs)	Utilisation mémoire (octe
mailpit-5d8c9cd4-4sp4r	axllent/mailpit	k8s-app: mailpit pod-template-hash: 5d8c9cd4	minikube	Running	0	1,00m	7,07Mi
mailpit-5d8c9cd4-gj5xq	axllent/mailpit	k8s-app: mailpit pod-template-hash: 5d8c9cd4	minikube	Running	0	1,00m	7,32Mi

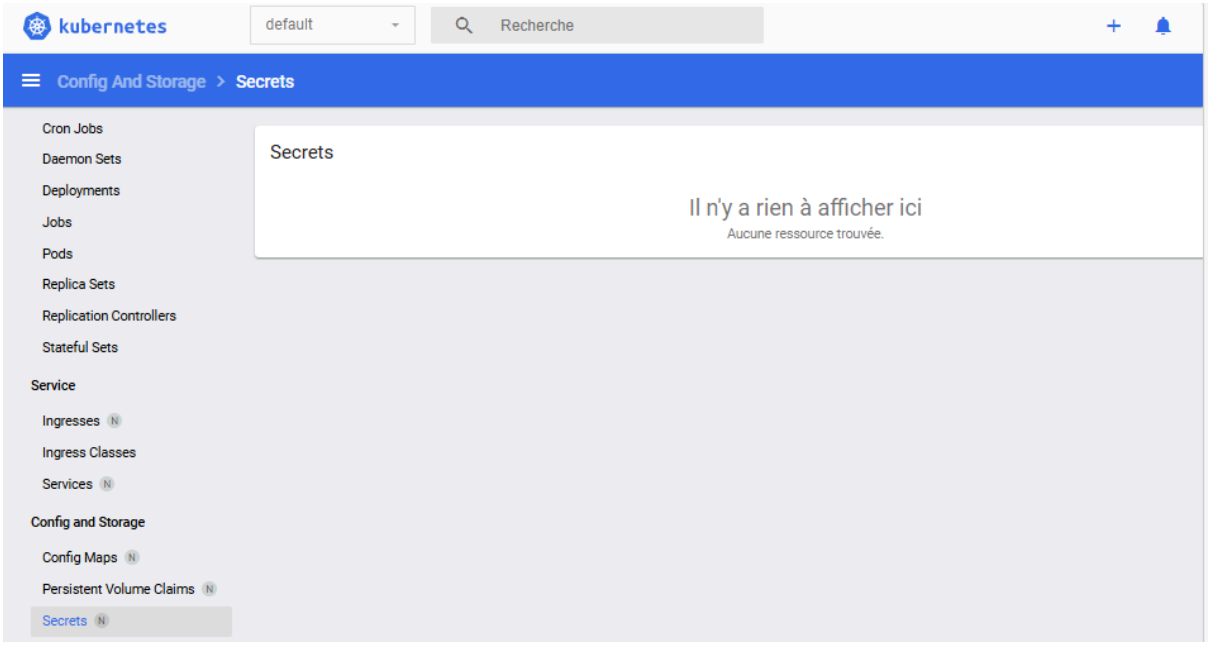
Replica Sets

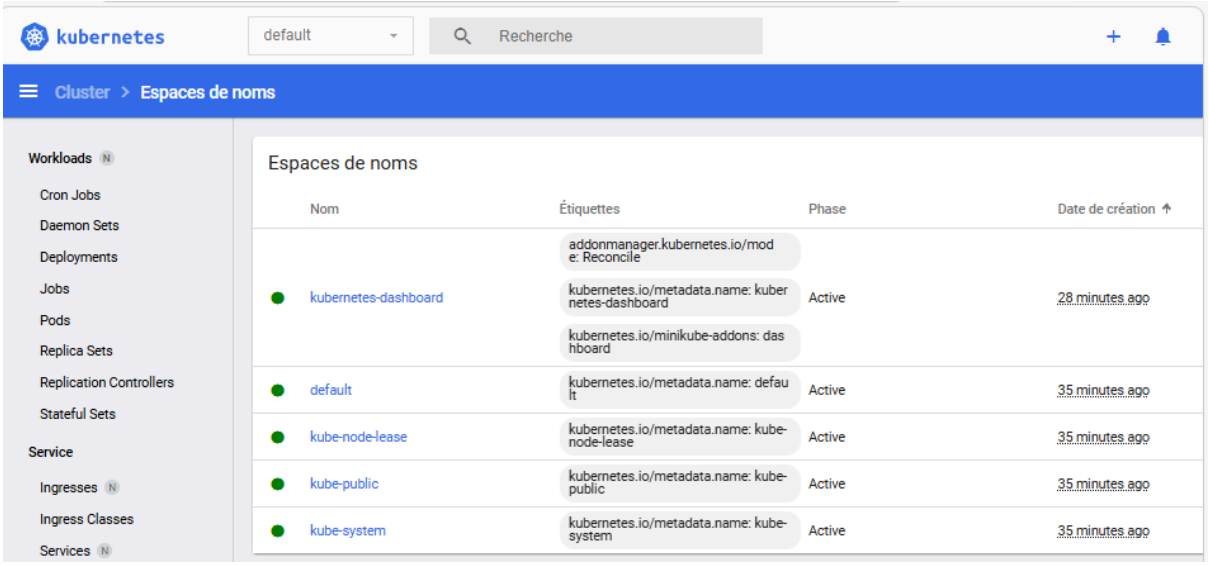
Nom	Images	Étiquettes	Pods
mailpit-5d8c9cd4	axllent/mailpit	k8s-app: mailpit pod-template-hash: 5d8c9cd4	2 / 2

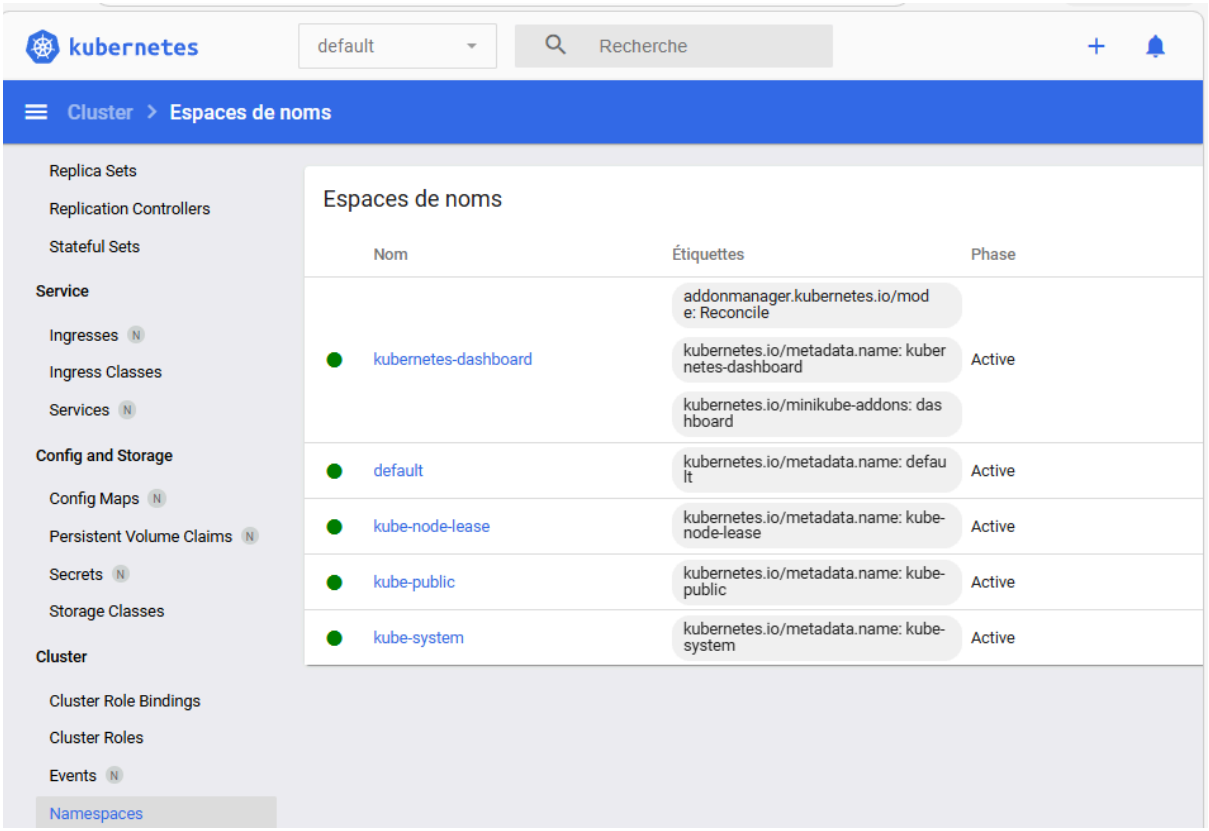
Services

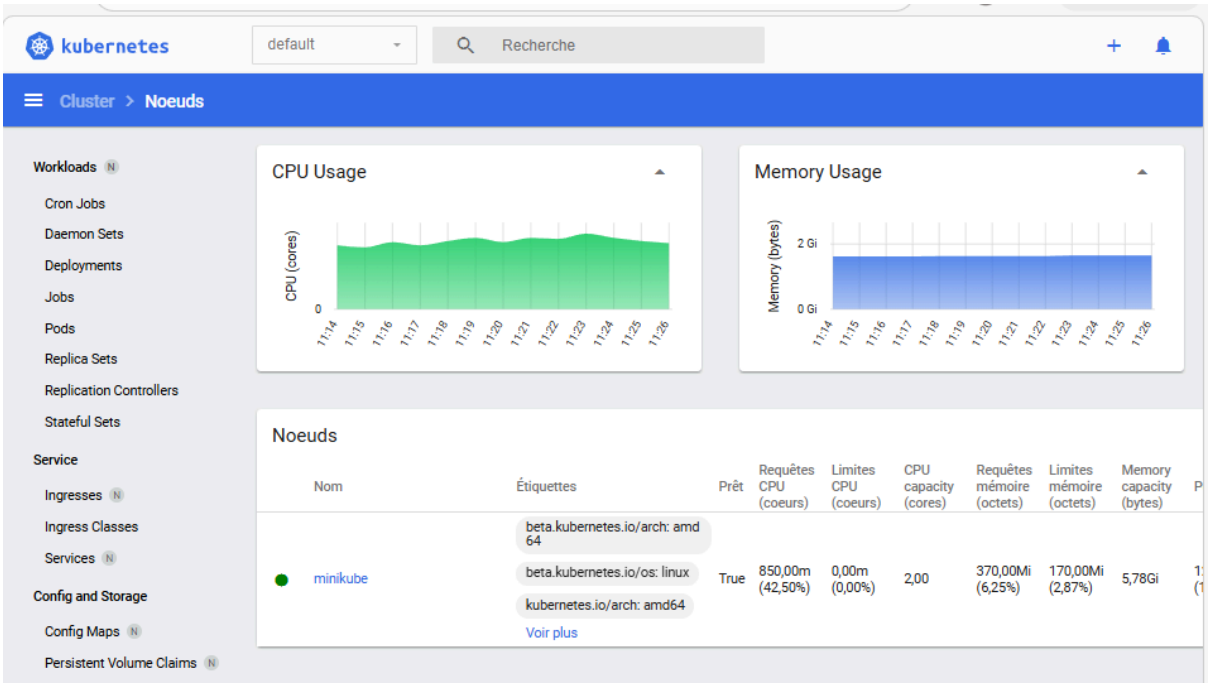
Nom	Étiquettes	Type	IP cluster	Terminaisons internes	Terminaison externes
kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-


Config And Storage > Config Maps
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Config Maps
Nom
Étiquettes
Date de création
kube-root-ca.crt
-
33 minutes ago

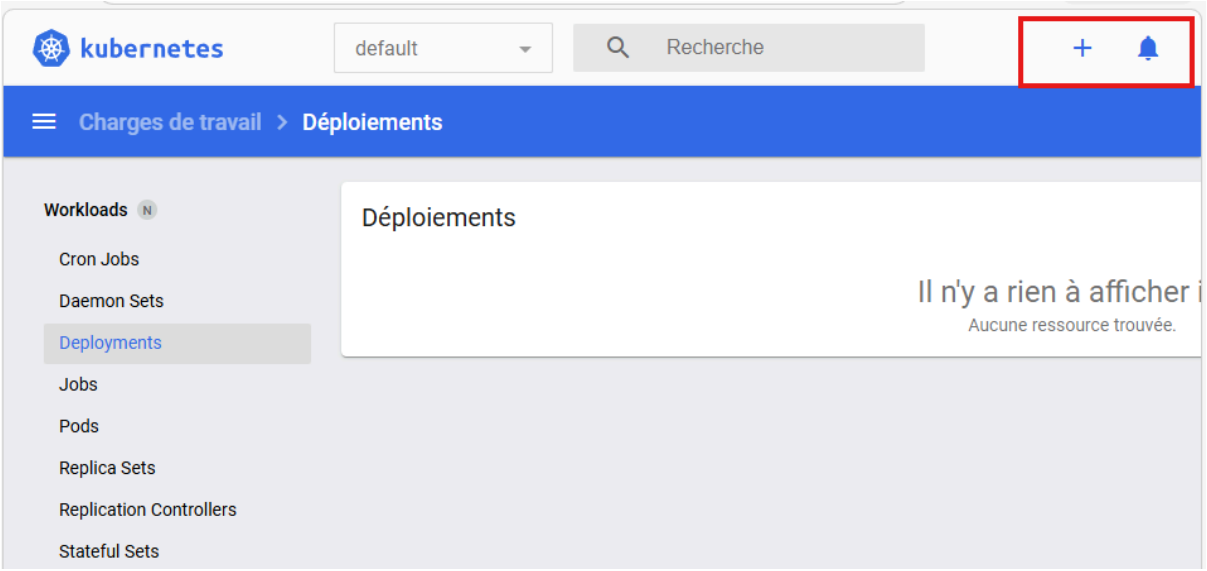

kubernetes
default
Recherche
Config And Storage > Secrets
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Secrets
Il n'y a rien à afficher ici
Aucune ressource trouvée.

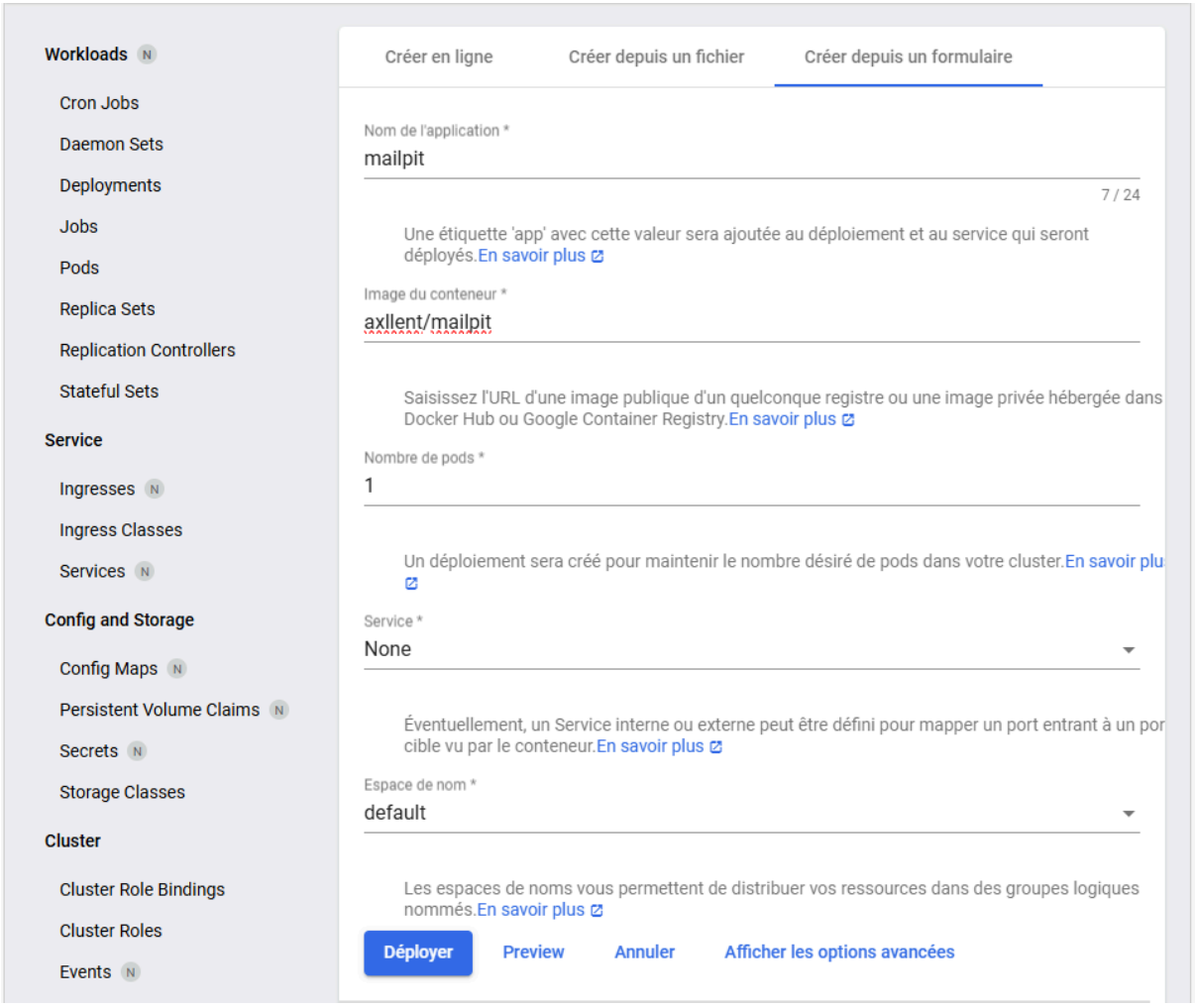

kubernetes
default
Recherche
Cluster > Espaces de noms
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Espaces de noms
Nom
Étiquettes
Phase
Date de création
kubernetes-dashboard
addonmanager.kubernetes.io/mode: Reconcile
kubernetes.io/metadata.name: kubernetes-dashboard
kubernetes.io/minikube-addons: dashboard
Active
28 minutes ago
default
kubernetes.io/metadata.name: default
Active
35 minutes ago
kube-node-lease
kubernetes.io/metadata.name: kube-node-lease
Active
35 minutes ago
kube-public
kubernetes.io/metadata.name: kube-public
Active
35 minutes ago
kube-system
kubernetes.io/metadata.name: kube-system
Active
35 minutes ago

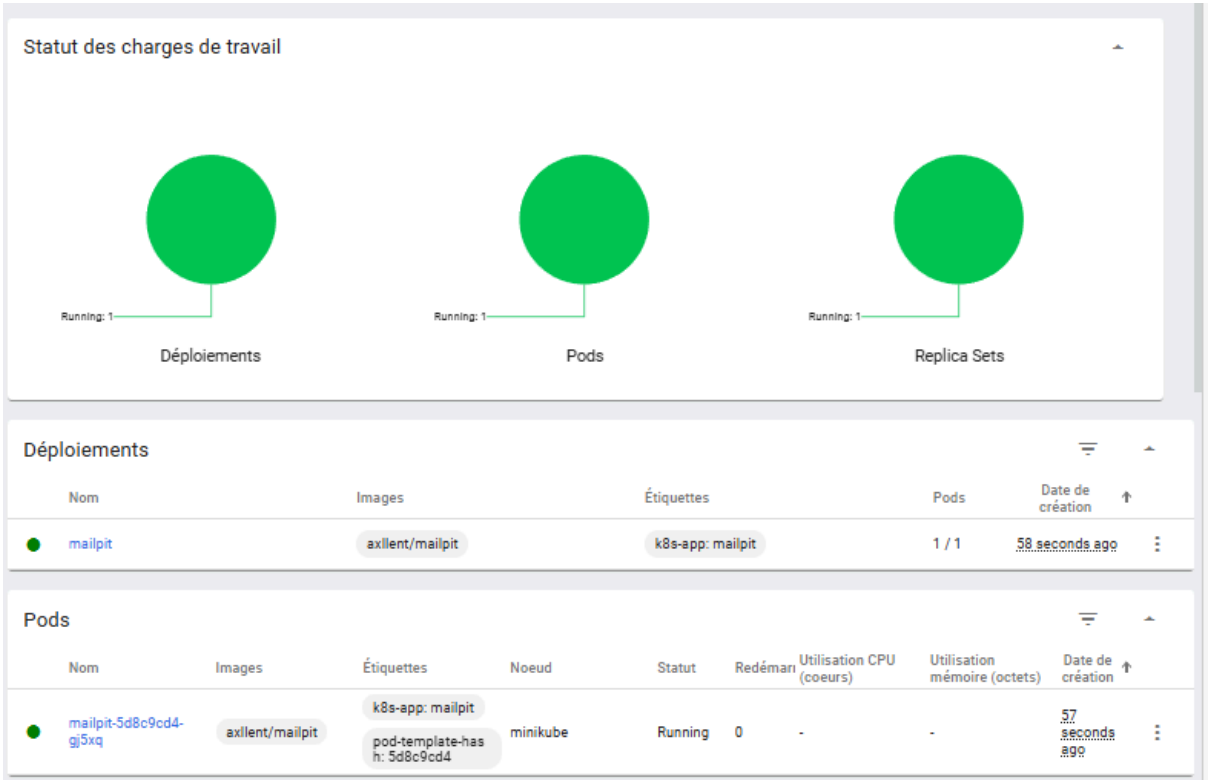

kubernetes
default
Recherche
Cluster > Espaces de noms
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
Cluster
Cluster Role Bindings
Cluster Roles
Events
Namespaces
Espaces de noms
Nom
Étiquettes
Phase
kubernetes-dashboard
addonmanager.kubernetes.io/mode: Reconcile
kubernetes.io/metadata.name: kubernetes-dashboard
kubernetes.io/minikube-addons: dashboard
Active
default
kubernetes.io/metadata.name: default
Active
kube-node-lease
kubernetes.io/metadata.name: kube-node-lease
Active
kube-public
kubernetes.io/metadata.name: kube-public
Active
kube-system
kubernetes.io/metadata.name: kube-system
Active

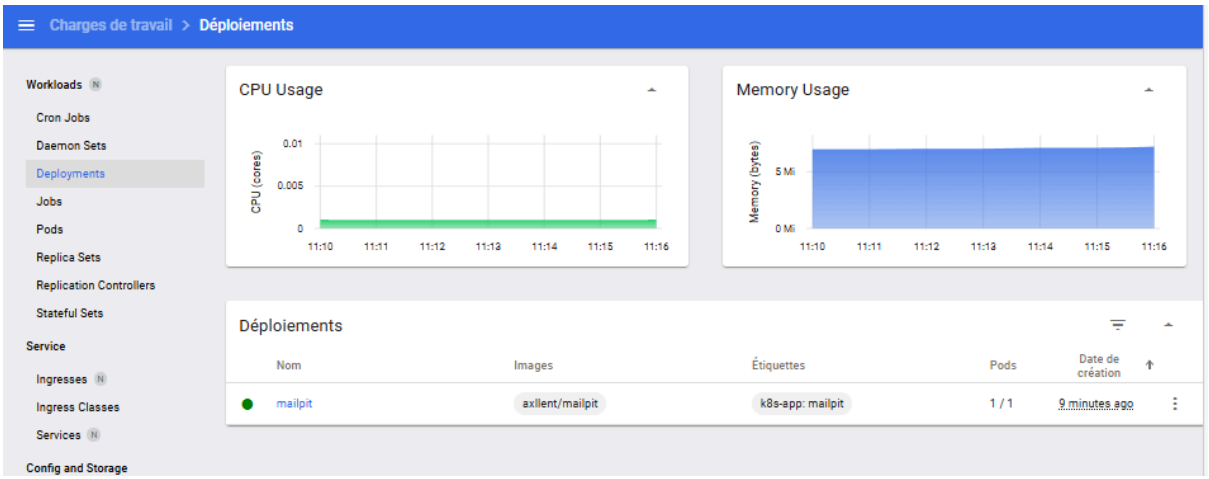

kubernetes
default
Recherche
Cluster > Noeuds
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Persistent Volume Claims
CPU Usage
CPU (cores)
Memory Usage
Memory (bytes)
2 Gi
0 Gi
Noeuds
Nom
Étiquettes
Prêt
Requêtes CPU (coeurs)
Limites CPU (coeurs)
CPU capacity (cores)
Requêtes mémoire (octets)
Limites mémoire (octets)
Memory capacity (bytes)
minikube
beta.kubernetes.io/arch: amd64
beta.kubernetes.io/os: linux
kubernetes.io/arch: amd64
Voir plus
True
850,00m (42,50%)
0,00m (0,00%)
2,00
370,00Mi (6,25%)
170,00Mi (2,87%)
5,78Gi

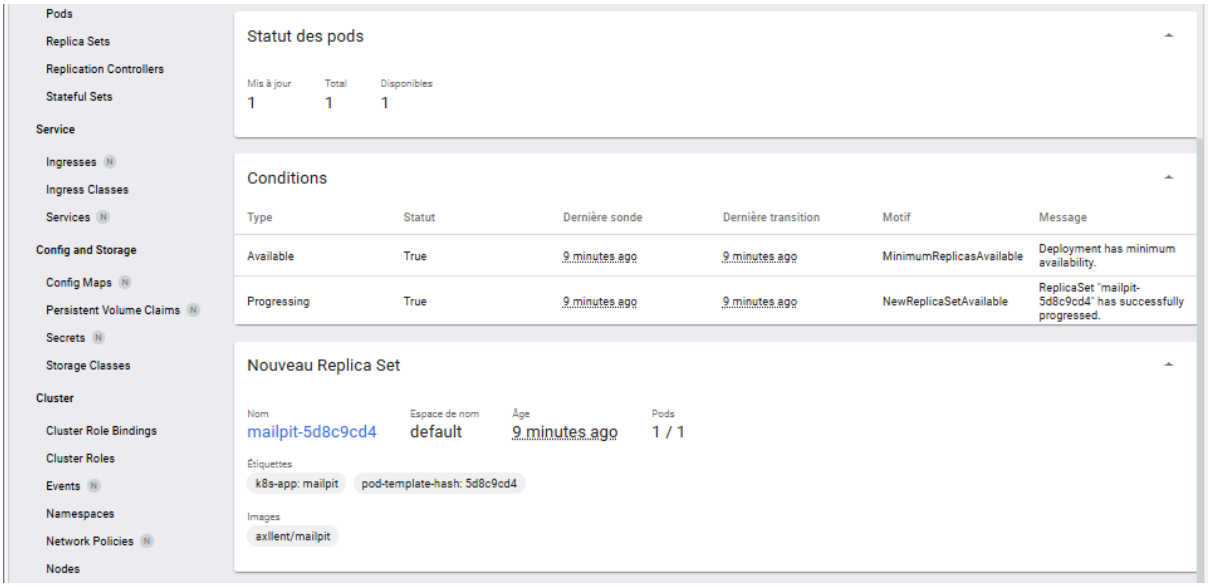
▪ Création d'un déploiement (mailpit) :

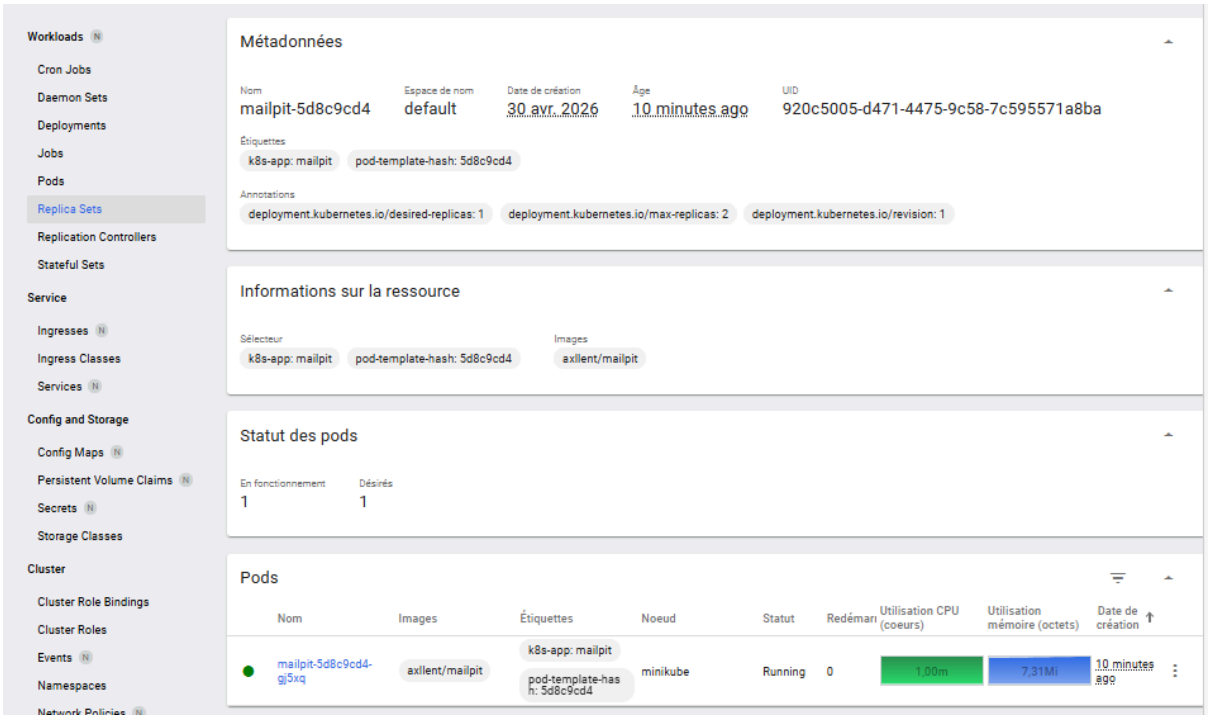




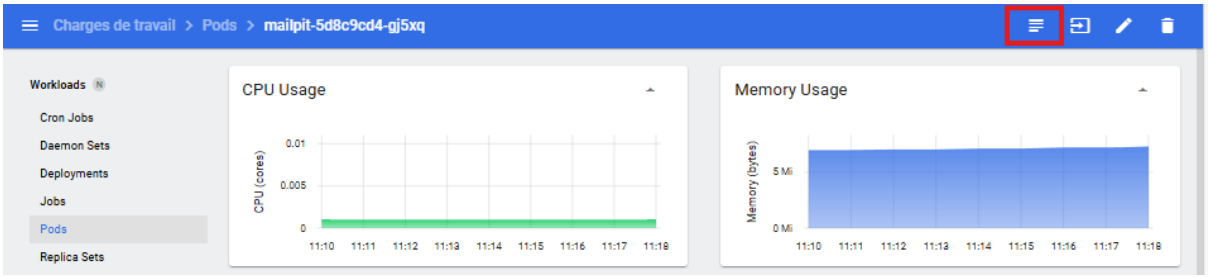

Statut des charges de travail
Running: 1
Déploiements
Running: 1
Pods
Running: 1
Replica Sets
Déploiements
Nom
Images
Étiquettes
Pods
Date de création
mailpit
axllent/mailpit
k8s-app: mailpit
1 / 1
58 seconds ago
Pods
Nom
Images
Étiquettes
Noeud
Statut
Redémarr
Utilisation CPU (coeurs)
Utilisation mémoire (octets)
Date de création
mailpit-5d8c9cd4-gj5xq
axllent/mailpit
k8s-app: mailpit
pod-template-hash: 5d8c9cd4
minikube
Running
0
-
-
57 seconds ago

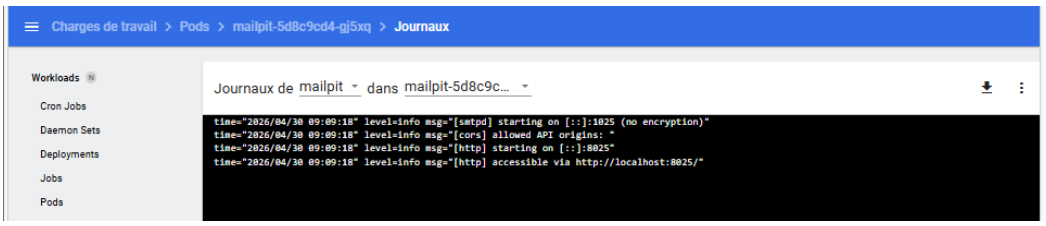

Charges de travail > Déploiements
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
CPU Usage
CPU (cores)
0.01
0.005
0
11:10
11:11
11:12
11:13
11:14
11:15
11:16
Memory Usage
Memory (bytes)
5 Mi
0 Mi
11:10
11:11
11:12
11:13
11:14
11:15
11:16
Déploiements
Nom
Images
Étiquettes
Pods
Date de création
mailpit
axllent/mailpit
k8s-app: mailpit
1 / 1
9 minutes ago



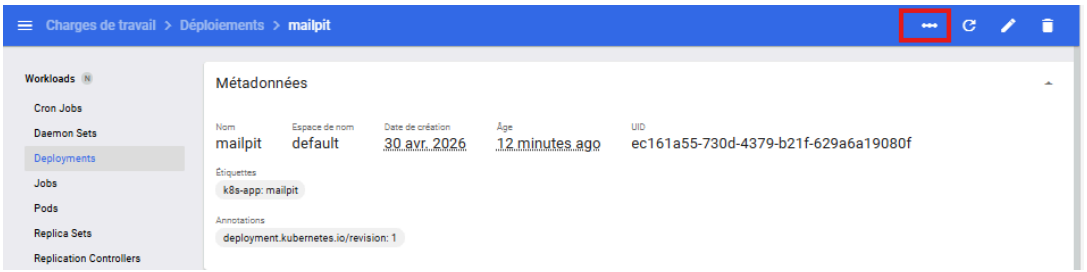


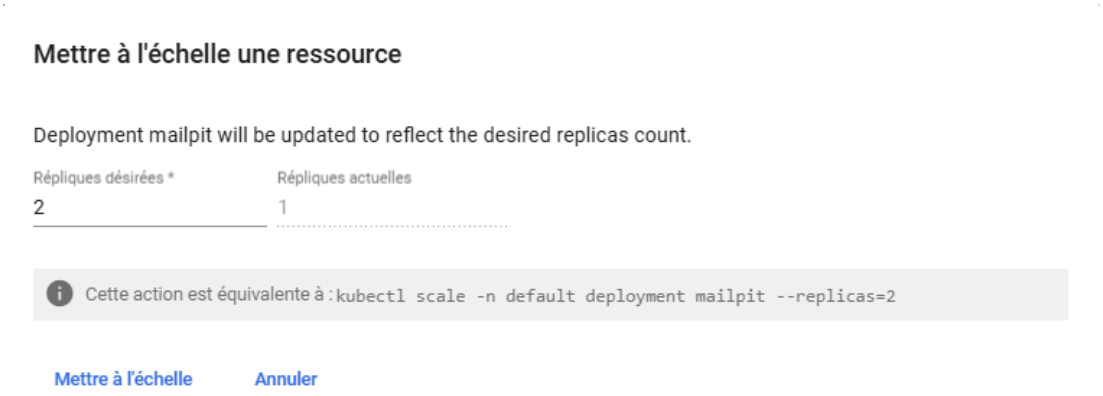
Voir les journaux :

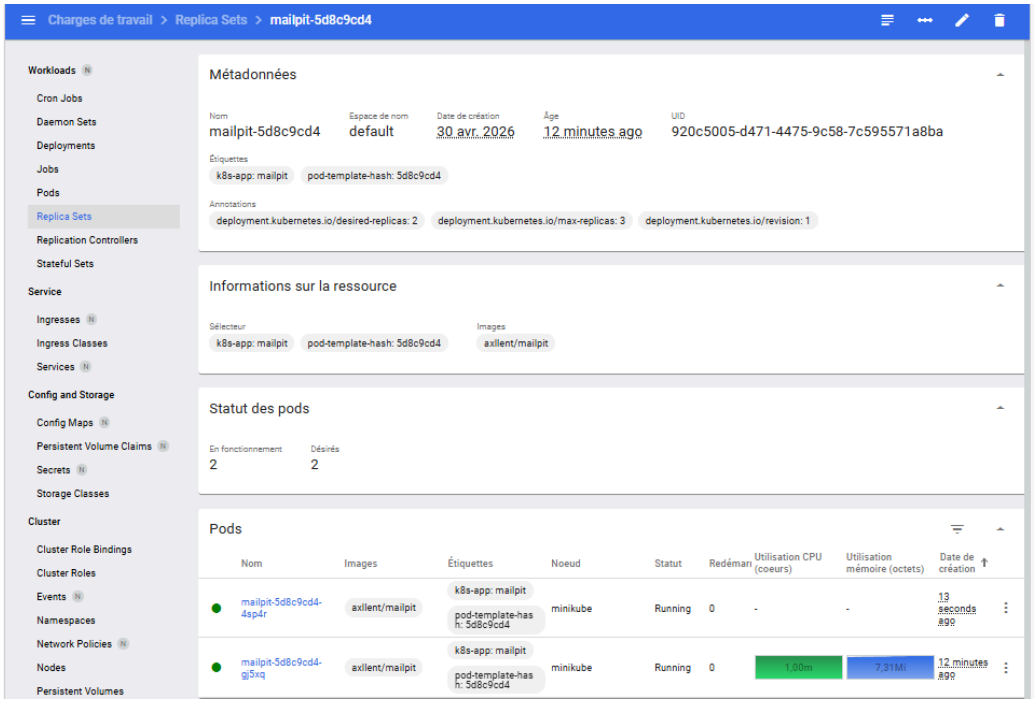




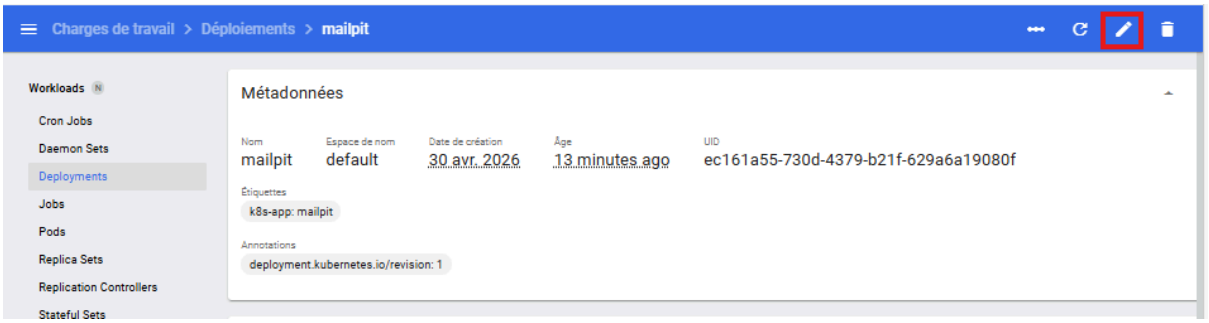
- Scalabilité :







Mise à jour :

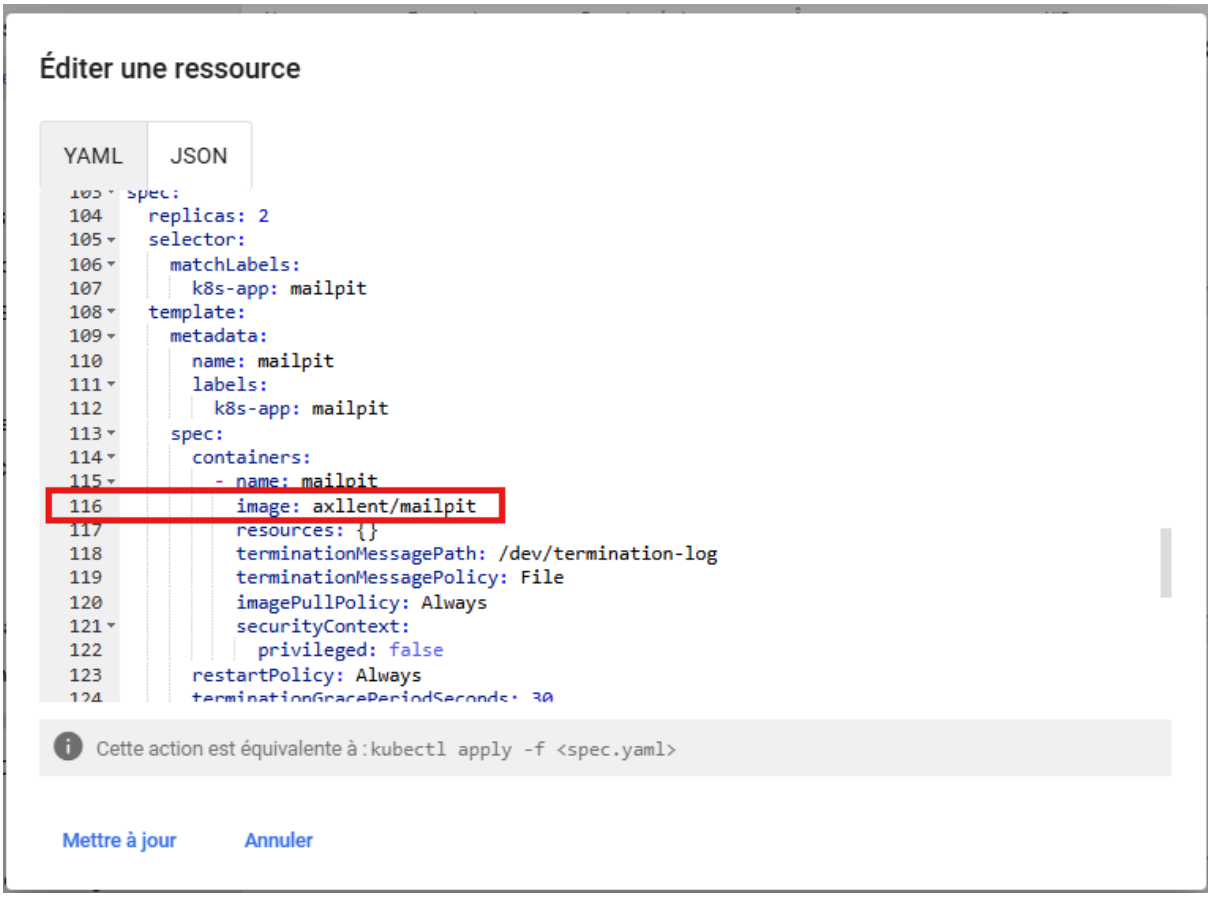


Éditer une ressource

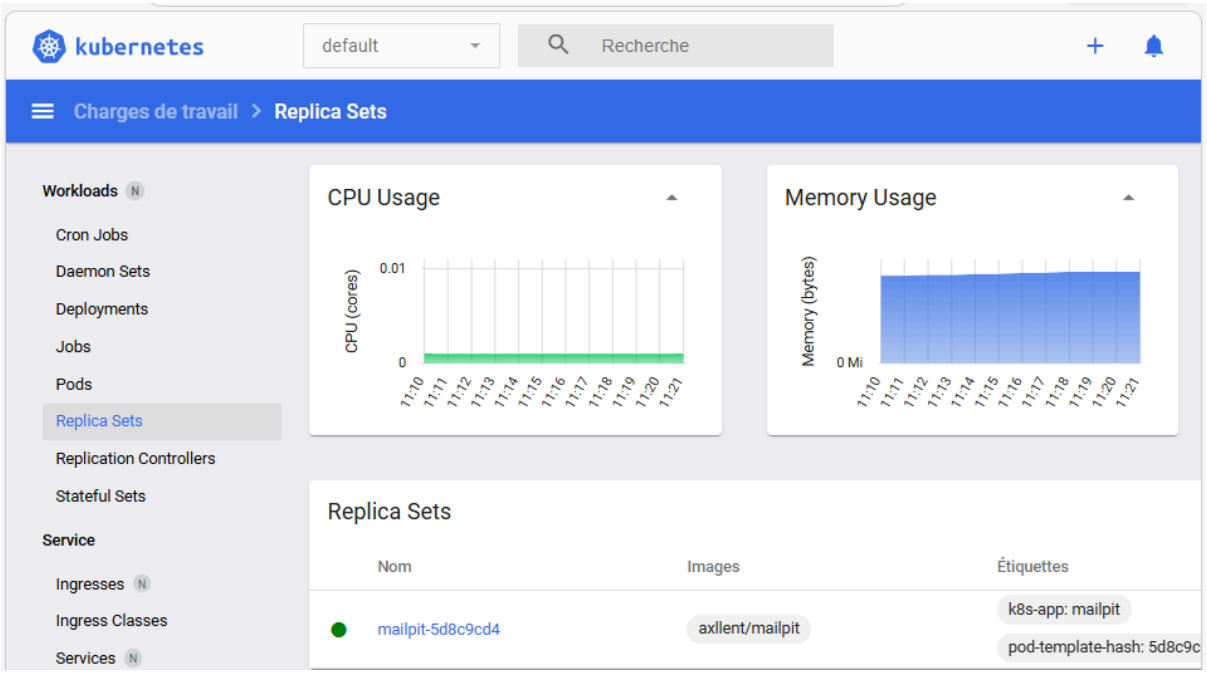
YAML JSON

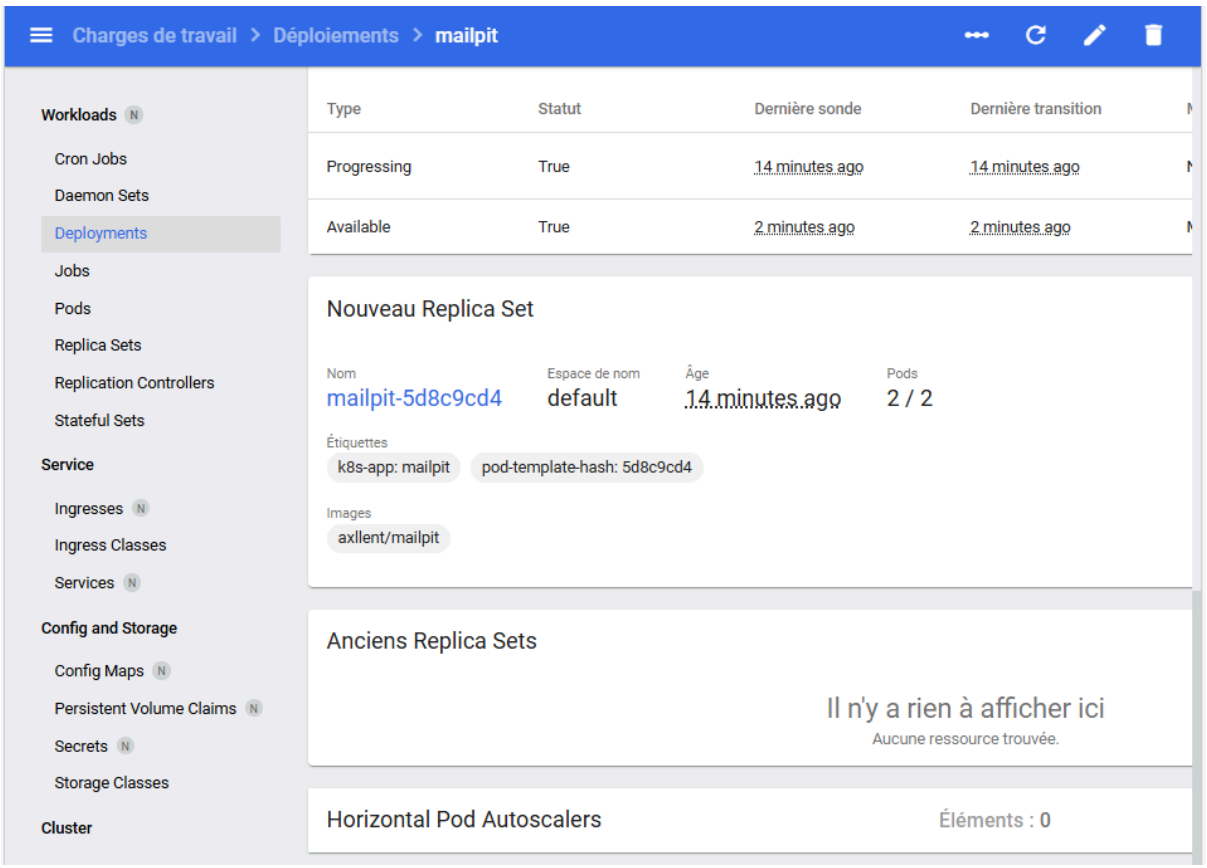
```yaml
kind: Deployment
apiVersion: apps/v1
metadata:
  name: mailpit
  namespace: default
  uid: ec161a55-730d-4379-b21f-629a6a19080f
  resourceVersion: '1987'
  generation: 2
  creationTimestamp: '2026-04-30T09:09:14Z'
  labels:
    k8s-app: mailpit
  annotations:
    deployment.kubernetes.io/revision: '1'
  managedFields:
    - manager: dashboard
      operation: Update
      apiVersion: apps/v1
      fieldsType: FieldsV1
      fieldsV1:
        f:spec:
          f:replicas: {}
```

Cette action est équivalente à : `kubectl apply -f <spec.yaml>`



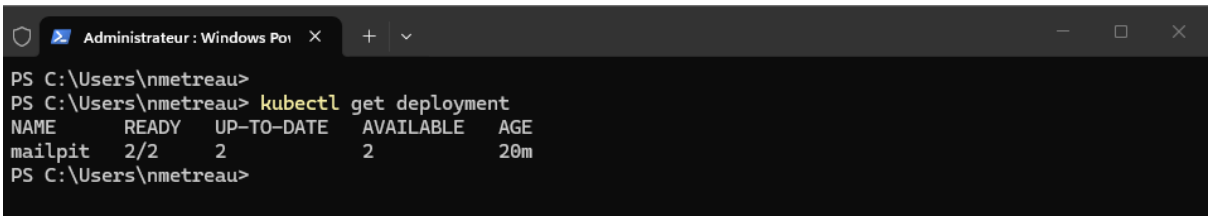
- Kubernetes va créer un nouveau ReplicaSet :

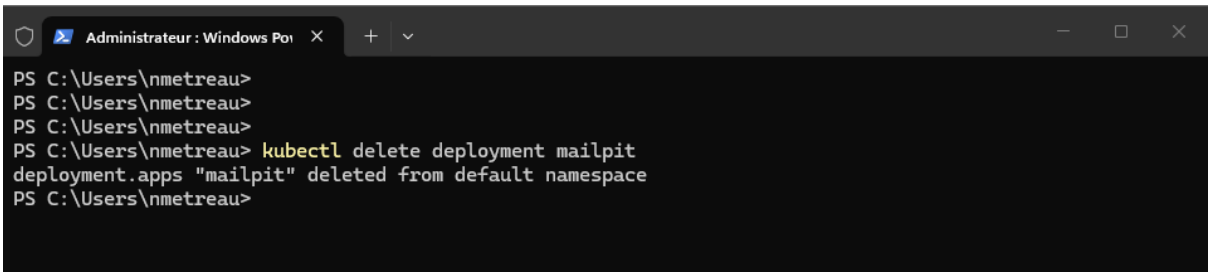


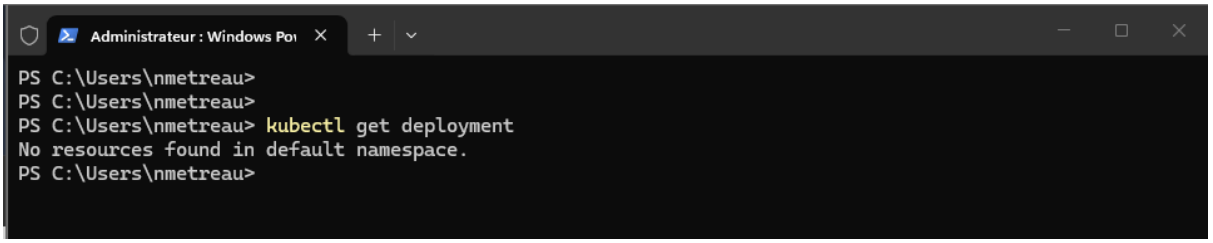

Charges de travail > Déploiements > mailpit
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
Cluster
Type
Statut
Dernière sonde
Dernière transition
Progressing
True
14 minutes ago
14 minutes ago
Available
True
2 minutes ago
2 minutes ago
Nouveau Replica Set
Nom
mailpit-5d8c9cd4
Espace de nom
default
Âge
14 minutes ago
Pods
2 / 2
Étiquettes
k8s-app: mailpit
pod-template-hash: 5d8c9cd4
Images
axllent/mailpit
Anciens Replica Sets
Il n'y a rien à afficher ici
Aucune ressource trouvée.
Horizontal Pod Autoscalers
Éléments : 0

4. Créer un déploiement à l'aide de l'outil kubectl.

▪ Nous allons supprimer un déploiement :

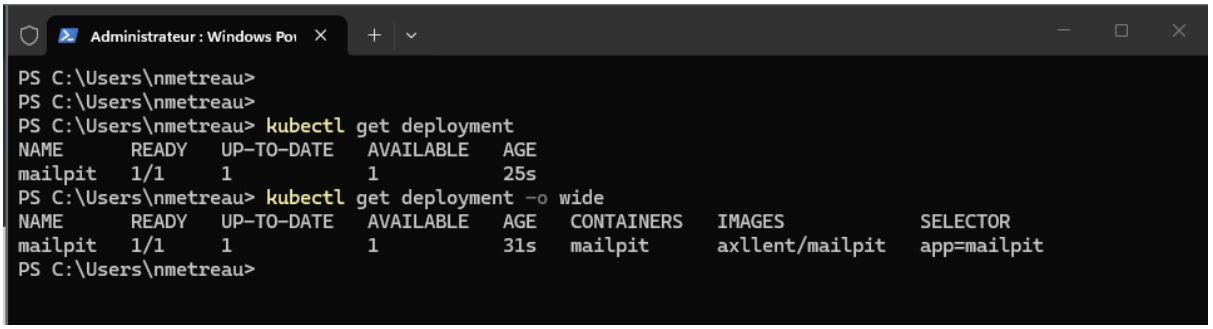


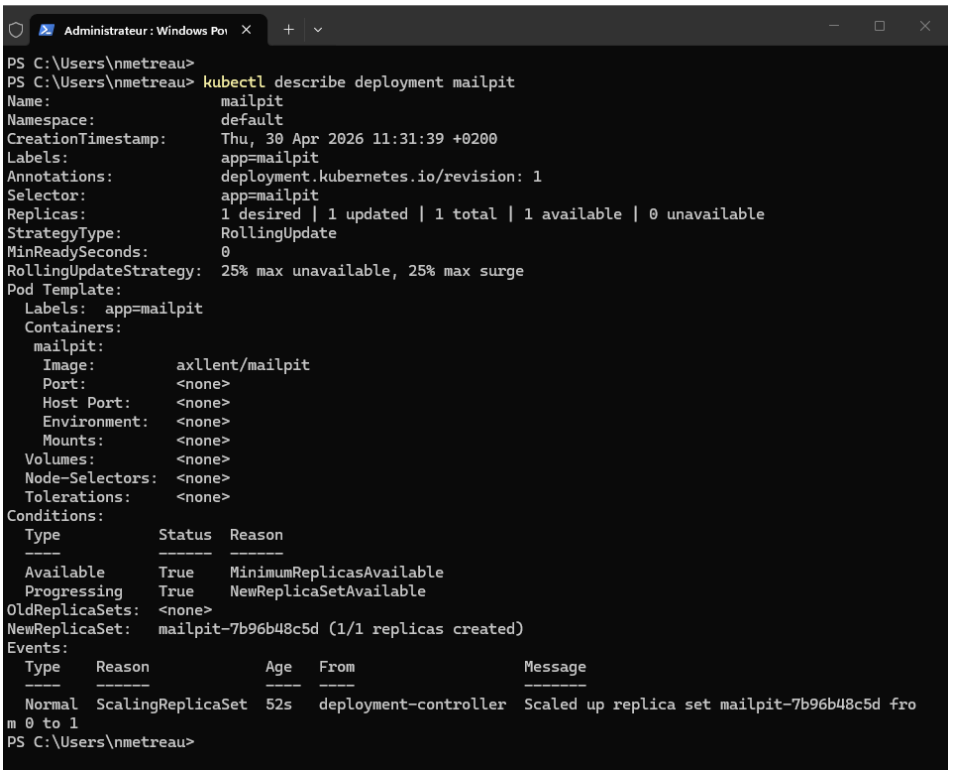




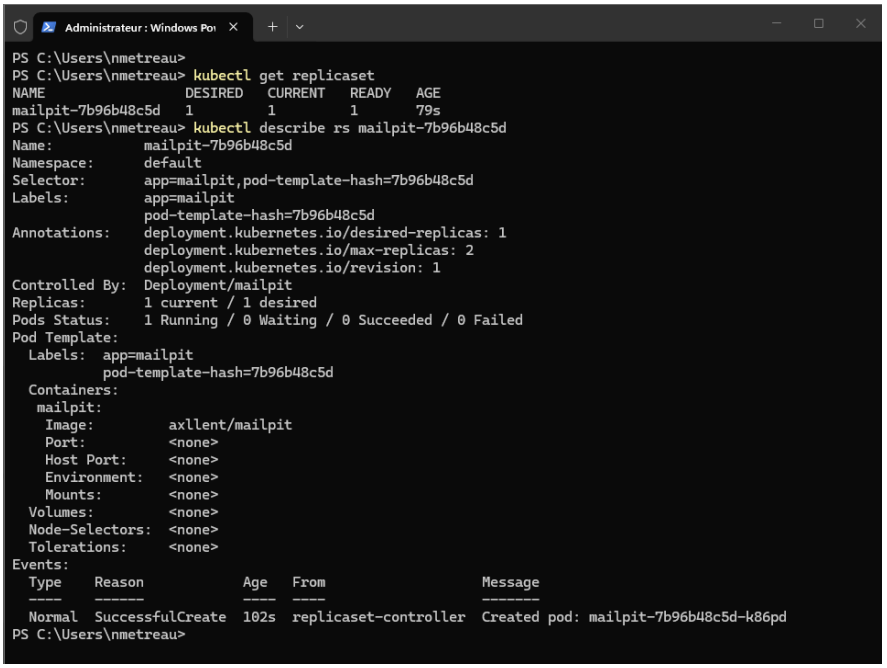
▪ Création d'un déploiement

- Etat du déploiement

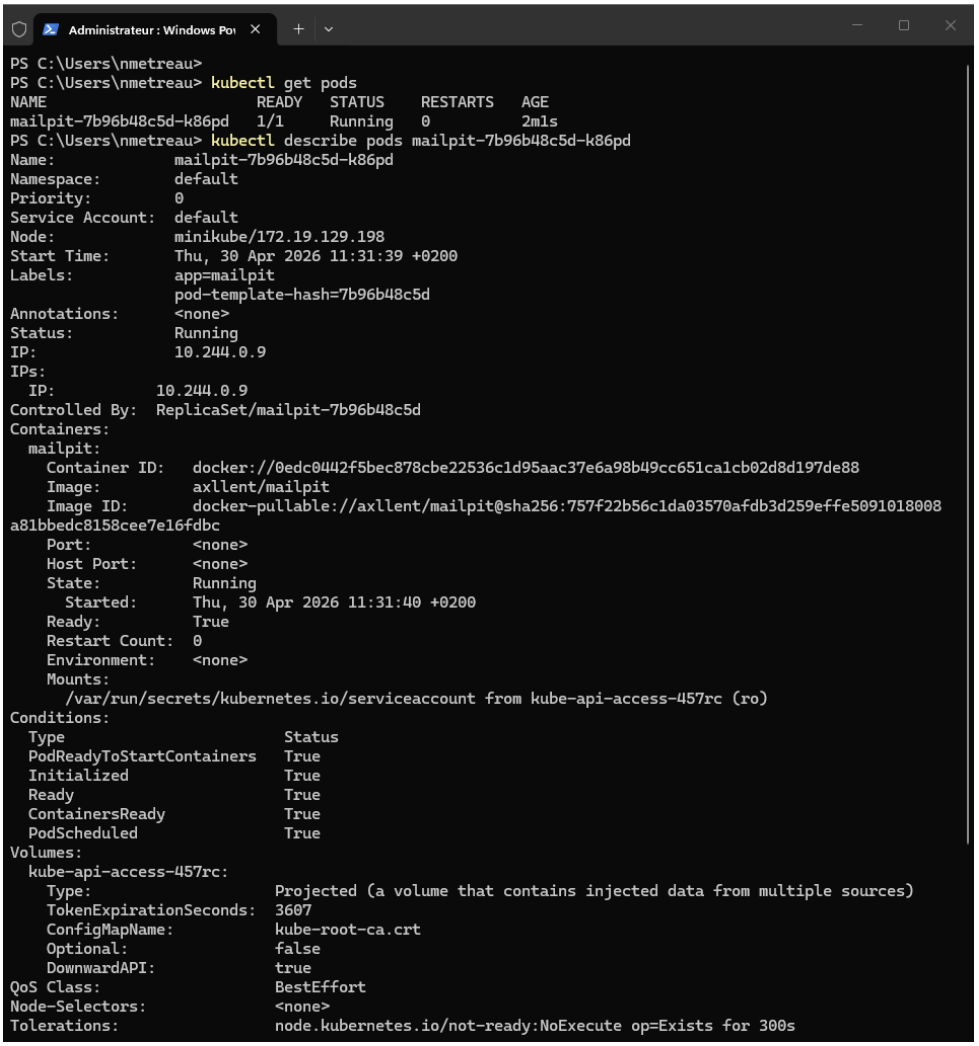




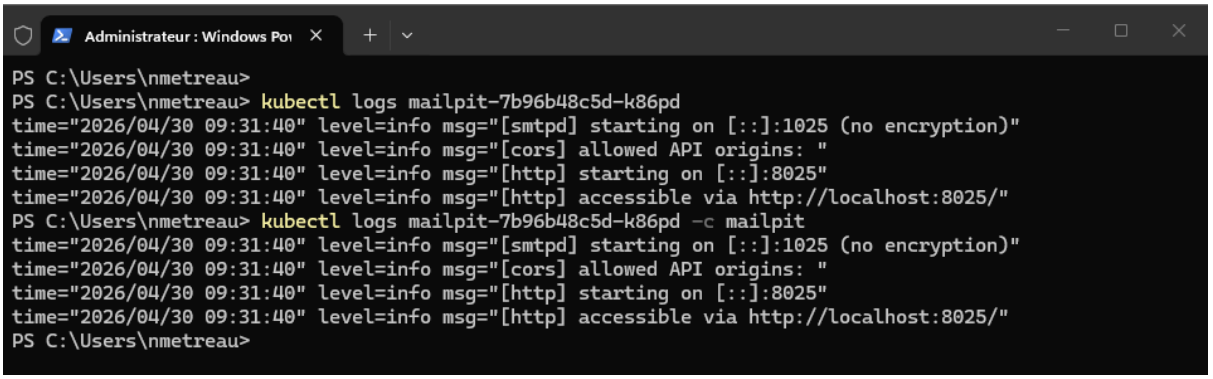
- Réplicas :



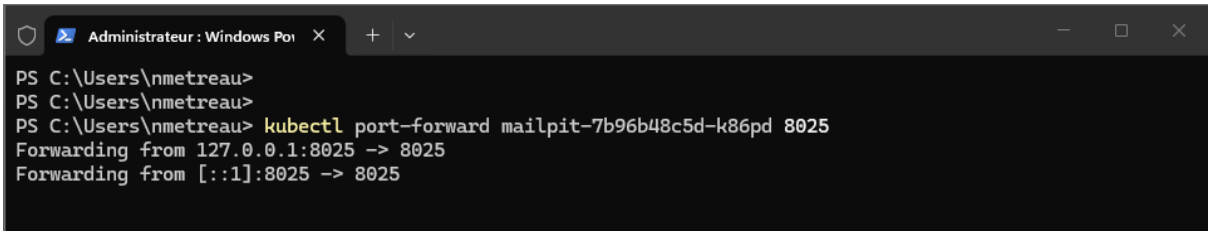
- Nous vérifions l'état du pod

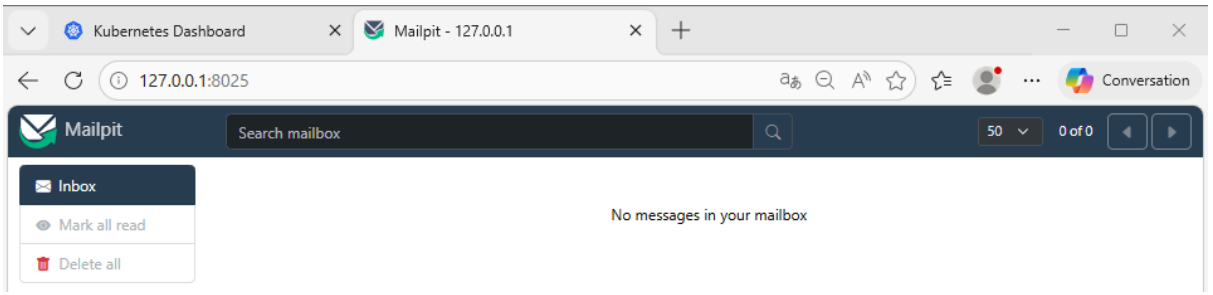


▪ Nous accédons aux logs du pod et des conteneurs

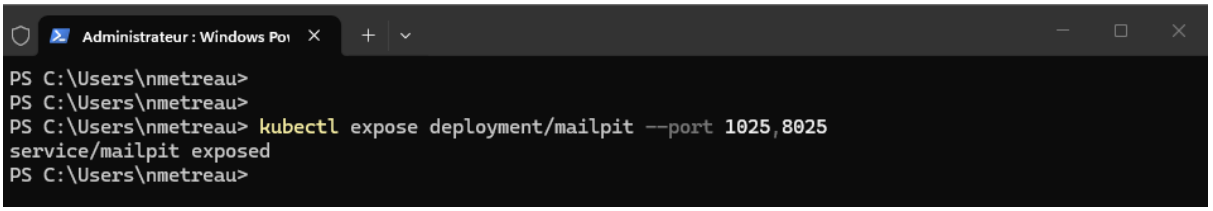


▪ Accès à l'application Mailpit



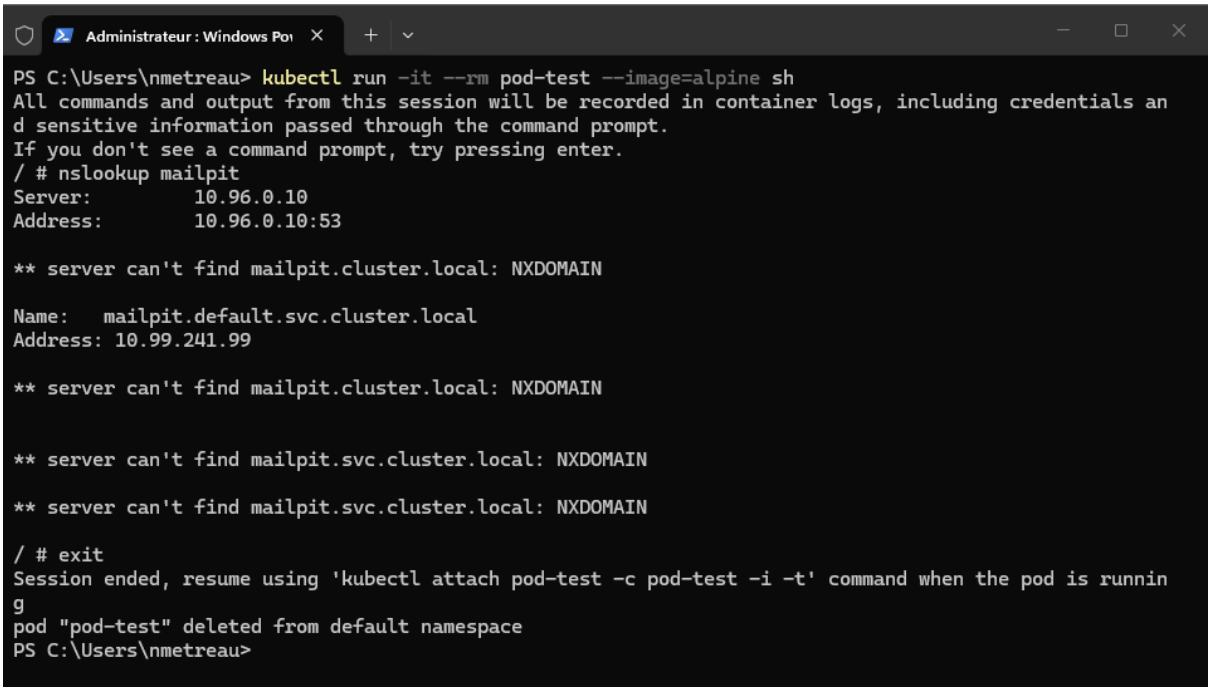


▪ Création d'un service Mailpit



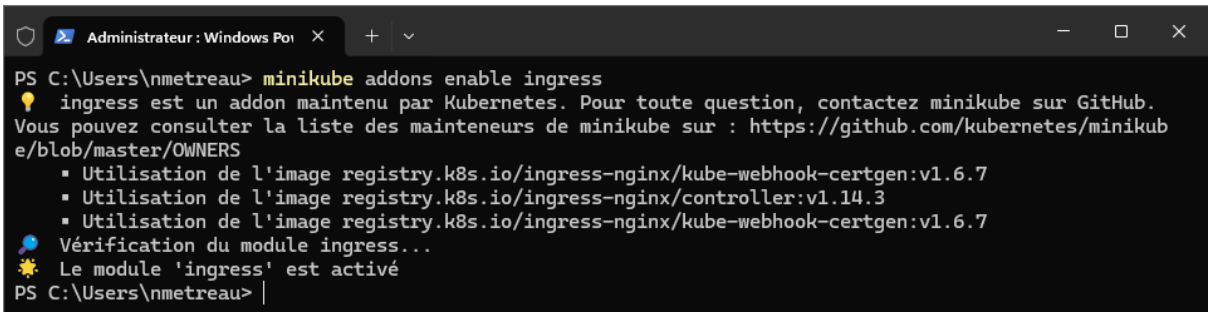
▪ Ouverture d'un terminal interactif à l'intérieur d'un pod Kubernetes :

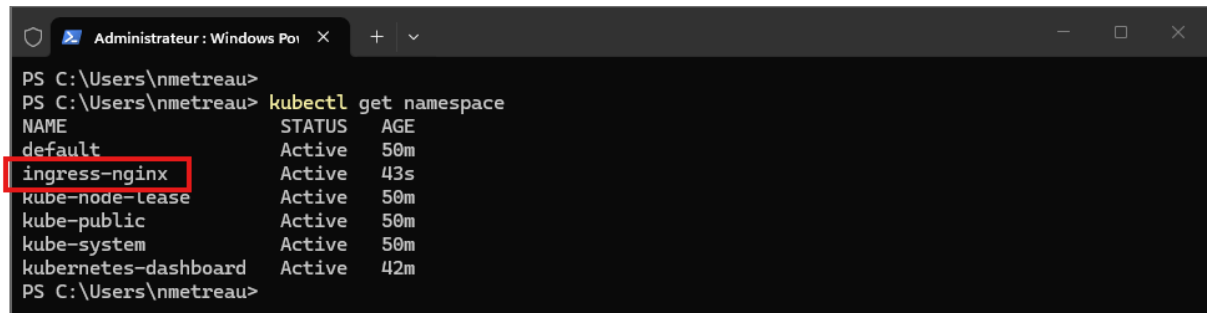
▪ Lancement d'un pod de test :

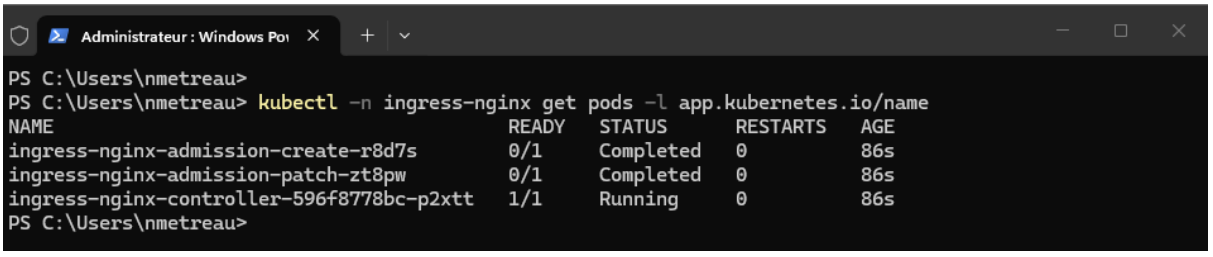


▪ Nous modifions le nombre de pods = 2 :

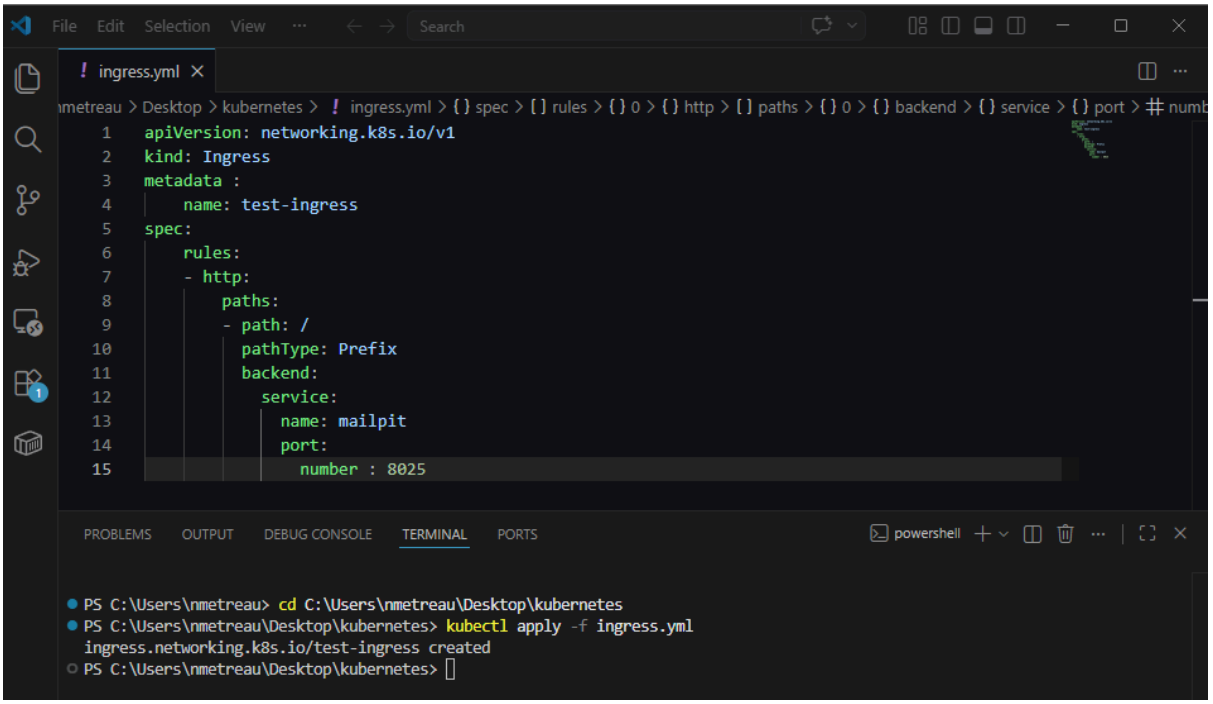
▪ Nous activons le contrôleur ingress dans minikube :

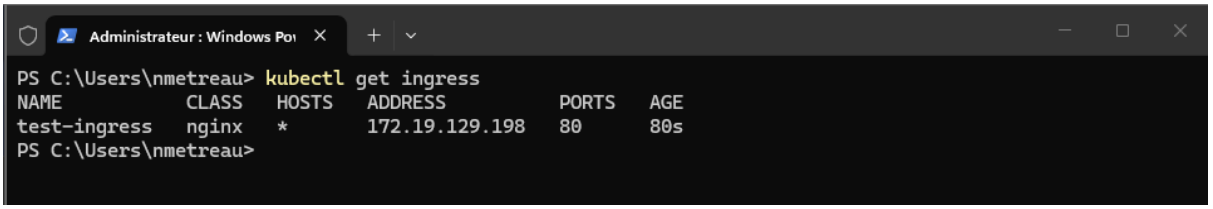


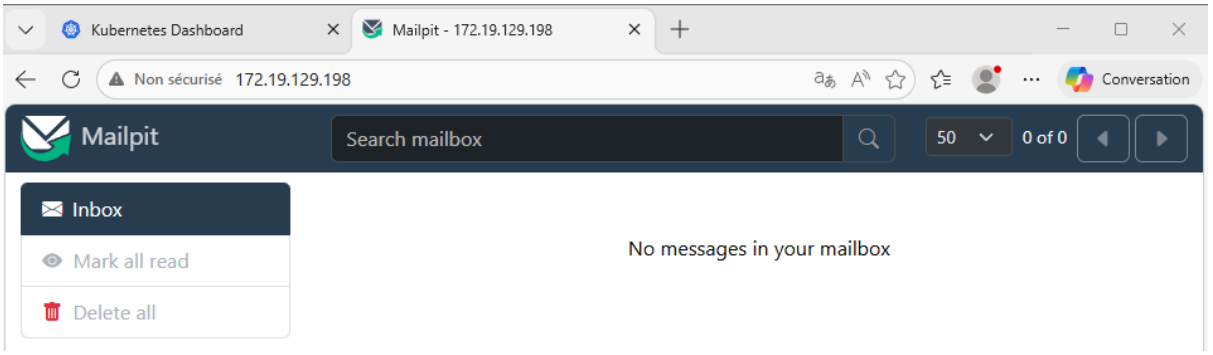




▪ Nous déclarons une règle Ingress et prise en compte du fichier YAML :

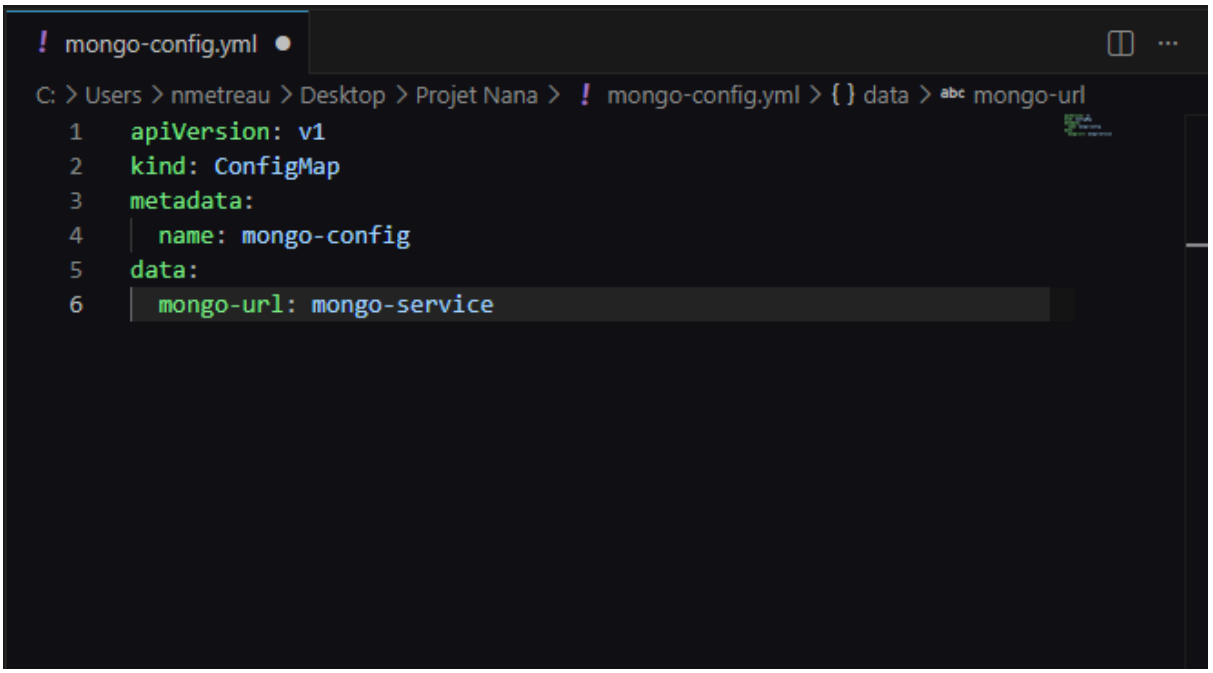




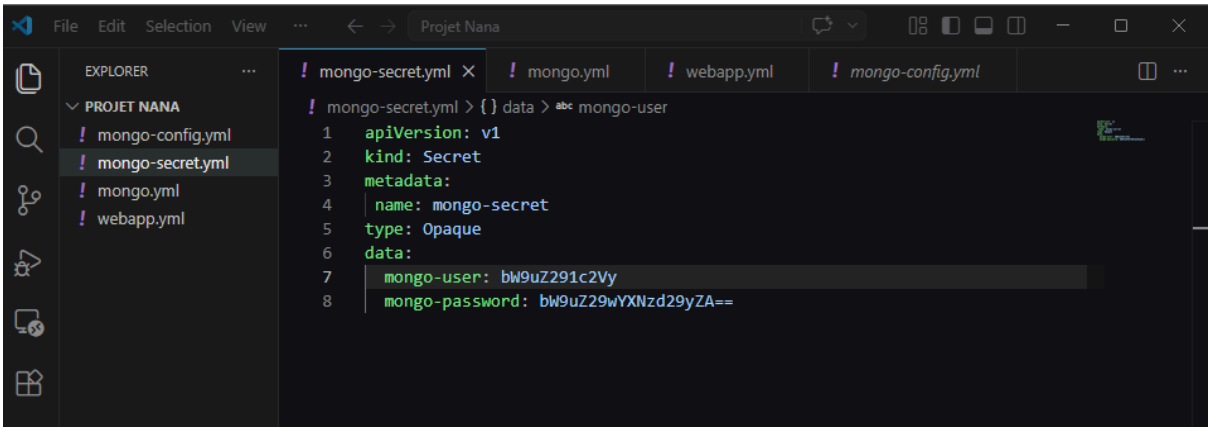


5. Automatisation de déploiement par fichier YAML.

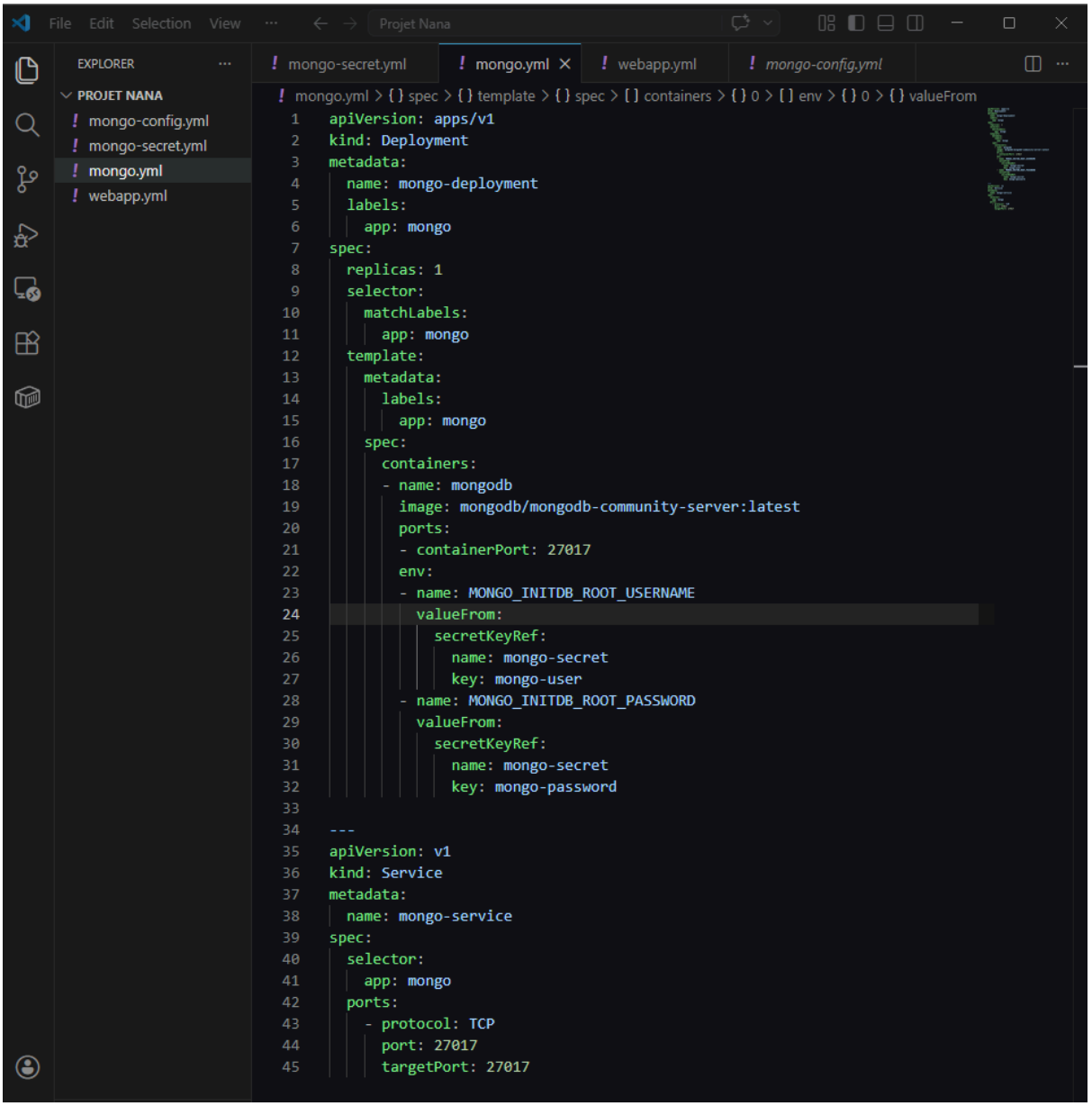
▪ Création du fichier mongo-config.yml :



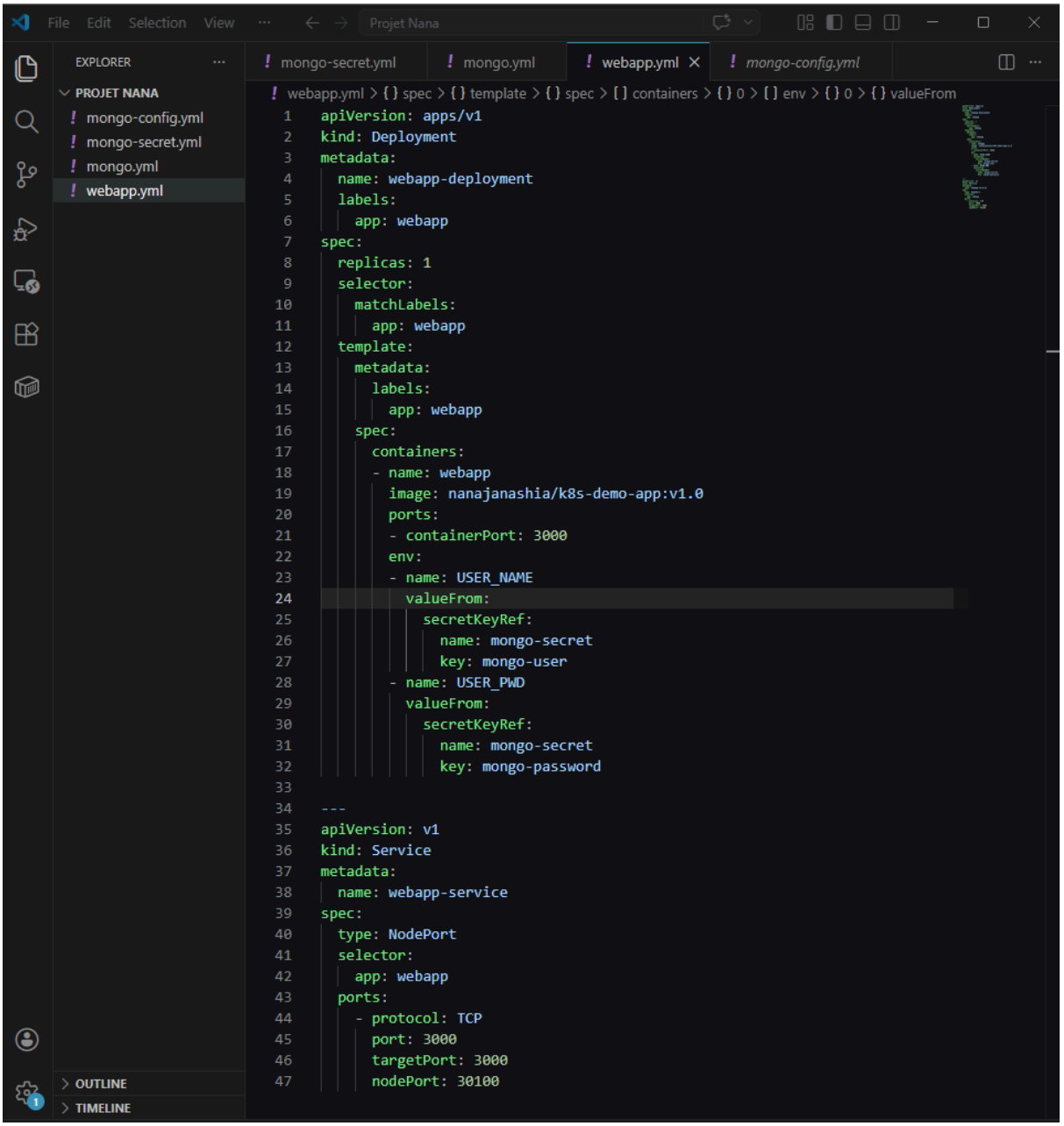
▪ Création du fichier mongo-secret.yml :



▪ Création d'un service interne :

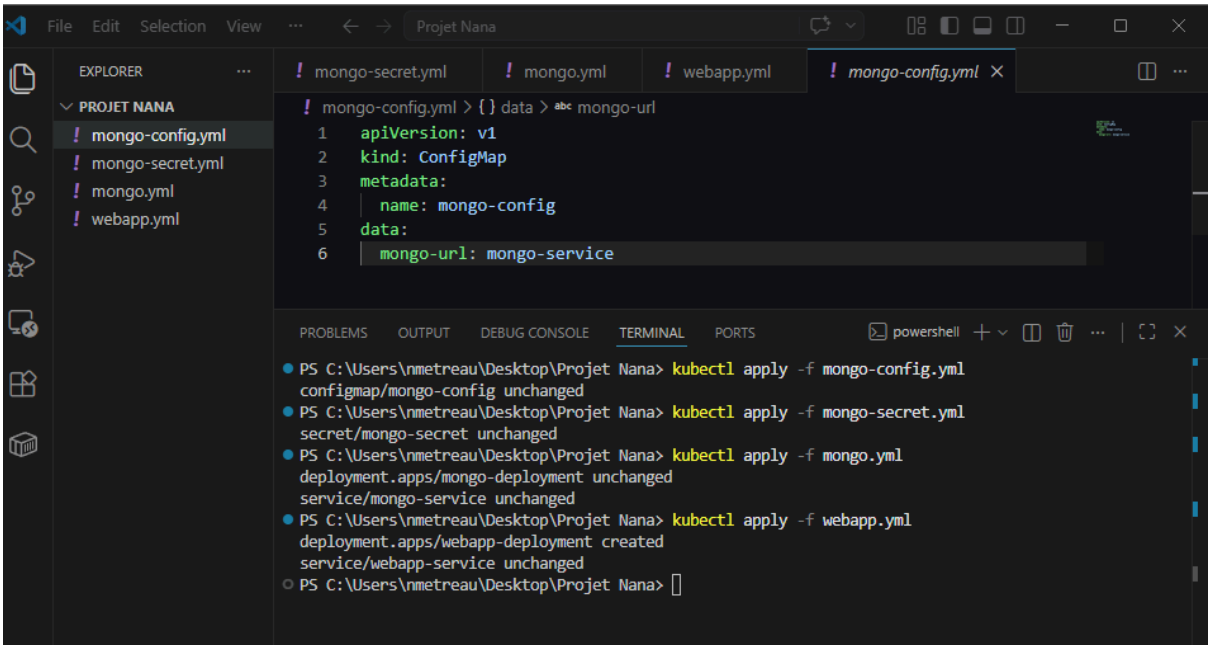


▪ Création d'un service externe

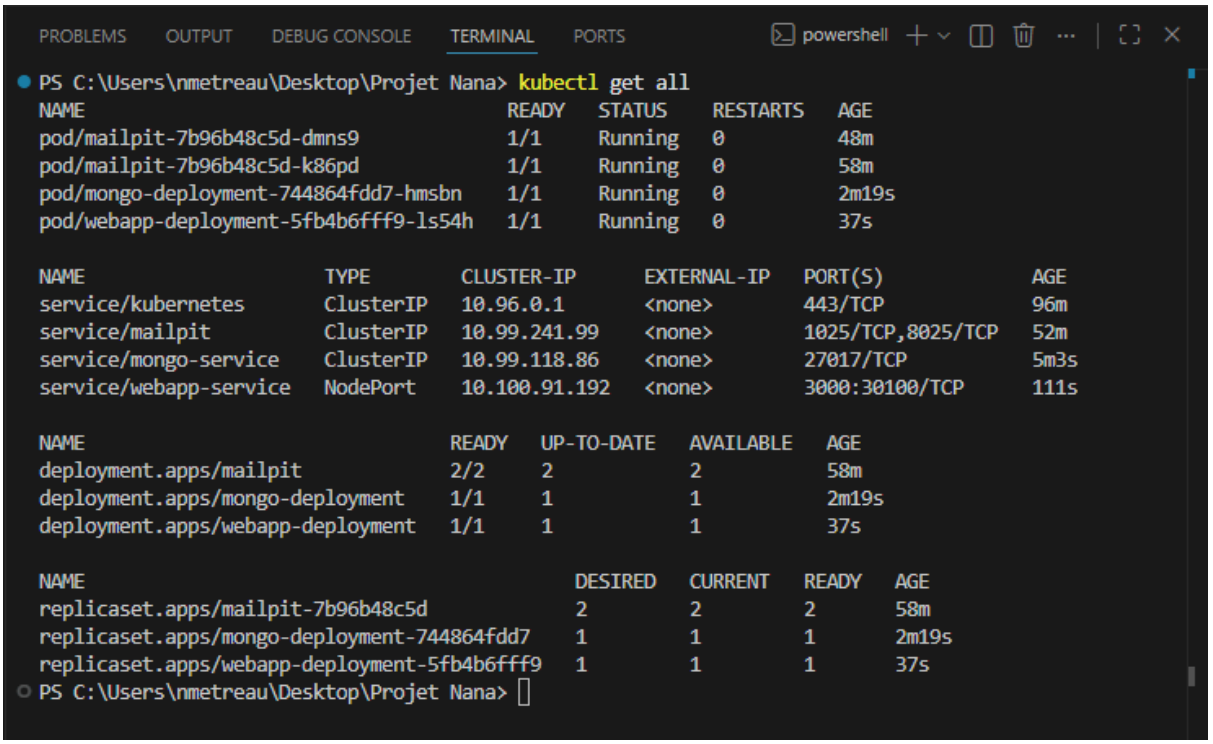



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp-deployment
  labels:
    app: webapp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: webapp
  template:
    metadata:
      labels:
        app: webapp
    spec:
      containers:
      - name: webapp
        image: nanajanashia/k8s-demo-app:v1.0
        ports:
        - containerPort: 3000
        env:
        - name: USER_NAME
          valueFrom:
            secretKeyRef:
              name: mongo-secret
              key: mongo-user
        - name: USER_PWD
          valueFrom:
            secretKeyRef:
              name: mongo-secret
              key: mongo-password

---
apiVersion: v1
kind: Service
metadata:
  name: webapp-service
spec:
  type: NodePort
  selector:
    app: webapp
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
      nodePort: 30100
```

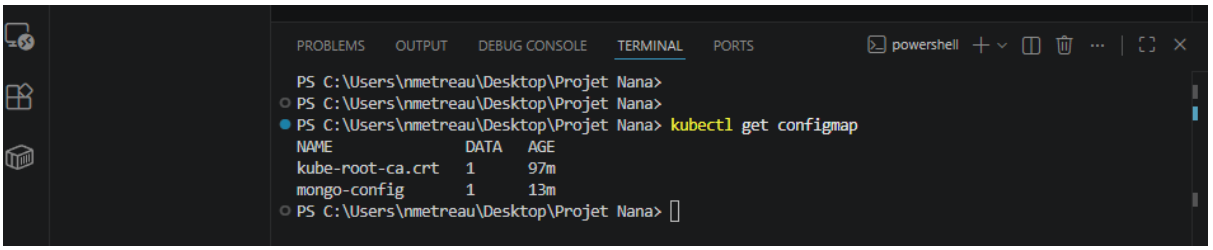
▪ Nous procédons à la création des composants en appliquant la commande kubectl apply -f aux 4 fichiers YAML dans l'ordre figurant ci-dessous :



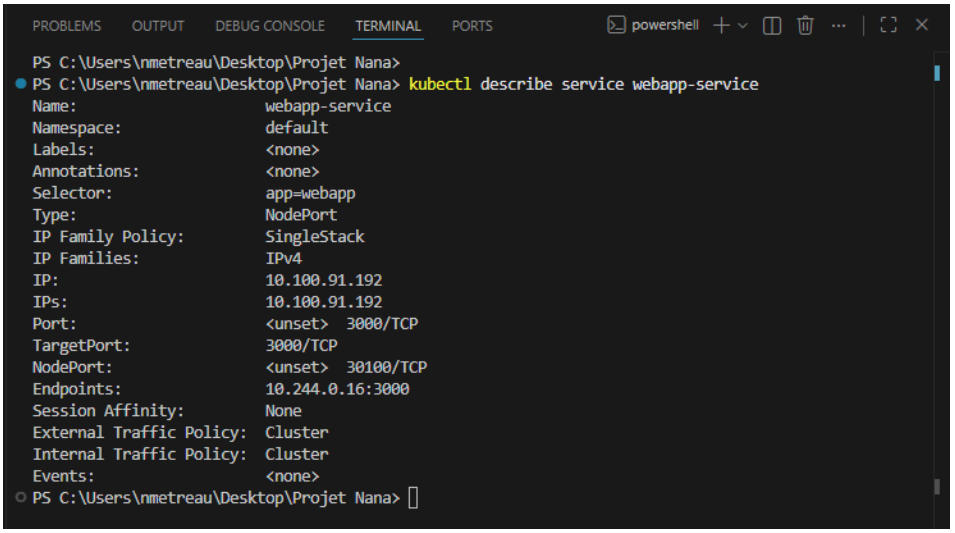
▪ Nous listons les objets créés :



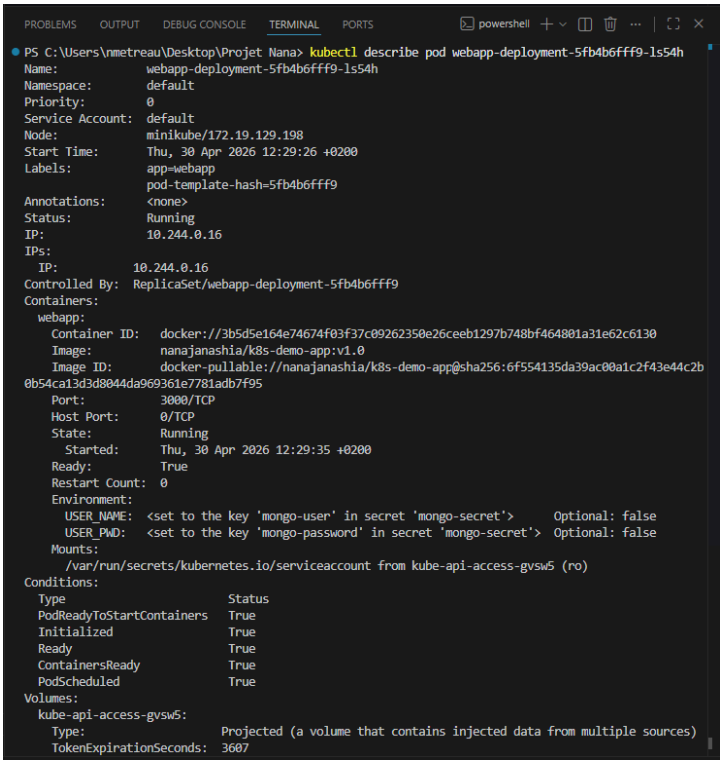
▪ Nous listons les autres objets :



▪ Nous décrivons le service webapp-service :



▪ Nous décrivons le pod associé :



▪ Nous consultons les logs de ce pod :

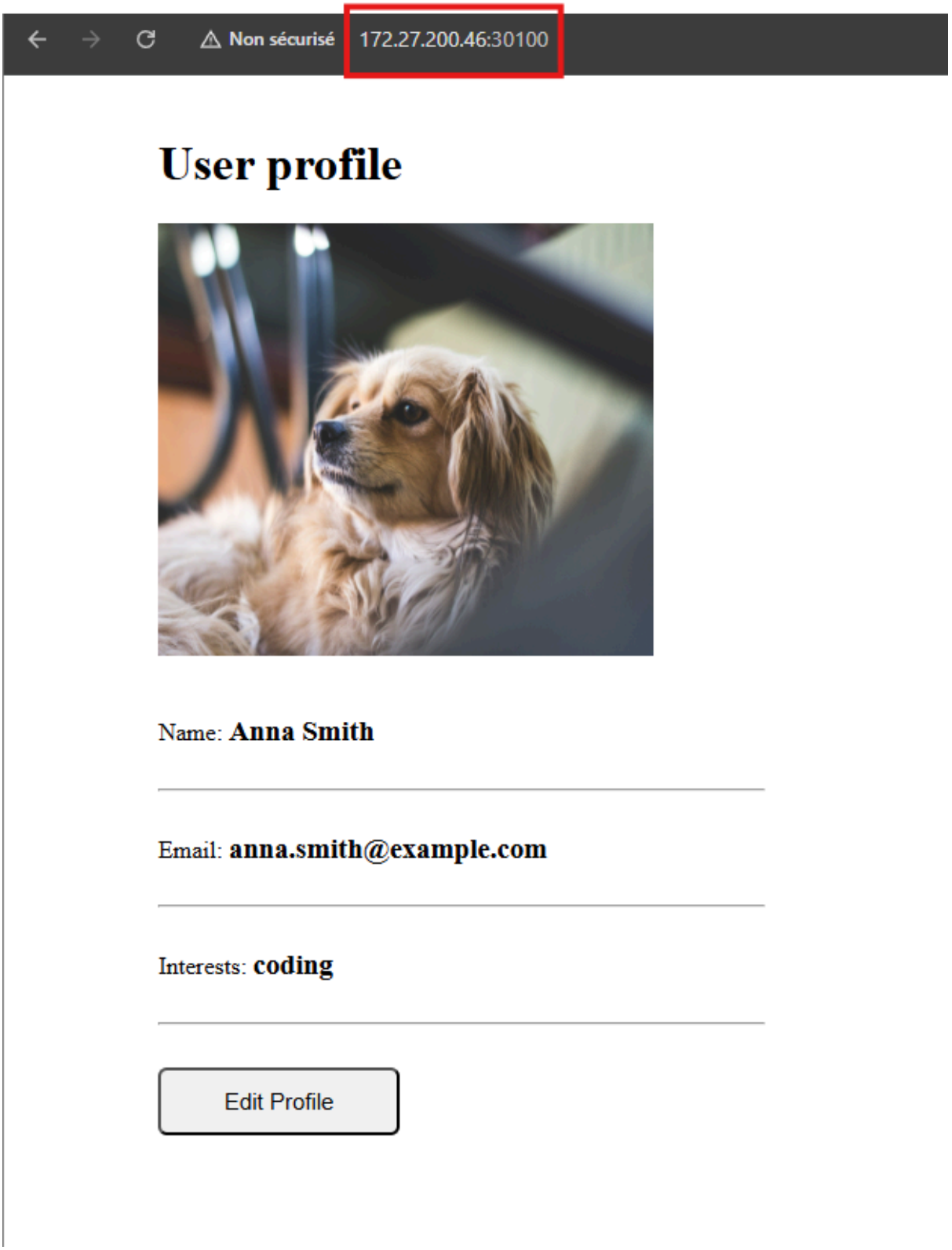
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                  powershell
PS C:\Users\nmetreau\Desktop\Projet Nana> kubectl get pod
NAME                                 READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-dmns9             1/1     Running   0          60m
mailpit-7b96b48c5d-k86pd             1/1     Running   0          70m
mongo-deployment-744864fdd7-hmsbn    1/1     Running   0          14m
webapp-deployment-5fb4b6fff9-ls54h   1/1     Running   0          12m
PS C:\Users\nmetreau\Desktop\Projet Nana>
```

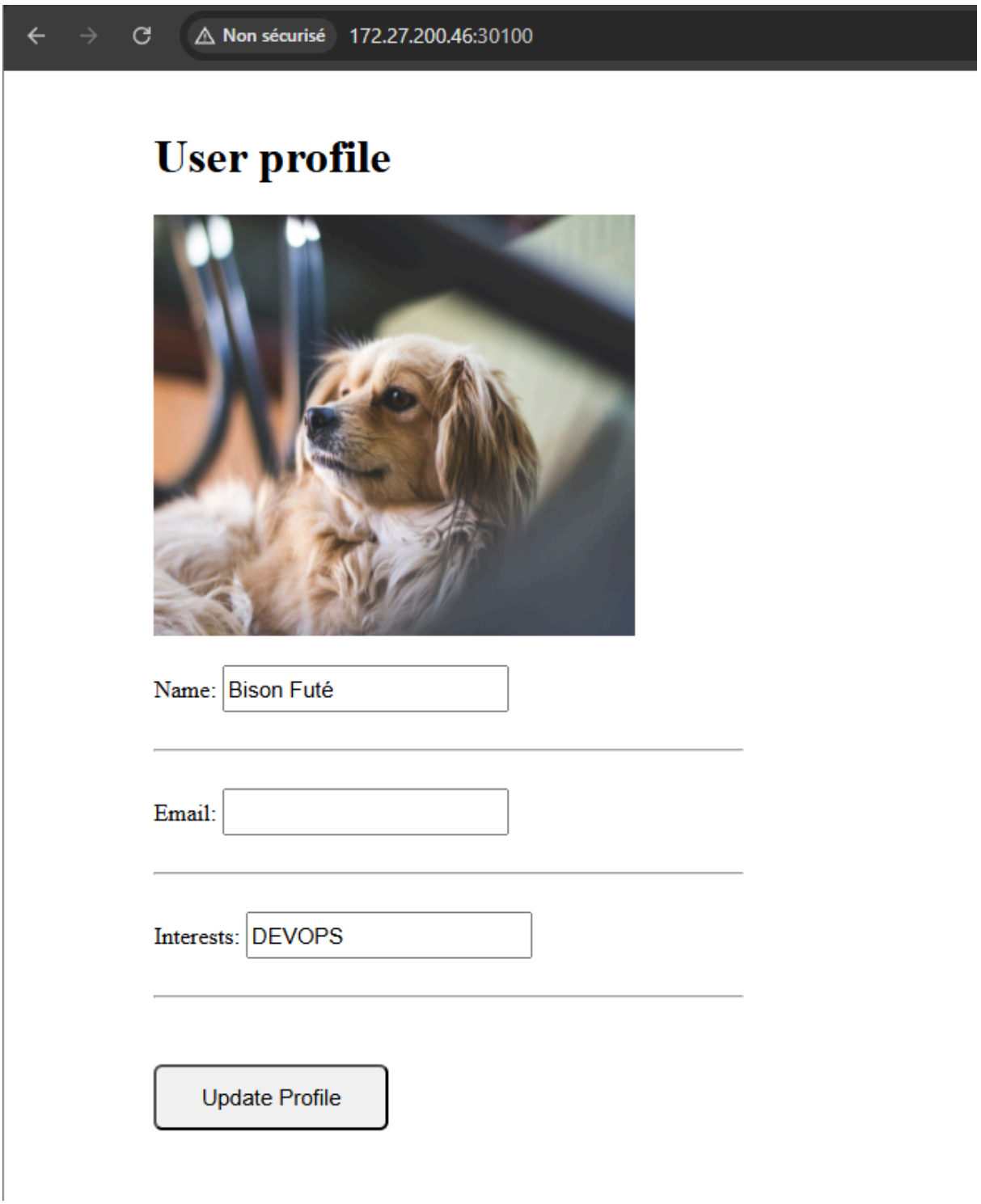
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                  powershell
PS C:\Users\nmetreau\Desktop\Projet Nana> kubectl get pod
NAME                                 READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-dmns9             1/1     Running   0          60m
mailpit-7b96b48c5d-k86pd             1/1     Running   0          70m
mongo-deployment-744864fdd7-hmsbn    1/1     Running   0          14m
webapp-deployment-5fb4b6fff9-ls54h   1/1     Running   0          12m
PS C:\Users\nmetreau\Desktop\Projet Nana> kubectl webapp-deployment-5fb4b6fff9-ls54h
error: unknown command "webapp-deployment-5fb4b6fff9-ls54h" for "kubectl"
PS C:\Users\nmetreau\Desktop\Projet Nana> kubectl logs webapp-deployment-5fb4b6fff9-ls54h
app listening on port 3000!
PS C:\Users\nmetreau\Desktop\Projet Nana>
```

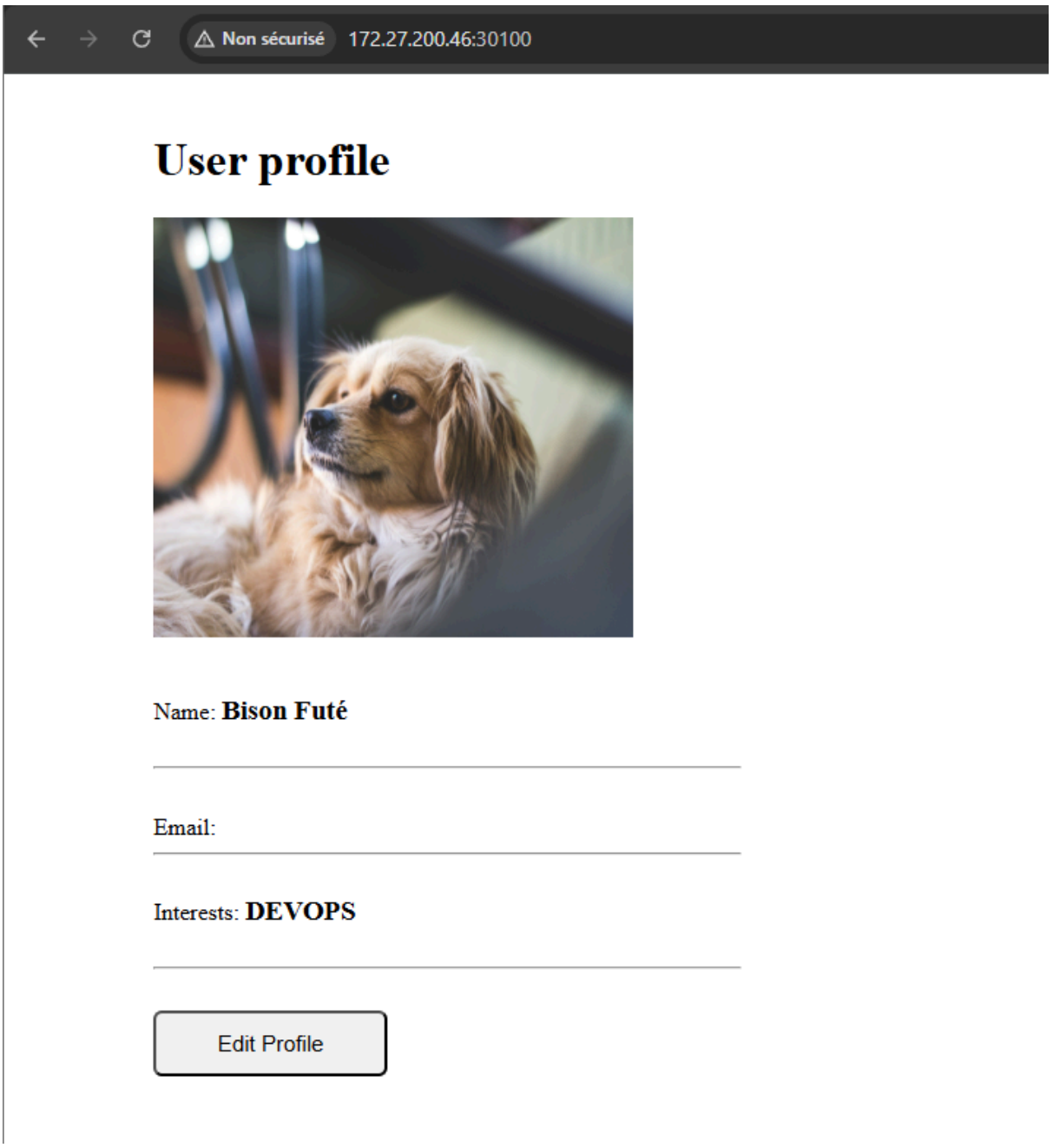
```
PS C:\Users\nmetreau\Desktop\Projet Nana> kubectl logs webapp-deployment-5fb4b6fff9-ls54h -f
app listening on port 3000!
```

▪ Nous listons les services :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                  powershell
PS C:\Users\nmetreau\Desktop\Projet Nana> kubectl get service
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)             AGE
kubernetes       ClusterIP   10.96.0.1       <none>        443/TCP             110m
mailpit          ClusterIP   10.99.241.99    <none>        1025/TCP,8025/TCP   66m
mongo-service    ClusterIP   10.99.118.86    <none>        27017/TCP           18m
webapp-service   NodePort    10.100.91.192   <none>        3000:30100/TCP      15m
PS C:\Users\nmetreau\Desktop\Projet Nana>
```

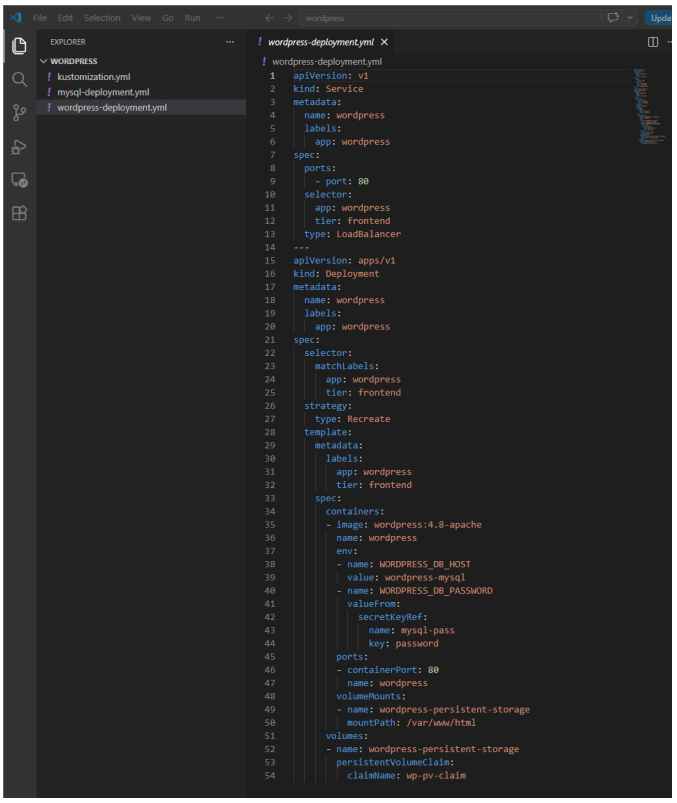

Non sécurisé
172.27.200.46:30100
User profile
Name: Anna Smith
Email: anna.smith@example.com
Interests: coding
Edit Profile

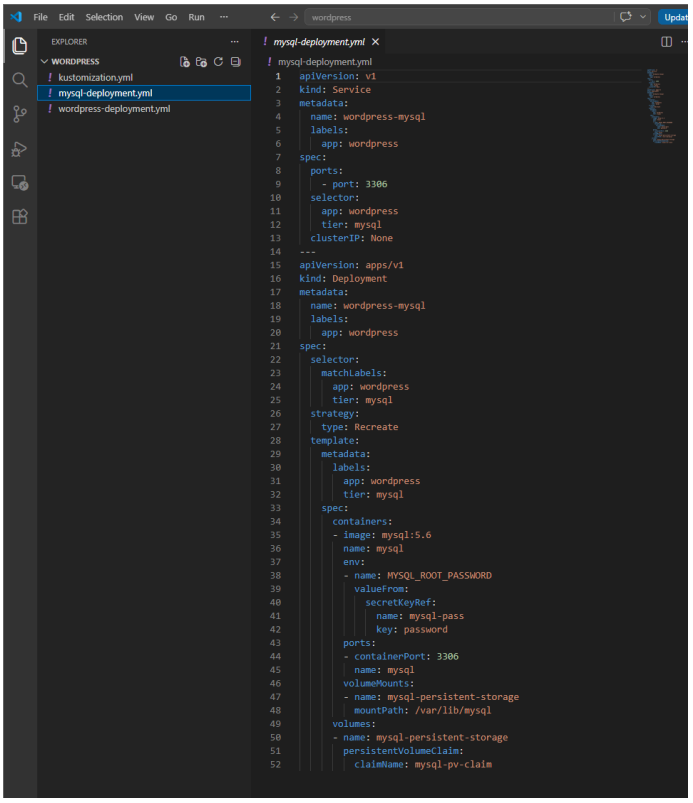

Non sécurisé
172.27.200.46:30100
User profile
Name: Bison Futé
Email:
Interests: DEVOPS
Update Profile

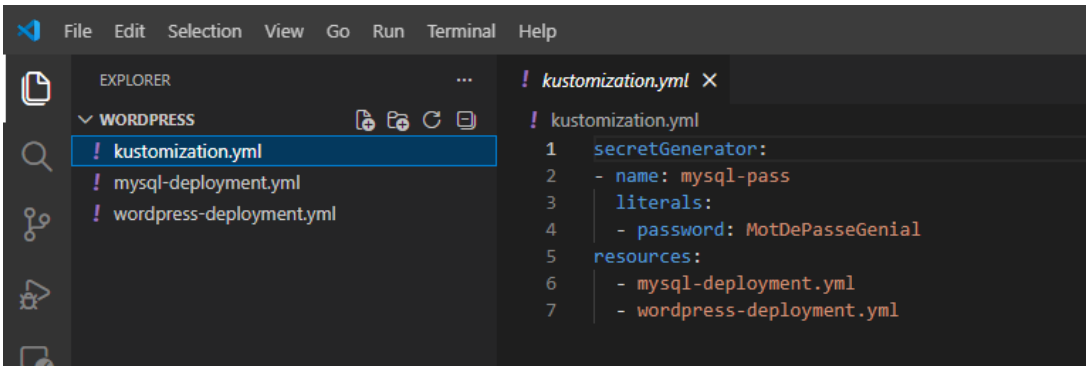

Non sécurisé
172.27.200.46:30100
User profile
Name: Bison Futé
Email:
Interests: DEVOPS
Edit Profile

6. Application.

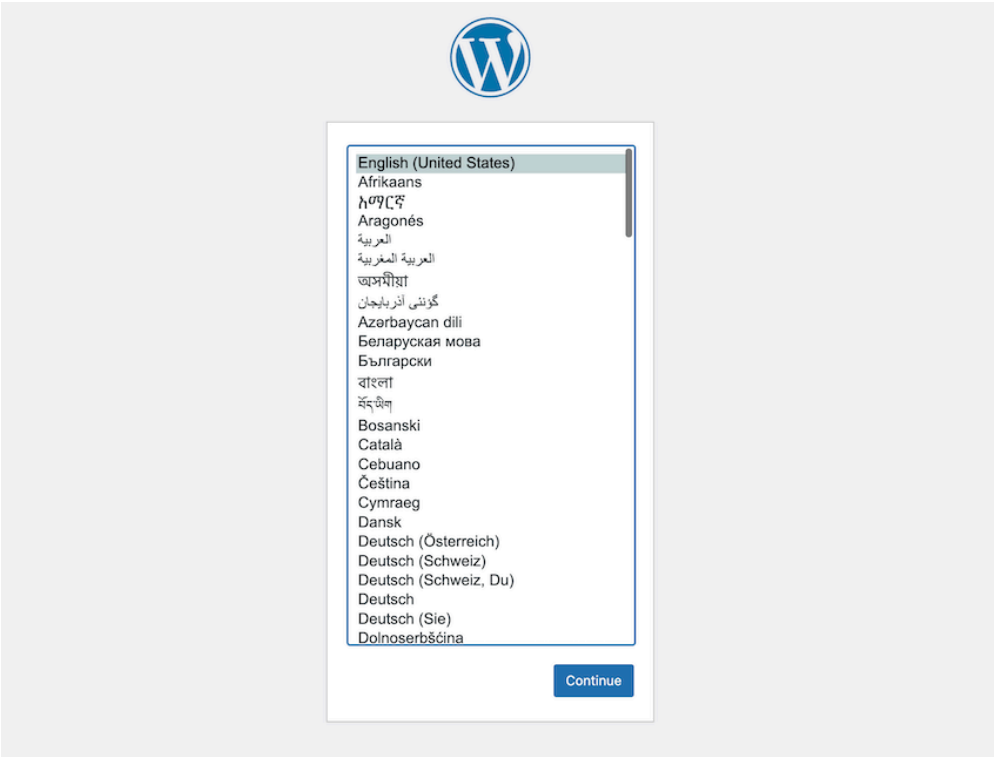
- Nous créons les différents fichiers nécessaire au déploiement de wordpress





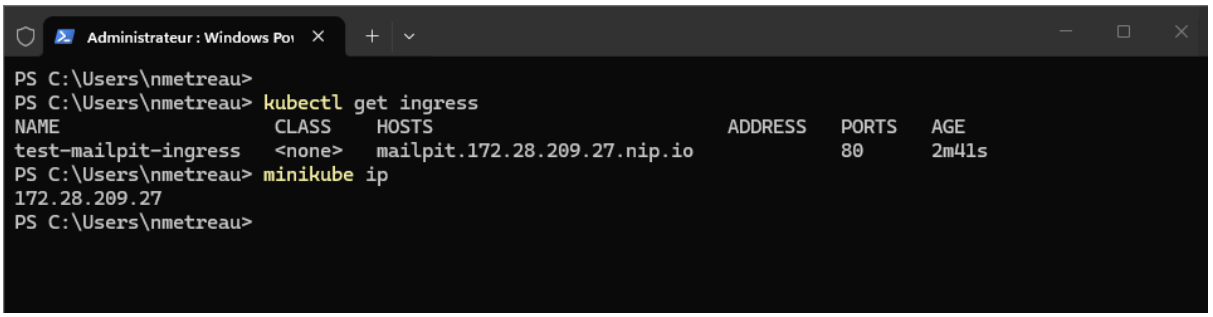


▪ Nous vérifions que cela marche

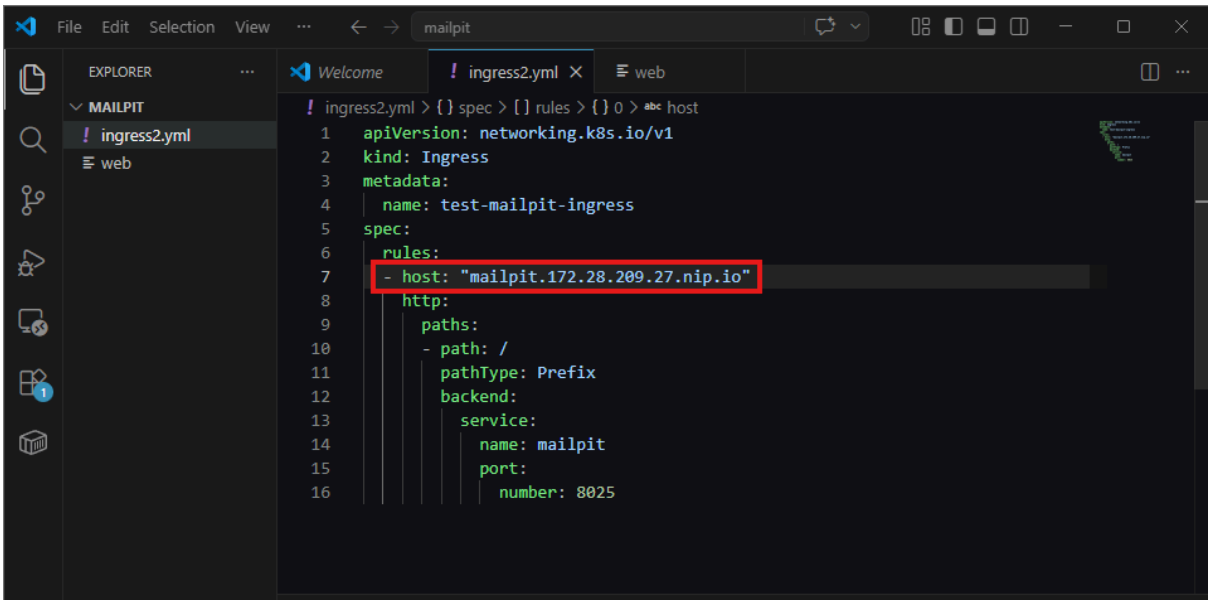


7. Suite Ingress (Mailpit) : hôtes virtuels et nom de domaine nip.io

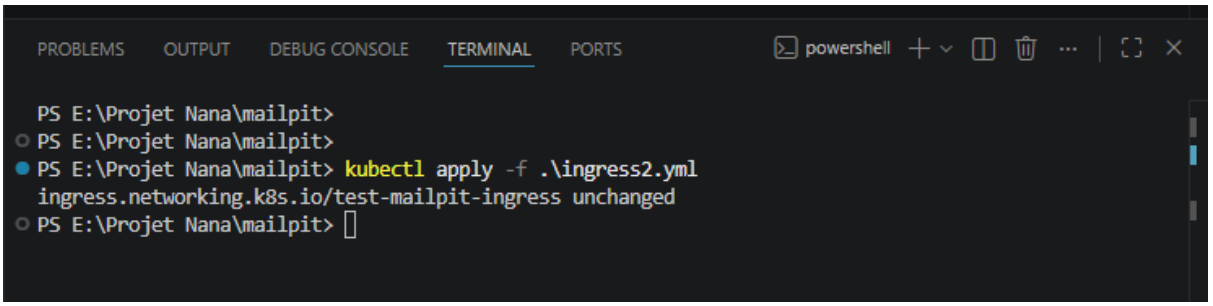
- Récupération des informations sur les objets ingress :



▪ Nous créons un hôte virtuel pour Mailpit : modification du contenu du fichier ingress.yaml afin de pointer la règle Ingress sur l'entrée DNS figurant ci-dessus :

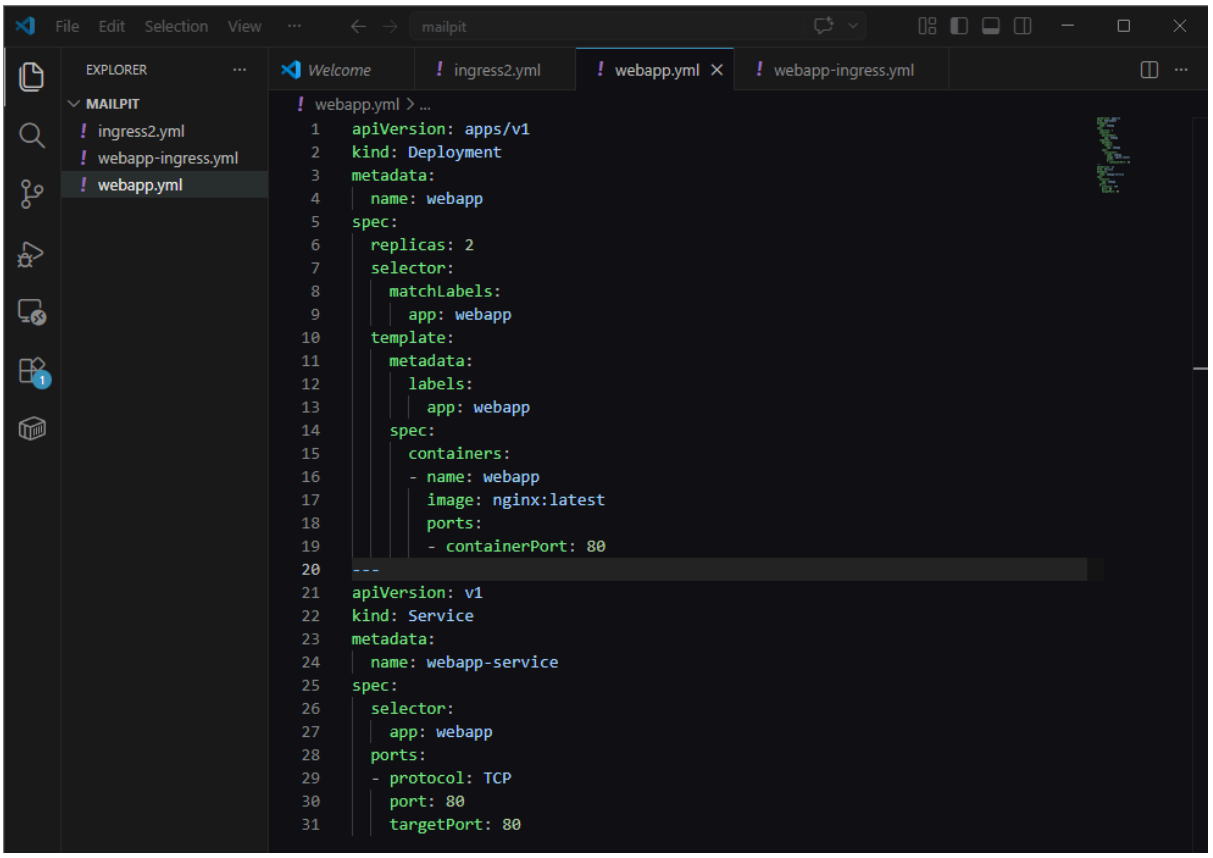


- Nous prenons en compte de la modification :

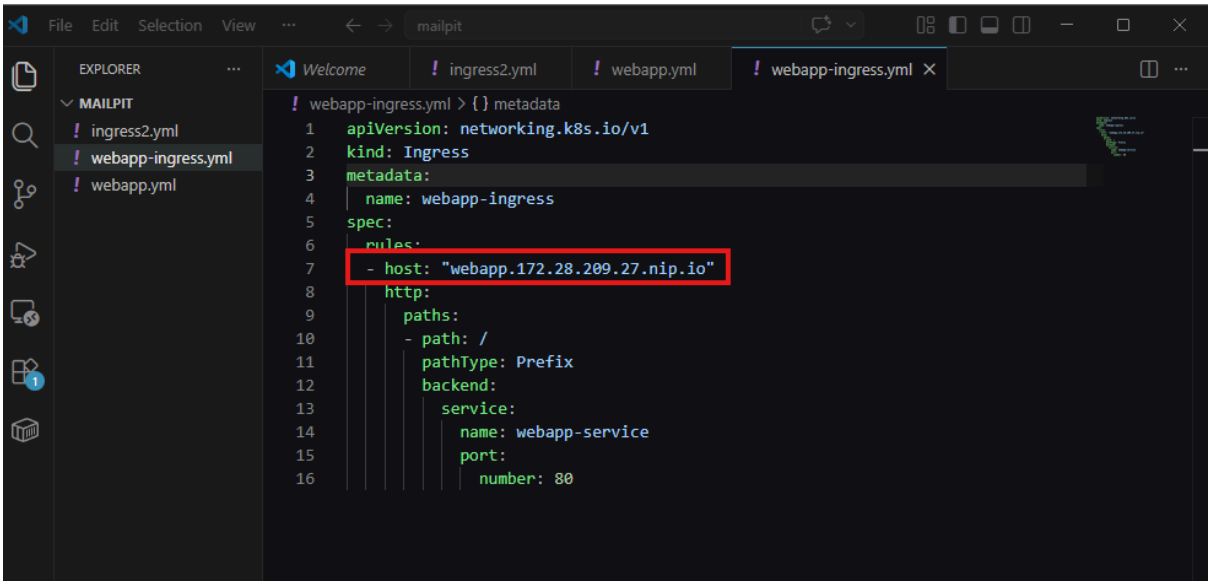


8. Hôtes virtuels et nom de domaine nip.io : autre exemple

- Création d'un objet Deployment et un objet Service :



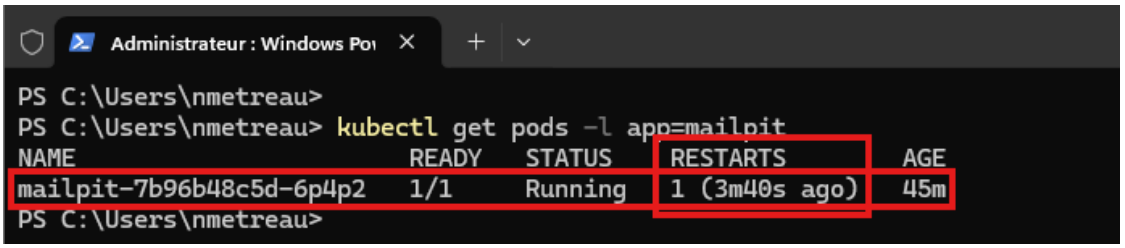
- Création d'un objet Ingress avec le champ host spécifiant le nom d'hôte virtuel



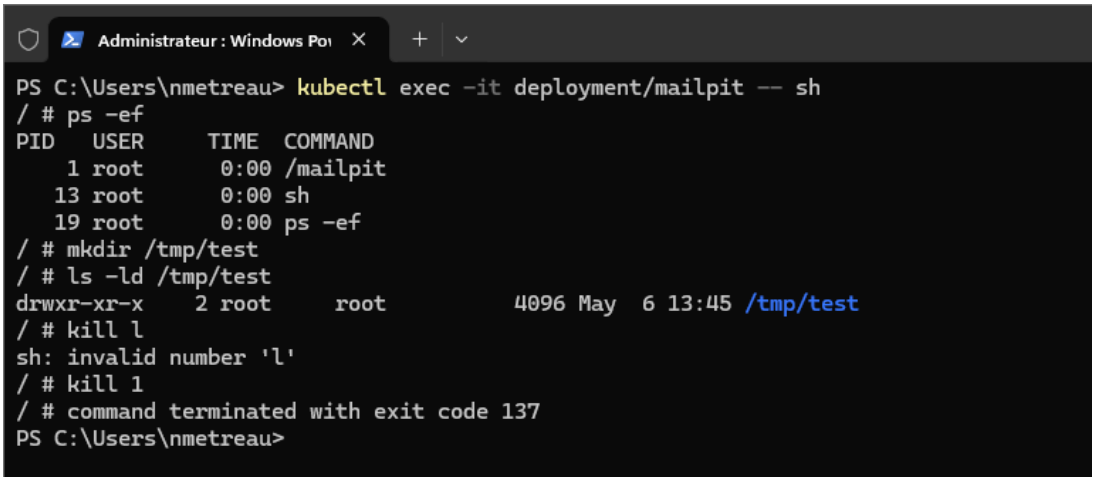
▪ Nous prenons en compte de 2 fichiers YAML et test :

9. Cycle de vie d'un conteneur dans Kubernetes

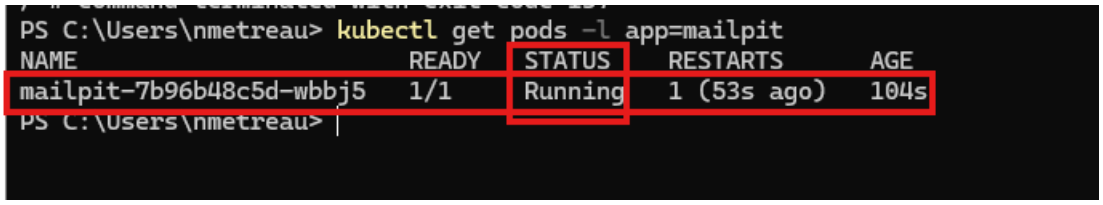
▪ Nous consultons l'état des pods de l'application :



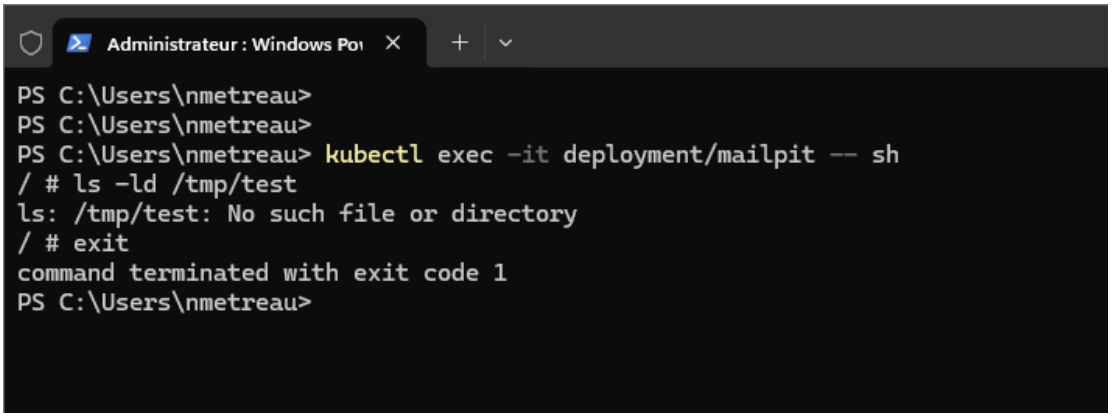
▪ Nous consultons l'état des pods de l'application portant le labell app=mailpit :



▪ Nous consultons l'état des pods de l'application :

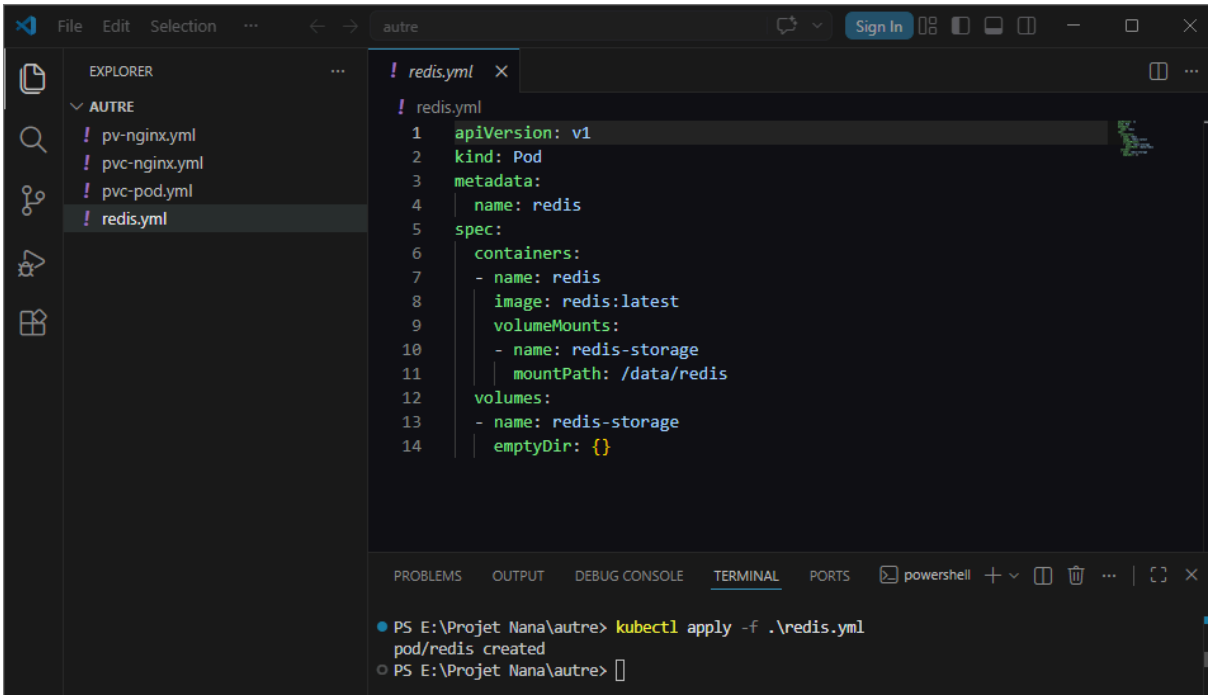


▪ Nous lançons à nouveau une connexion sur le conteneur Mailpit. Le répertoire /tmp/test n'est plus présent :

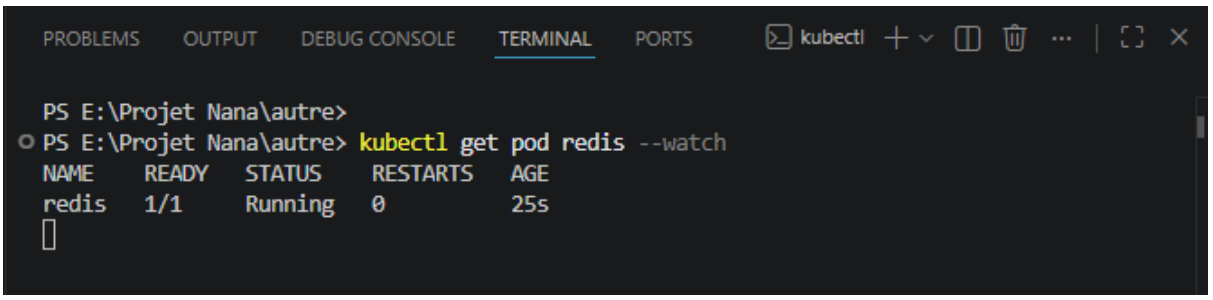


10. Persistance des données

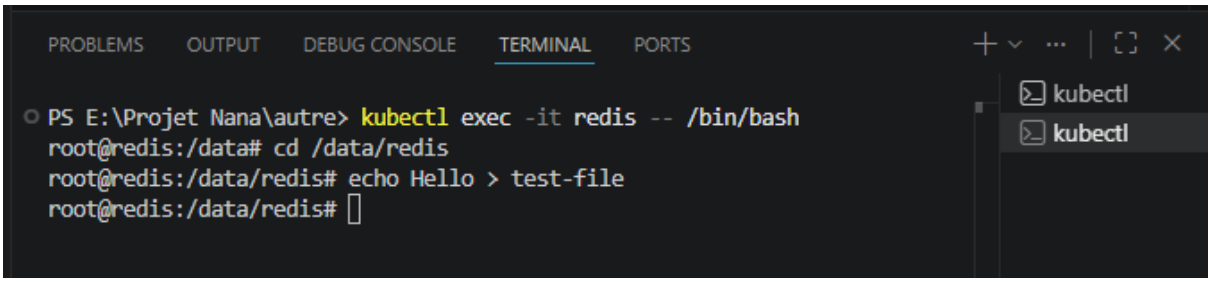
▪ Nous créons le fichier de configuration du Pod redis.yaml et nous créons le pod avec la commande kubectl apply -f :



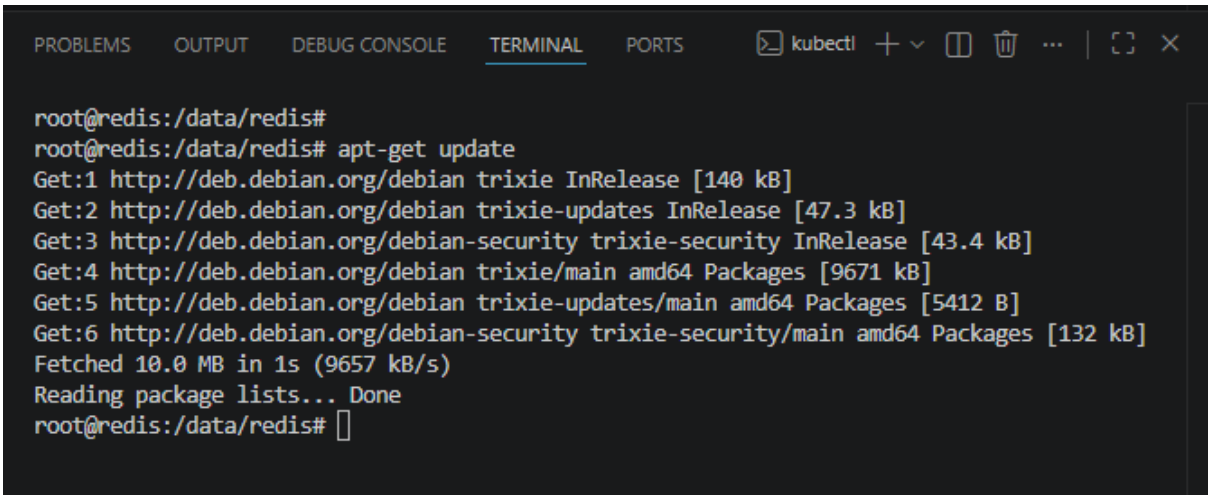
▪ Nous vérifions que le conteneur du pod soit en cours d'exécution puis nous surveillons les modifications apportées au pod :

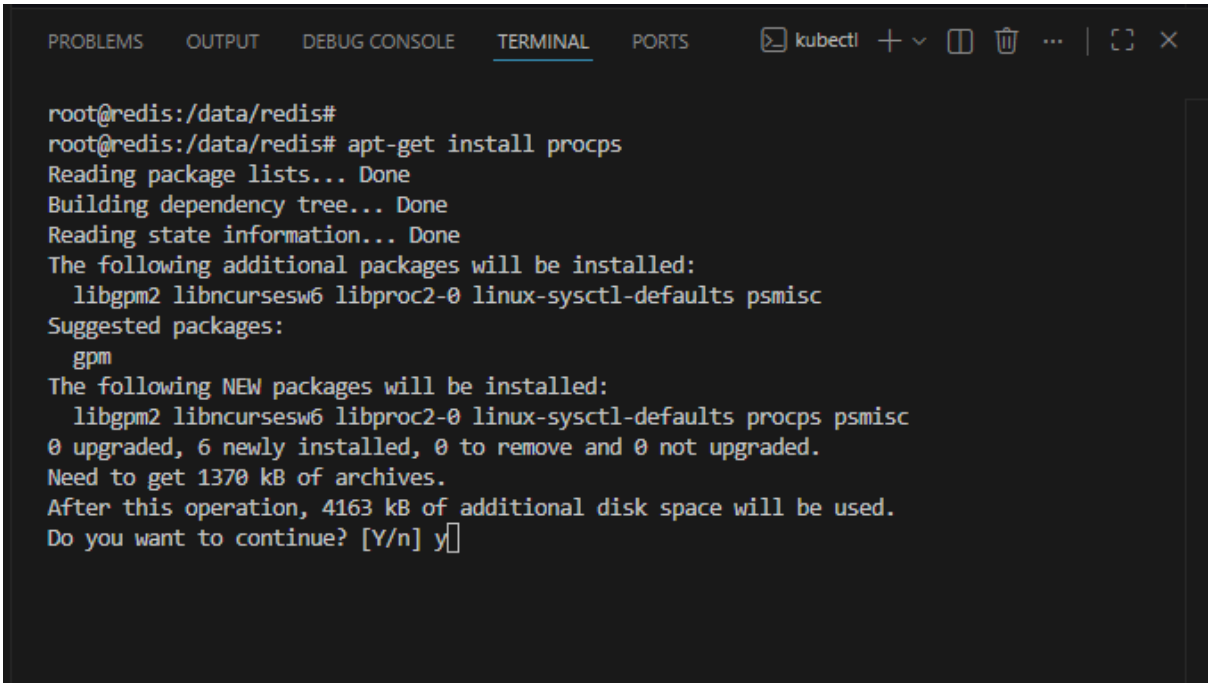


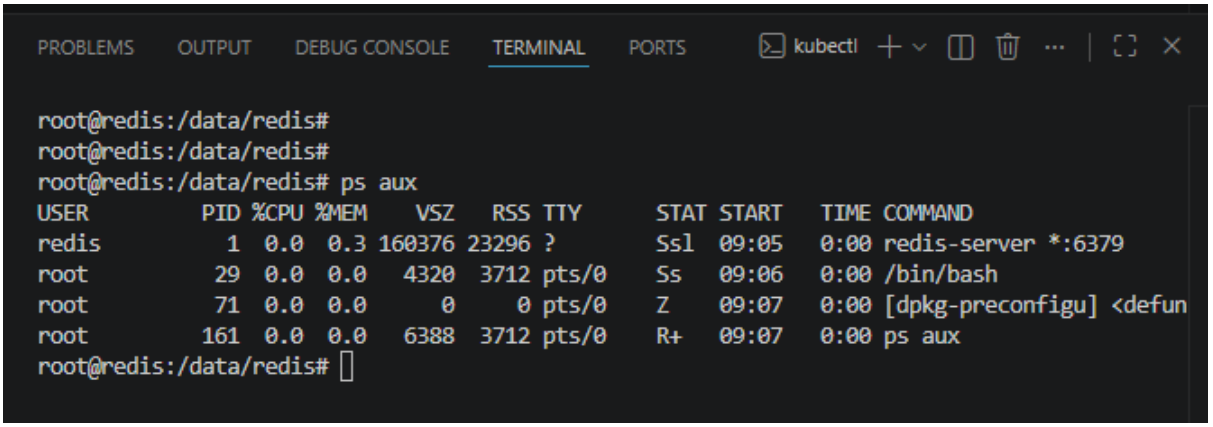
▪ Dans un second terminal, nous accédons à la console shell du conteneur en cours d'exécution puis nous créons un fichier dans /data/redis/ :



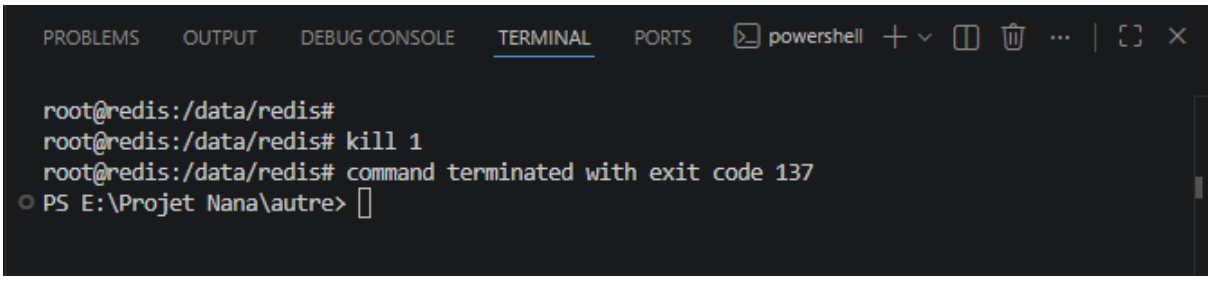
▪ Nous installons procps, puis listons les processus en cours d'exécution :



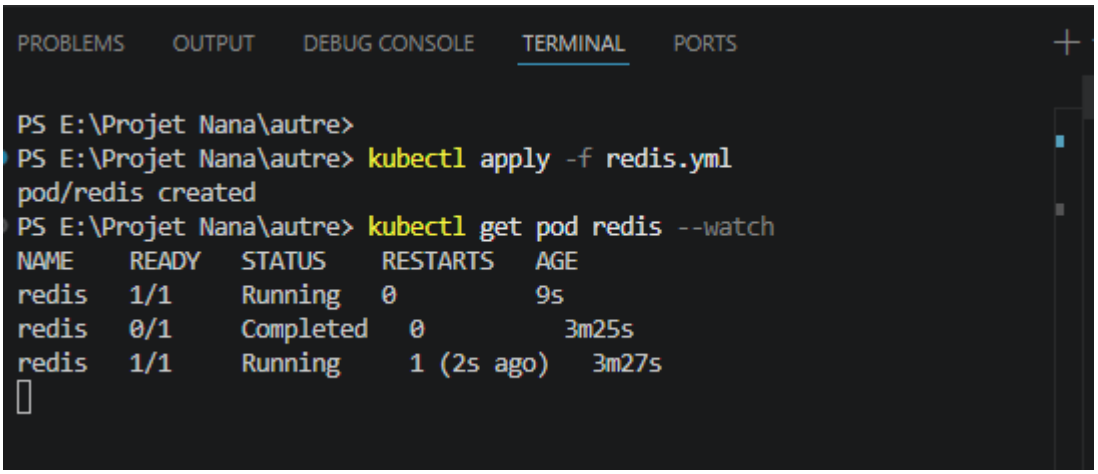




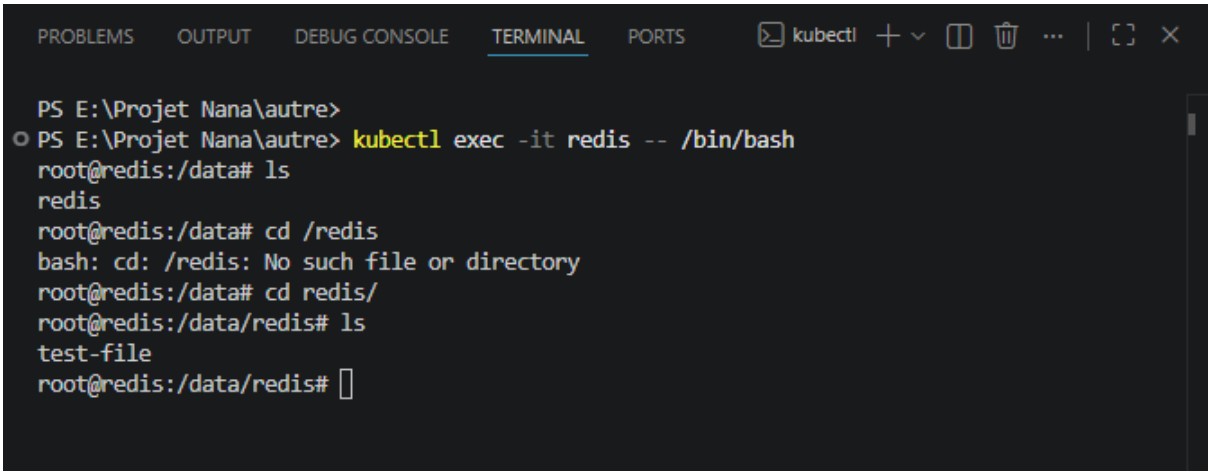
▪ Nous arrêtons le processus Redis :



▪ Nous revenons dans le premier terminal et constatons les changements apportés au Pod de Redis :



▪ Nous accédons à la console shell du conteneur redémarré et constatons que le fichier test-file est toujours présent dans le répertoire /data/redis :



▪ Nous supprimons le pod :

```
PS E:\Projet Nana\autre> kubectl delete pod redis
pod "redis" deleted from default namespace
PS E:\Projet Nana\autre>
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Projet Nana\autre>
PS E:\Projet Nana\autre> kubectl apply -f redis.yml
pod/redis created
PS E:\Projet Nana\autre> kubectl get pod redis --watch
NAME    READY   STATUS    RESTARTS      AGE
redis   1/1     Running   0             9s
redis   0/1     Completed   0           3m25s
redis   1/1     Running     1 (2s ago)  3m27s
redis   1/1     Terminating   1 (83s ago)   4m48s
redis   1/1     Terminating   1 (83s ago)   4m48s
redis   0/1     Completed     1 (83s ago)   4m48s
redis   0/1     Completed     1 (84s ago)   4m49s
redis   0/1     Completed     1 (84s ago)   4m49s
```

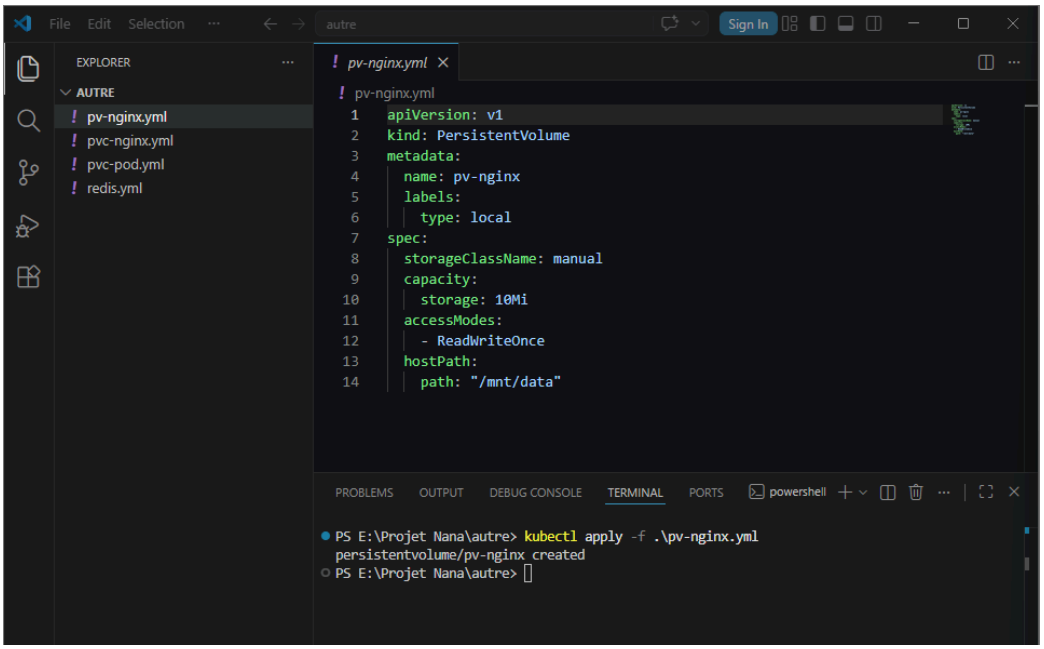
kubectl
powershell

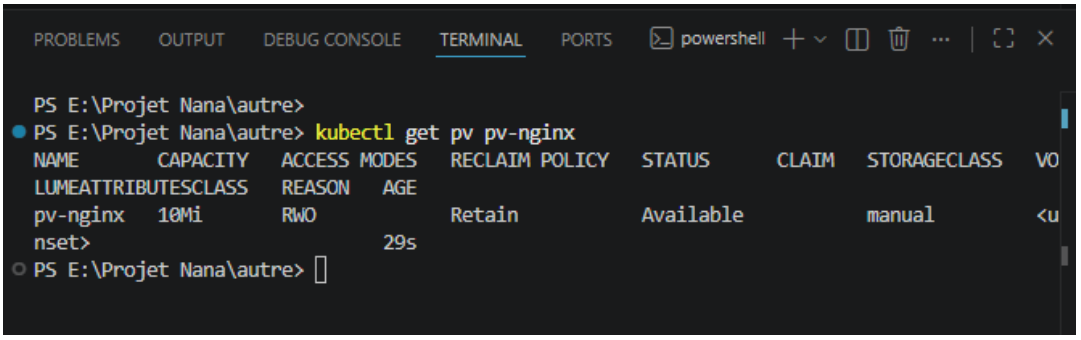
▪ Nous accédons à minikube via la commande minikube ssh puis nous créons un dossier /mnt/data.

- Dans le dossier /mnt/data, nous créons un fichier index.html.
- Nous testons l'existence du fichier index.html puis fermons la session shell sur le nœud du cluster.

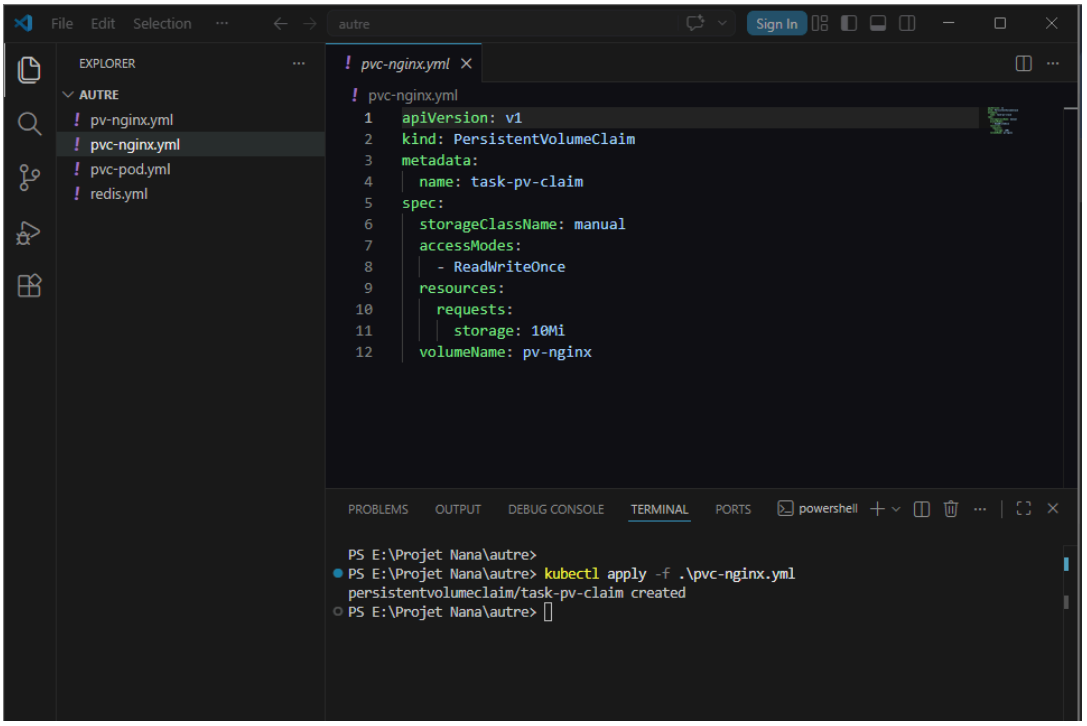
```
Administrateur : Windows Po
PS C:\Users\nmetreau>
PS C:\Users\nmetreau>
PS C:\Users\nmetreau> minikube ssh
                       _             _
            _         _ ( )           ( )
  ___ ___  (_)  ___  (_)| |/')  _   _ | |_      __
/' _ ` _ `\| |/' _ `\| || , <  ( ) ( )| '_`\  /'__`\
| ( ) ( ) || || ( ) || || |\`\ | (_) || |_) )(  ___/
(_) (_) (_)(_)(_) (_)(_)(_) (_)`\___/'(_,__/'`\____)

$ sudo mkdir /mnt/data
$ sudo sh -c "echo 'PV et PVC' > /mnt/data/index.html"
$ cat /mnt/data/index.html
PV et PVC
$ exit
logout
PS C:\Users\nmetreau>
```

▪ Création d'un PersistentVolume de type hostpath





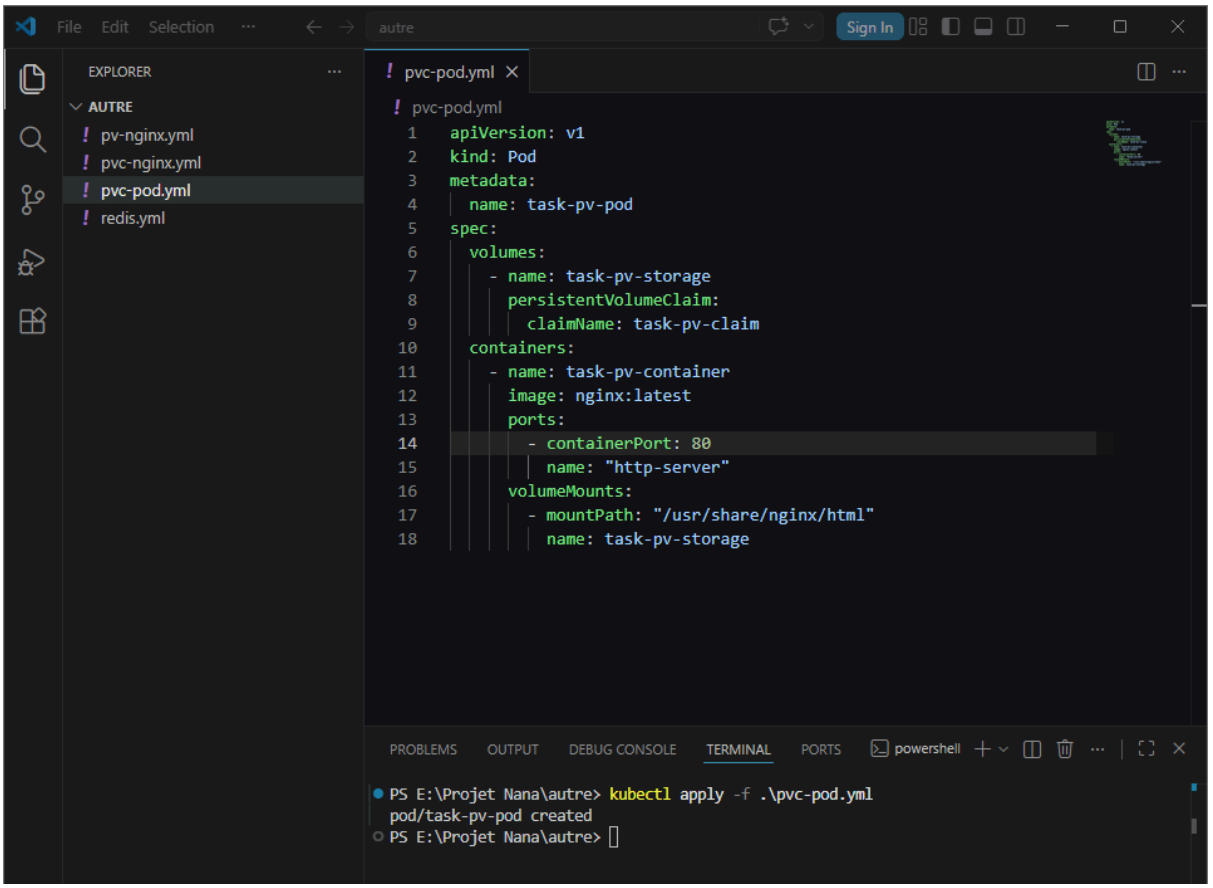
▪ Création d'un PersistentVolumeClaim :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell

PS E:\Projet Nana\autre> kubectl get pv pv-nginx
NAME       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                  STOR
AGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
pv-nginx   10Mi       RWO            Retain           Bound    default/task-pv-claim  manu
al         <unset>                          100s
PS E:\Projet Nana\autre>
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell

PS E:\Projet Nana\autre> kubectl get pvc
NAME            STATUS   VOLUME     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUT
ESCLASS   AGE
task-pv-claim   Bound    pv-nginx   10Mi       RWO            manual         <unset>
          58s
PS E:\Projet Nana\autre>
```

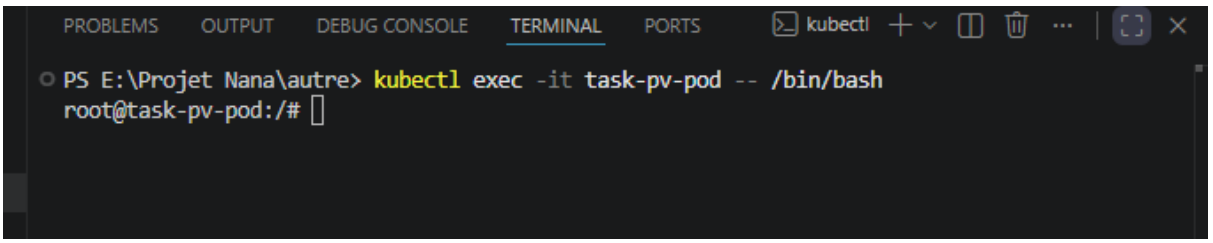
- Création d'un Pod qui utilise le PersistentVolumeClaim comme volume de stockage

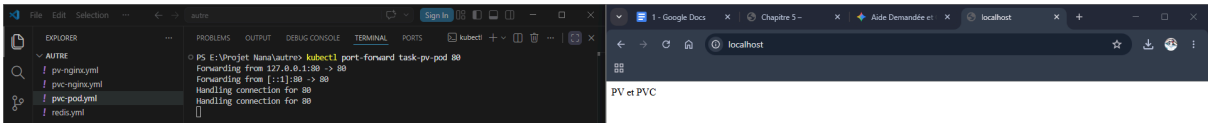


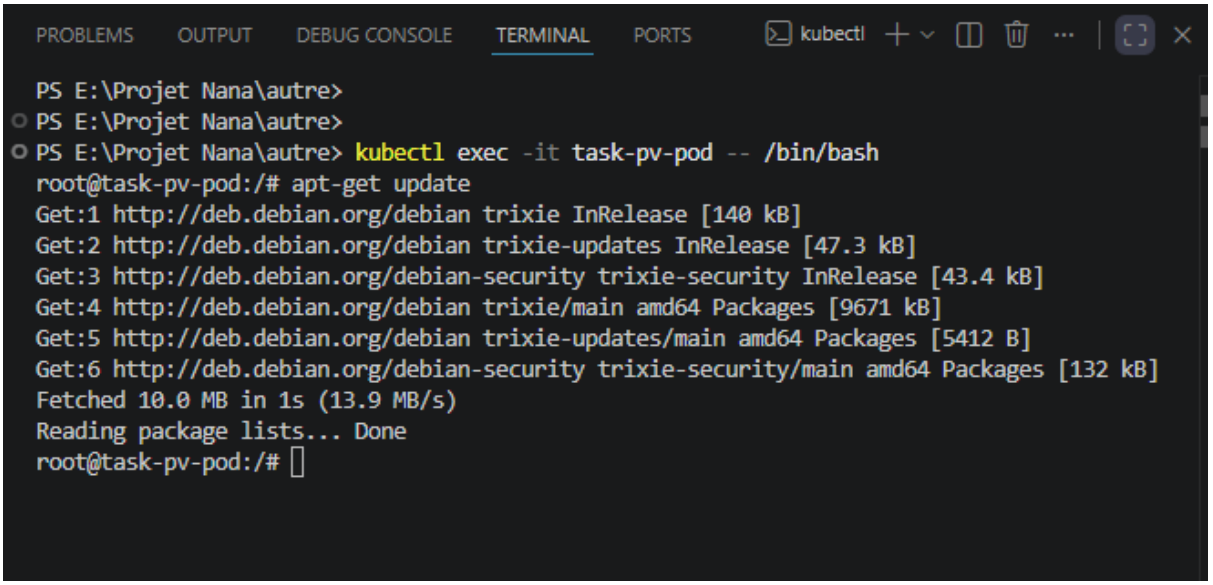
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell

PS E:\Projet Nana\autre> kubectl get pod task-pv-pod
NAME          READY   STATUS    RESTARTS   AGE
task-pv-pod   1/1     Running   0          37s
PS E:\Projet Nana\autre> kubectl describe pod task-pv-pod
Name:             task-pv-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.28.158.3
Start Time:       Thu, 07 May 2026 11:24:35 +0200
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.0.8
IPs:
  IP:  10.244.0.8
Containers:
  task-pv-container:
    Container ID:   docker://0b3c866c7dd75695e7a5cb264afd7d1c282c00eeabab116cb8574a3eb1745f
ca
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:6e23479198b998e5e25921dff8455837c7636a67
111a04a635cf1bb363d199dc
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell

PS E:\Projet Nana\autre>
PS E:\Projet Nana\autre> kubectl get pod task-pv-pod
NAME          READY   STATUS    RESTARTS   AGE
task-pv-pod   1/1     Running   0          85s
PS E:\Projet Nana\autre>
```

```
PS E:\Projet Nana\autre>
PS E:\Projet Nana\autre> kubectl describe pod task-pv-pod
Name:             task-pv-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.28.158.3
Start Time:       Thu, 07 May 2026 11:24:35 +0200
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.0.8
IPs:
  IP:  10.244.0.8
Containers:
  task-pv-container:
    Container ID:   docker://0b3c866c7dd75695e7a5cb264afd7d1c282c00eeabab116cb8574a3eb1745f
ca
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:6e23479198b998e5e25921dff8455837c7636a67
111a04a635cf1bb363d199dc
    Port:           80/TCP (http-server)
    Host Port:      0/TCP (http-server)
    State:          Running
      Started:      Thu, 07 May 2026 11:24:37 +0200
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /usr/share/nginx/html from task-pv-storage (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-xt6nf (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  task-pv-storage:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same n
amespace)
    ClaimName:  task-pv-claim
    ReadOnly:   false
  kube-api-access-xt6nf:
    Type:                    Projected (a volume that contains injected data from multiple
sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    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
```

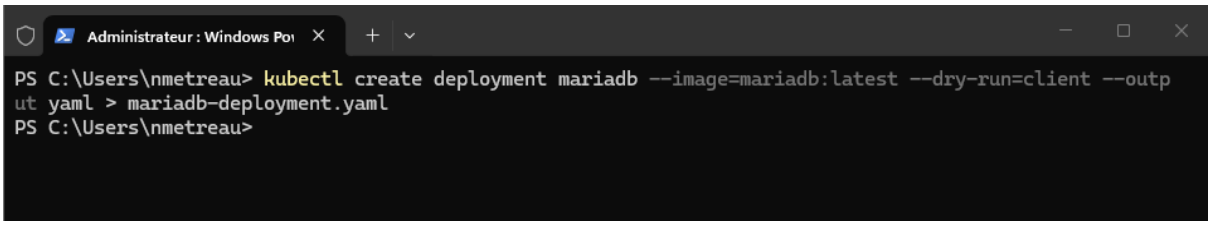

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
kubectl
PS E:\Projet Nana\autre> kubectl exec -it task-pv-pod -- /bin/bash
root@task-pv-pod:/#

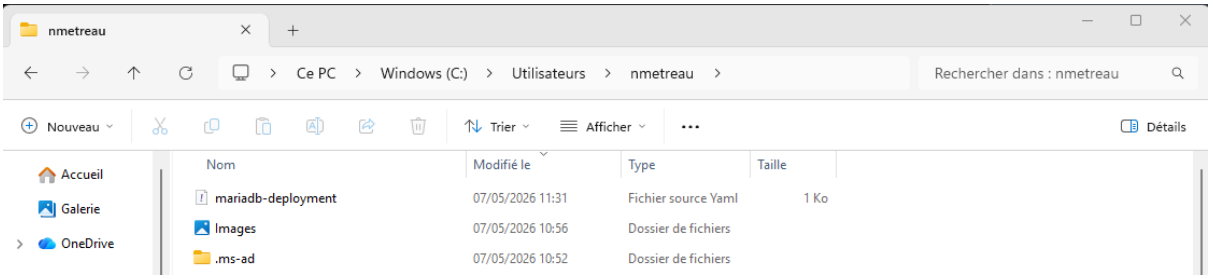

PS E:\Projet Nana\autre> kubectl port-forward task-pv-pod 80
Forwarding from 127.0.0.1:80 -> 80
Forwarding from [::1]:80 -> 80
Handling connection for 80
Handling connection for 80
localhost
PV et PVC


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
kubectl
PS E:\Projet Nana\autre>
PS E:\Projet Nana\autre>
PS E:\Projet Nana\autre> kubectl exec -it task-pv-pod -- /bin/bash
root@task-pv-pod:/# apt-get update
Get:1 http://deb.debian.org/debian trixie InRelease [140 kB]
Get:2 http://deb.debian.org/debian trixie-updates InRelease [47.3 kB]
Get:3 http://deb.debian.org/debian-security trixie-security InRelease [43.4 kB]
Get:4 http://deb.debian.org/debian trixie/main amd64 Packages [9671 kB]
Get:5 http://deb.debian.org/debian trixie-updates/main amd64 Packages [5412 B]
Get:6 http://deb.debian.org/debian-security trixie-security/main amd64 Packages [132 kB]
Fetched 10.0 MB in 1s (13.9 MB/s)
Reading package lists... Done
root@task-pv-pod:/#

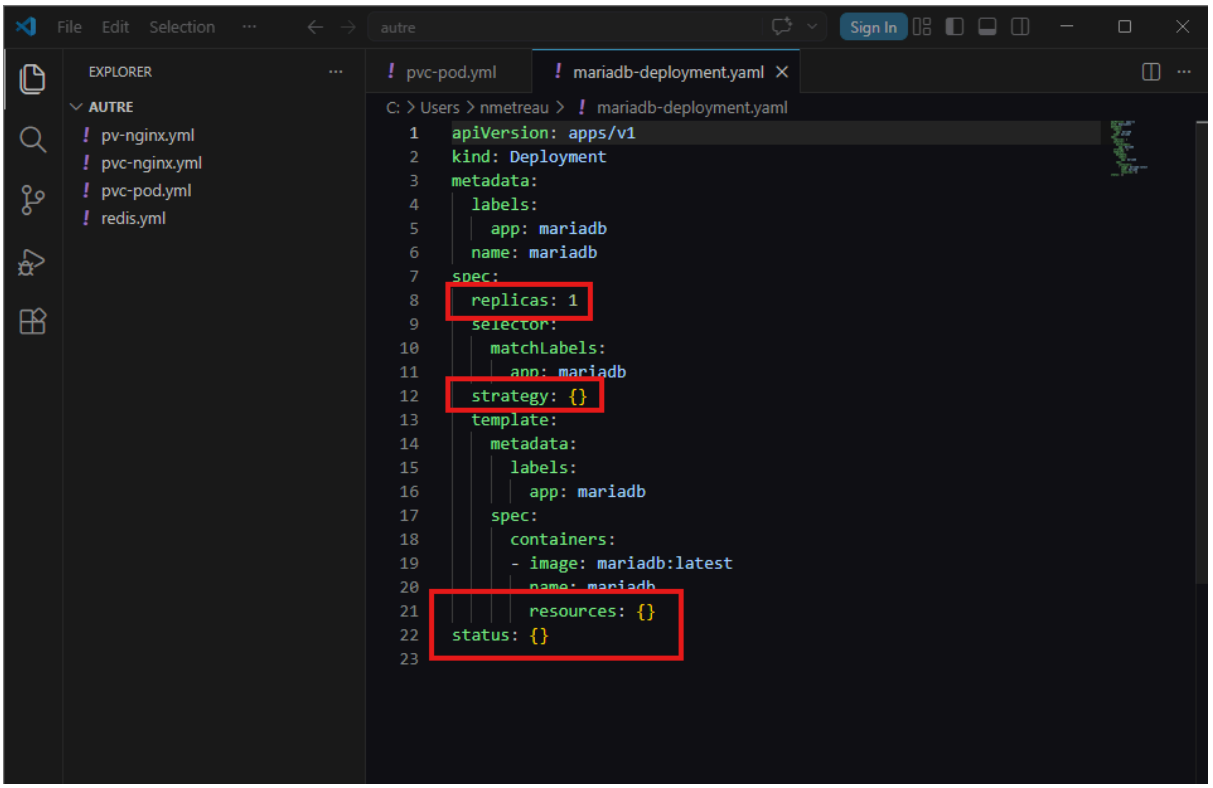
11. Hébergement d'un groupe de pods fonctionnant en cluster

▪ Version initiale du fichier de déploiement mariadb-deployment.yaml :

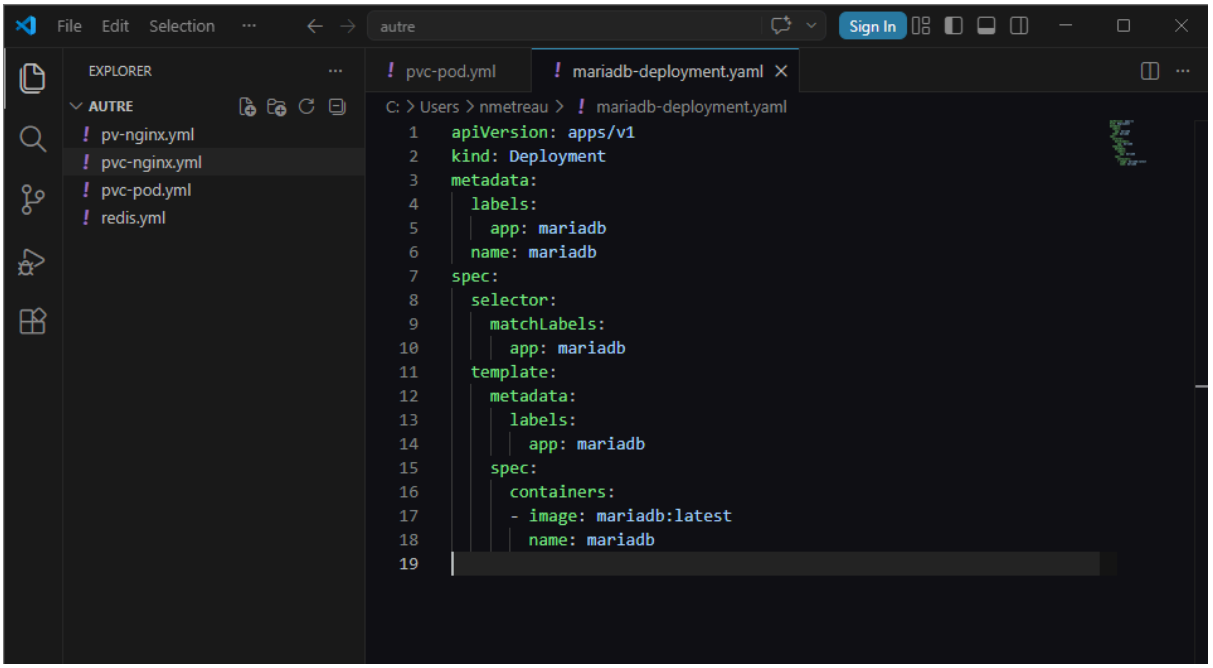




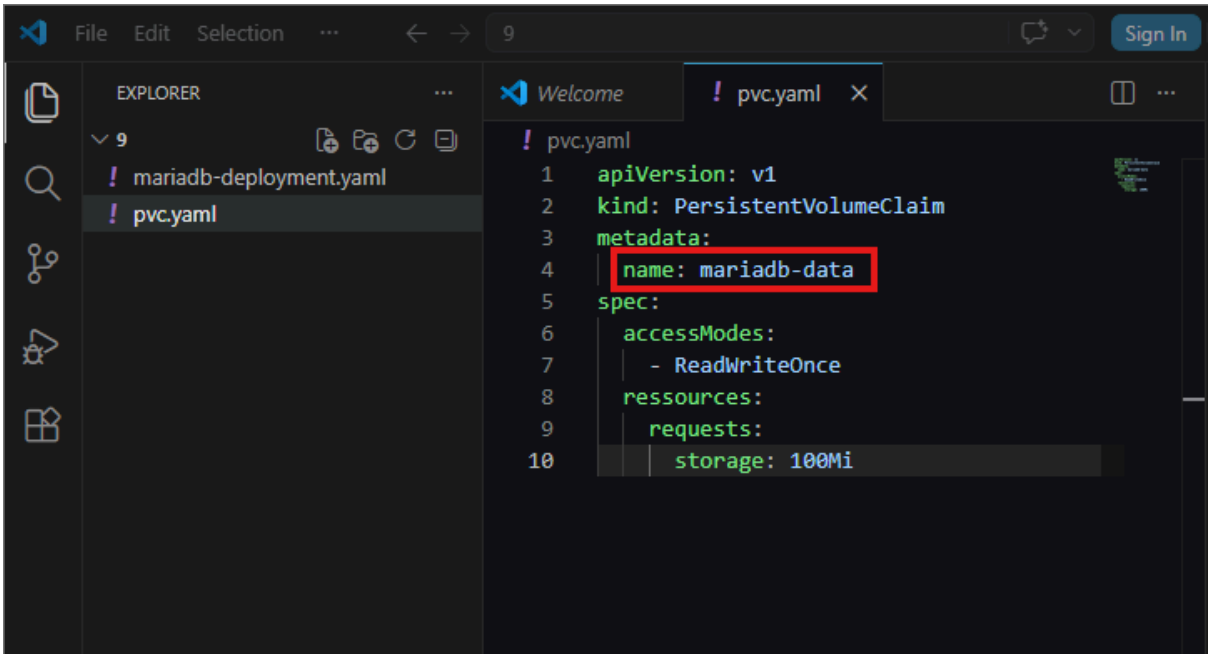
▪ Suppression des champs inutiles :

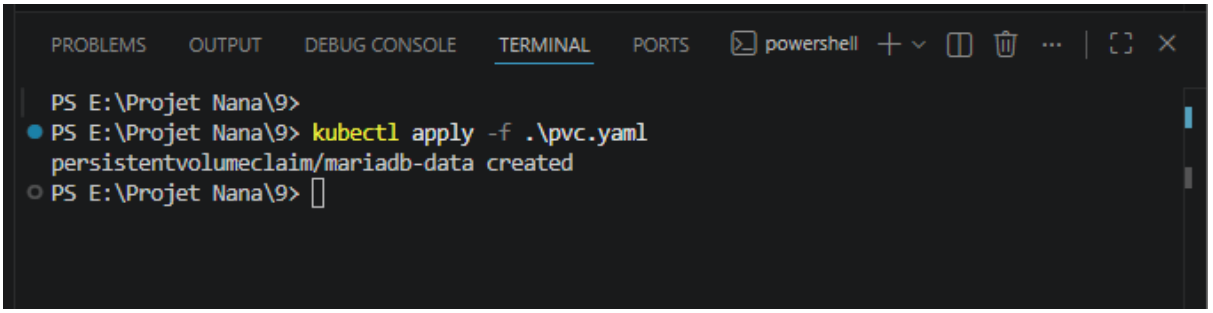


▪ Contenu après modification :



▪ Demande de volume persistant :

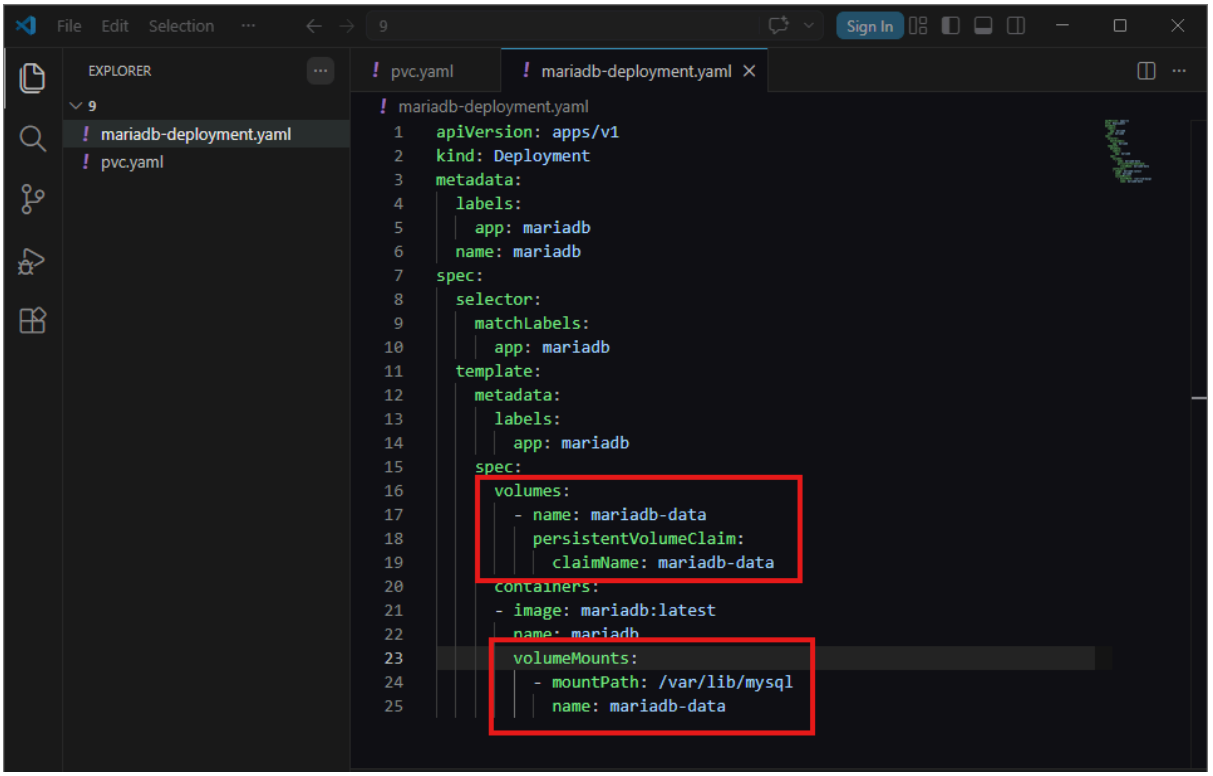




▪ Etat de la demande de volume persistant :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Projet Nana\9>
PS E:\Projet Nana\9> kubectl get pvc mariadb-data
NAME           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mariadb-data   Bound    pvc-159e048e-88f7-47b0-bd54-49523a51f23b   100Mi      RWO            standard       <unset>                 30s
PS E:\Projet Nana\9>
```

▪ Indication au conteneur de MariaDB d'utiliser le volume persistant :



▪ Le répertoire /var/lib/mysql est l'emplacement standard où sont stockés les fichiers des bases de données.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell

PS E:\Projet Nana\9> kubectl apply -f .\mariadb-deployment.yaml
deployment.apps/mariadb created
PS E:\Projet Nana\9>
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell

PS E:\Projet Nana\9> kubectl get pod -l app=mariadb
NAME                       READY   STATUS    RESTARTS      AGE
mariadb-5c476b5c79-lwtnd   0/1     Error     2 (29s ago)   39s
PS E:\Projet Nana\9>
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                                    powershell

PS E:\Projet Nana\9>
PS E:\Projet Nana\9> kubectl logs -l app=mariadb
2026-05-07 09:51:44+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2+maria~ubu2404 started.
2026-05-07 09:51:45+00:00 [Warn] [Entrypoint]: /sys/fs/cgroup///memory.pressure not writable, functionality unavailable to MariaDB
2026-05-07 09:51:45+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2026-05-07 09:51:45+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2+maria~ubu2404 started.
2026-05-07 09:51:45+00:00 [ERROR] [Entrypoint]: Database is uninitialized and password option is not specified
        You need to specify one of MARIADB_ROOT_PASSWORD, MARIADB_ROOT_PASSWORD_HASH, MARIADB_ALLOW_EMPTY_ROOT_PASSWORD and MARIADB_RANDOM_ROOT_PASSWORD
PS E:\Projet Nana\9>
```

▪ Lors du premier démarrage, MariaDB a besoin de connaître le mot de passe de l'administrateur (champ env dans la définition du conteneur) :

pvc.yaml | mariadb-deployment.yaml

mariadb-deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: mariadb
  name: mariadb
spec:
  selector:
    matchLabels:
      app: mariadb
  template:
    metadata:
      labels:
        app: mariadb
    spec:
      volumes:
        - name: mariadb-data
          persistentVolumeClaim:
            claimName: mariadb-data
      containers:
      - image: mariadb:latest
        name: mariadb
        volumeMounts:
          - mountPath: /var/lib/mysql
            name: mariadb-data
        env:
          - name: MARIADB_ROOT_PASSWORD
            value: mot-de-pass-root
```

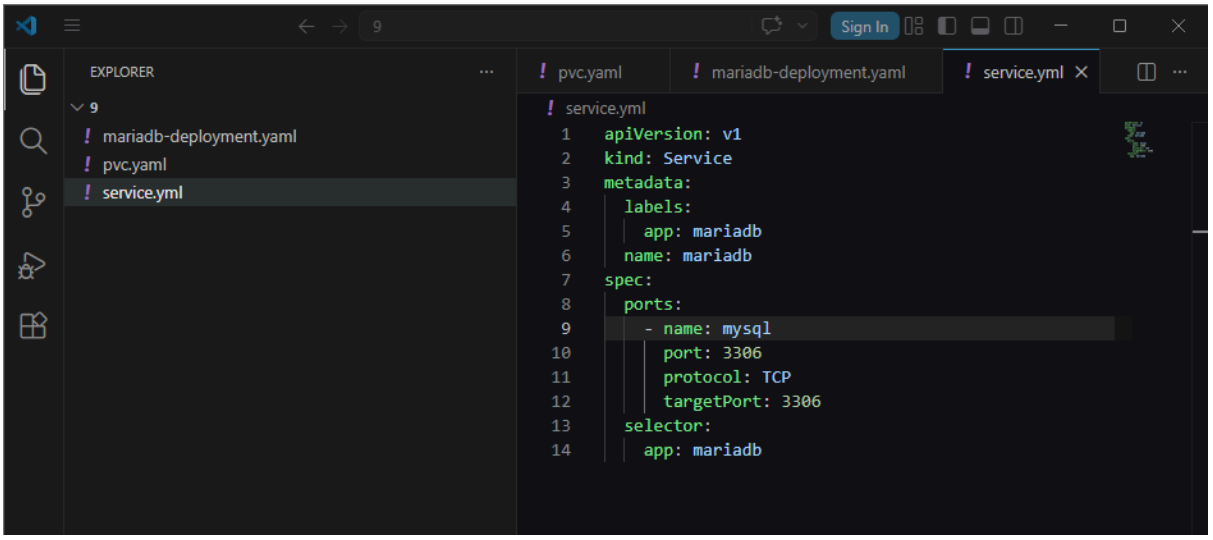
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Projet Nana\9> kubectl apply -f .\mariadb-deployment.yaml
deployment.apps/mariadb configured
PS E:\Projet Nana\9>
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Projet Nana\9> kubectl get pod -l app=mariadb
NAME                       READY   STATUS    RESTARTS   AGE
mariadb-b5ff7ff4f-hfj72    1/1     Running   0          21s
PS E:\Projet Nana\9>
```

▪ Connexion au conteneur afin de tester le fonctionnement de la bdd (commande mariadb-admin status).

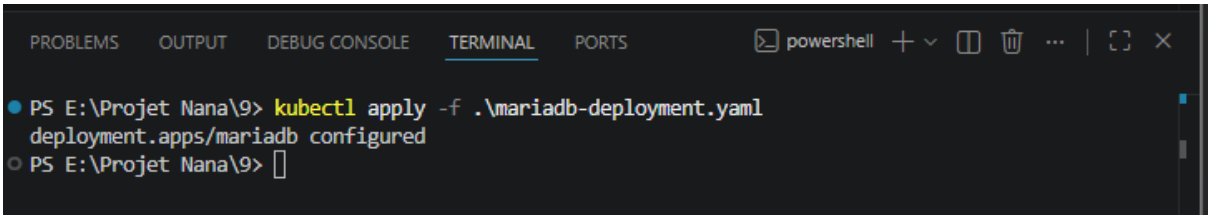
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    powershell

PS E:\Projet Nana\9>
PS E:\Projet Nana\9> kubectl exec -it deployment/mariadb -- bash
root@mariadb-b5ff7ff4f-hfj72:/# mariadb-admin status -p$MARIADB_ROOT_PASSWORD
Uptime: 82  Threads: 1  Questions: 1  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second avg: 0.012
root@mariadb-b5ff7ff4f-hfj72:/# exit
exit
PS E:\Projet Nana\9>
```

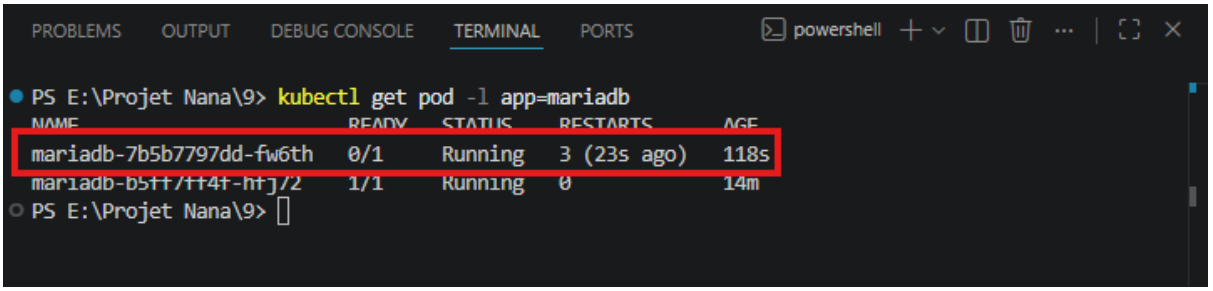
▪ Création du service (fichier service.yaml) :




```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  ...  powershell

PS E:\Projet Nana\9> kubectl apply -f .\service.yml
service/mariadb created
PS E:\Projet Nana\9>
```

▪ Surveillance de la base de données :

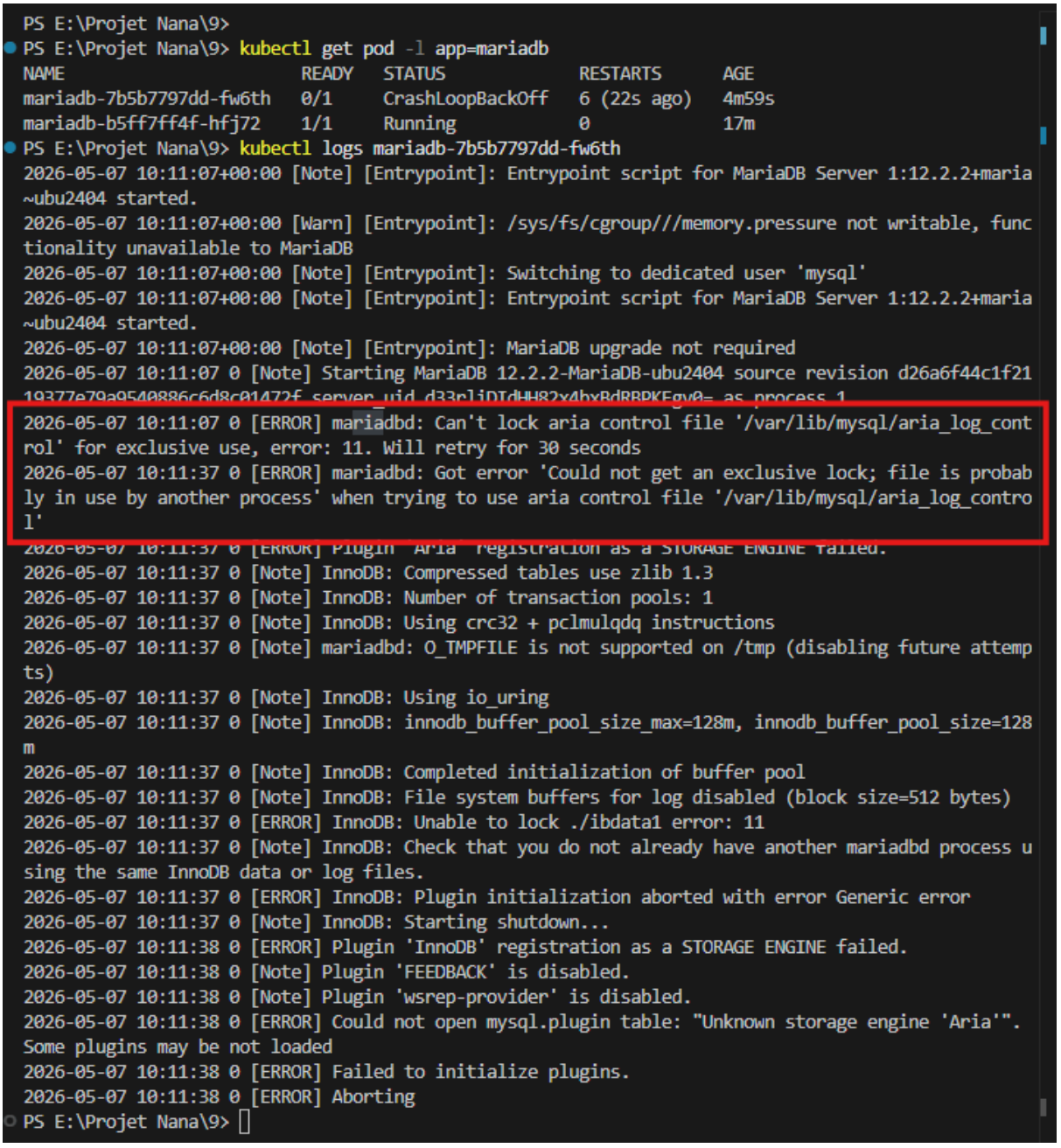




```
PS E:\Projet Nana\9> kubectl describe pods mariadb-7b5b7797dd-fw6th
Name:             mariadb-7b5b7797dd-fw6th
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.28.158.3
Start Time:       Thu, 07 May 2026 12:07:01 +0200
Labels:           app=mariadb
                  pod-template-hash=7b5b7797dd
Annotations:      <none>
Status:           Running
IP:               10.244.0.11
IPs:
  IP:           10.244.0.11
Controlled By:  ReplicaSet/mariadb-7b5b7797dd
Containers:
  mariadb:
    Container ID:   docker://a88b0402f5ec2eb235da14470243f7b108948965c2ac9e5f8ad63da44c45c3ed
    Image:          mariadb:latest
    Image ID:       docker-pullable://mariadb@sha256:e0236fc6386e7eacd9359e59d0a078bd7aa0d18280d36
d13061121bedeaee903
    Port:           <none>
    Host Port:      <none>
    State:          Waiting
      Reason:       CrashLoopBackOff
    Last State:     Terminated
      Reason:       Error
      Exit Code:    1
      Started:      Thu, 07 May 2026 12:09:09 +0200
      Finished:     Thu, 07 May 2026 12:09:40 +0200
    Ready:          False
    Restart Count:  4
    Liveness:       exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s timeout=1s
 period=10s #success=1 #failure=3
    Readiness:      exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s timeout=1s
 period=10s #success=1 #failure=3
    Startup:        exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s timeout=1s
 period=10s #success=1 #failure=3
    Environment:
      MARIADB_ROOT_PASSWORD:  mot-de-pass-root
    Mounts:
      /var/lib/mysql from mariadb-data (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-cn8mx (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       False
  ContainersReady             False
  PodScheduled                True
Volumes:
  mariadb-data:
```

```
Volumes:
  mariadb-data:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespac
e)
    ClaimName:  mariadb-data
    ReadOnly:   false
  kube-api-access-cn8mx:
    Type:                    Projected (a volume that contains injected data from multiple sources
)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    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  2m58s                default-scheduler  Successfully assigned default/maria
db-7b5b7797dd-fw6th to minikube
  Normal   Pulled     2m57s                kubelet            Successfully pulled image "mariadb:
latest" in 904ms (904ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     2m25s                kubelet            Successfully pulled image "mariadb:
latest" in 847ms (847ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     113s                 kubelet            Successfully pulled image "mariadb:
latest" in 880ms (880ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     82s                  kubelet            Successfully pulled image "mariadb:
latest" in 849ms (849ms including waiting). Image size: 335568219 bytes.
  Normal   Pulling    51s (x5 over 2m58s)  kubelet            Pulling image "mariadb:latest"
  Normal   Created    50s (x5 over 2m57s)  kubelet            Container created
  Normal   Started    50s (x5 over 2m57s)  kubelet            Container started
  Normal   Pulled     50s                  kubelet            Successfully pulled image "mariadb:
latest" in 882ms (882ms including waiting). Image size: 335568219 bytes.
  Warning  Unhealthy  28s (x15 over 2m48s)  kubelet           Startup probe failed: mariadb-admin
: connect to server at 'localhost' failed
error: 'Can't connect to local server through socket '/run/mysqld/mysqld.sock' (2)'
Check that mariadbd is running and that the socket: '/run/mysqld/mysqld.sock' exists!
  Normal   Killing  28s (x5 over 2m28s)  kubelet  Container mariadb failed startup probe, will be
restarted
  Warning  BackOff  18s (x3 over 19s)    kubelet  Back-off restarting failed container mariadb in
pod mariadb-7b5b7797dd-fw6th_default(858b59ef-931e-46d4-805b-125853884cd3)
PS E:\Projet Nana\9>
```

```
PS E:\Projet Nana\9>
PS E:\Projet Nana\9> kubectl get pod -l app=mariadb
NAME                       READY   STATUS    RESTARTS      AGE
mariadb-7b5b7797dd-fw6th   0/1     Running   6 (12s ago)   4m17s
mariadb-b5ff7ff4f-hfj72    1/1     Running   0             16m
PS E:\Projet Nana\9>
```



Le conteneur n'arrive pas à démarrer en raison de la présence d'un lock dans les fichiers de bdd. Ce lock s'explique en raison du fait qu'un pod est déjà démarré et que celui-ci utilise déjà ces fichiers.

▪ Le problème vient du fait qu'une bdd ne peut pas partager ses fichiers.

mariadb-deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: mariadb
  name: mariadb
spec:
  selector:
    matchLabels:
      app: mariadb
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: mariadb
    spec:
      volumes:
        - name: mariadb-data
          persistentVolumeClaim:
            claimName: mariadb-data
      containers:
      - image: mariadb:latest
        name: mariadb
        volumeMounts:
          - mountPath: /var/lib/mysql
            name: mariadb-data
        env:
          - name: MARIADB_ROOT_PASSWORD
            value: mot-de-pass-root
        startupProbe: &probe
            exec:
```

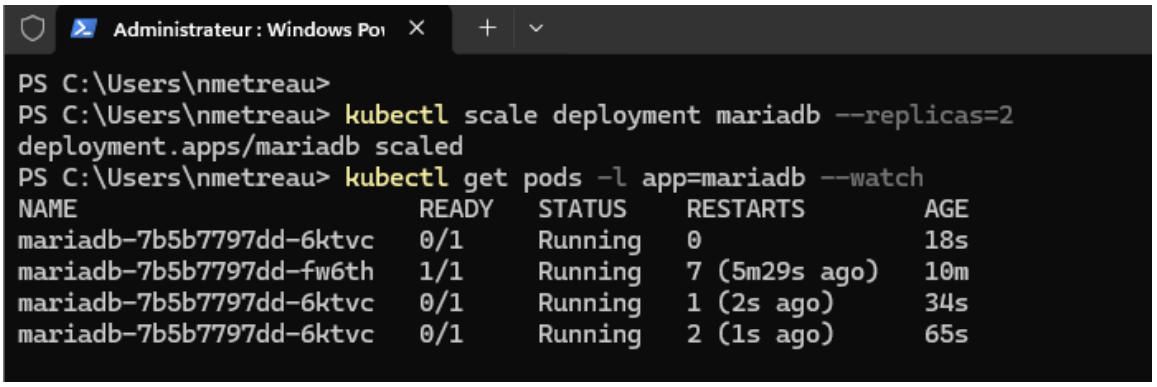
- Dans le cas où le pod reste dans l'état CrashLoopBackOff, nous pouvons supprimer le pod

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS        powershell

PS E:\Projet Nana\9>
PS E:\Projet Nana\9> kubectl apply -f .\mariadb-deployment.yaml
deployment.apps/mariadb configured
PS E:\Projet Nana\9>
```

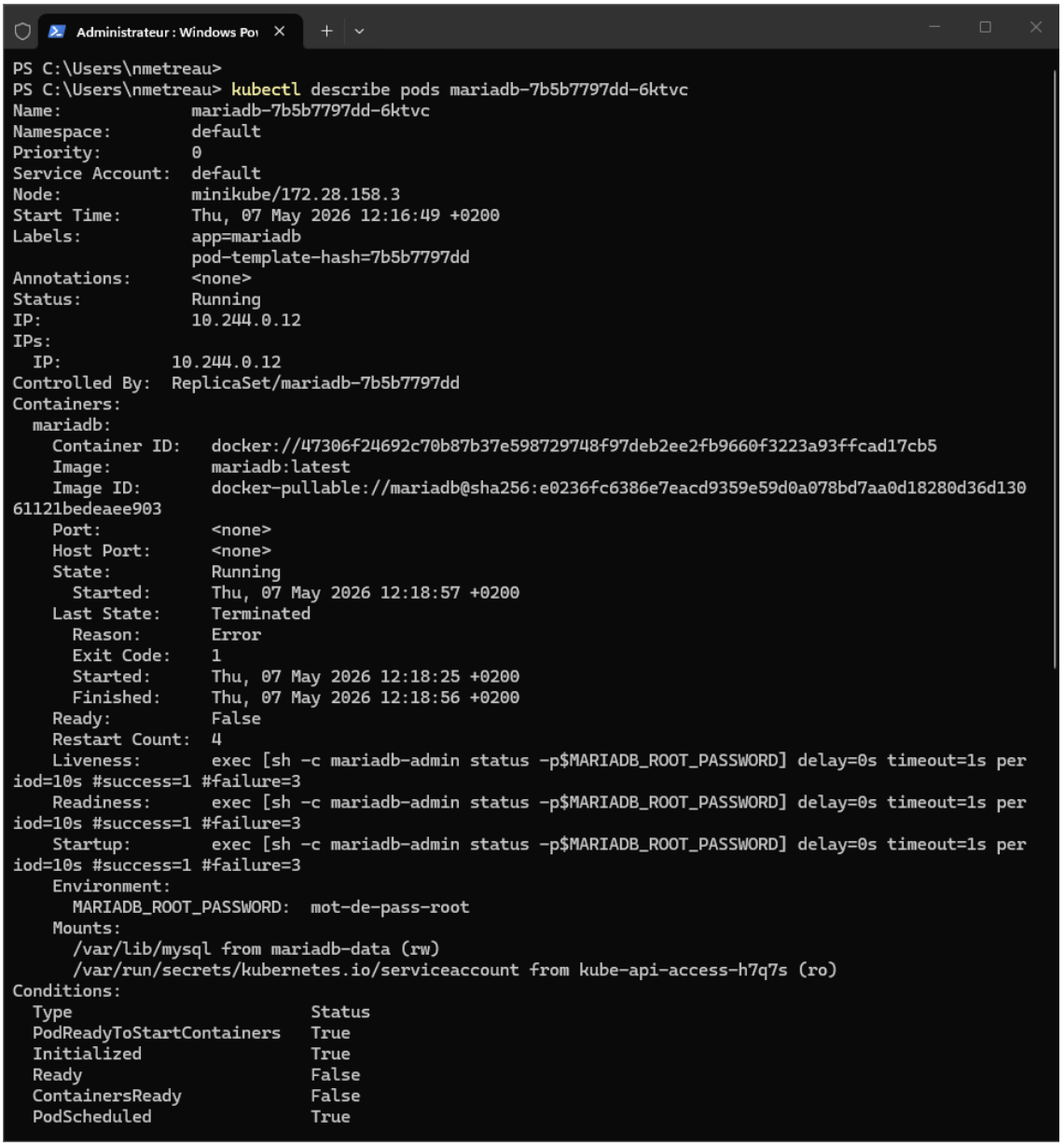
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS        powershell

PS E:\Projet Nana\9> kubectl get pod -l app=mariadb
NAME                       READY   STATUS    RESTARTS        AGE
mariadb-7b5b7797dd-fw6th   1/1     Running   7 (4m31s ago)   9m8s
PS E:\Projet Nana\9>
```

▪ Mise en place d'un StatefulSet :

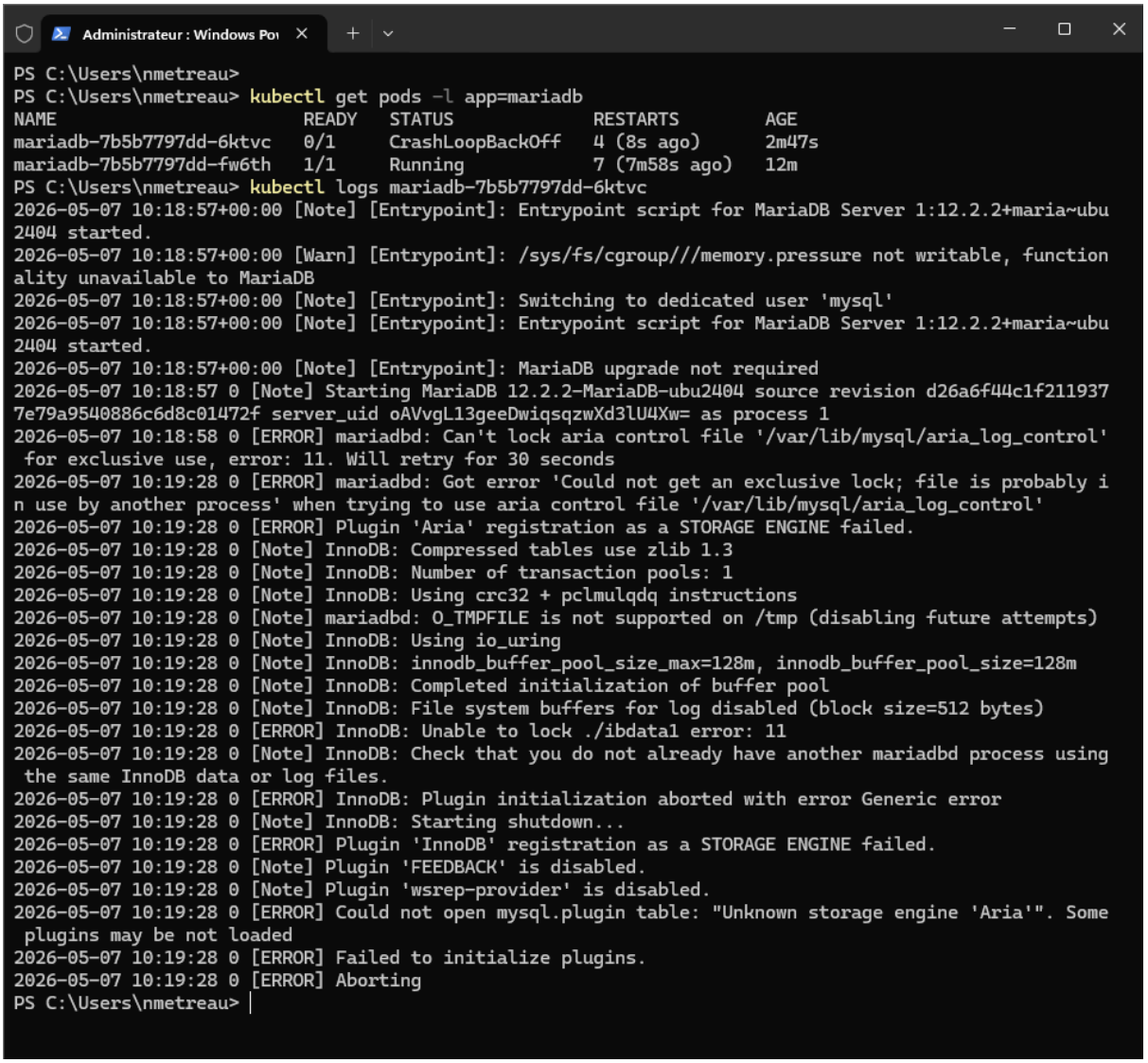
- Augmentation du nombre de pods associés au déploiement et montée en charge automatique



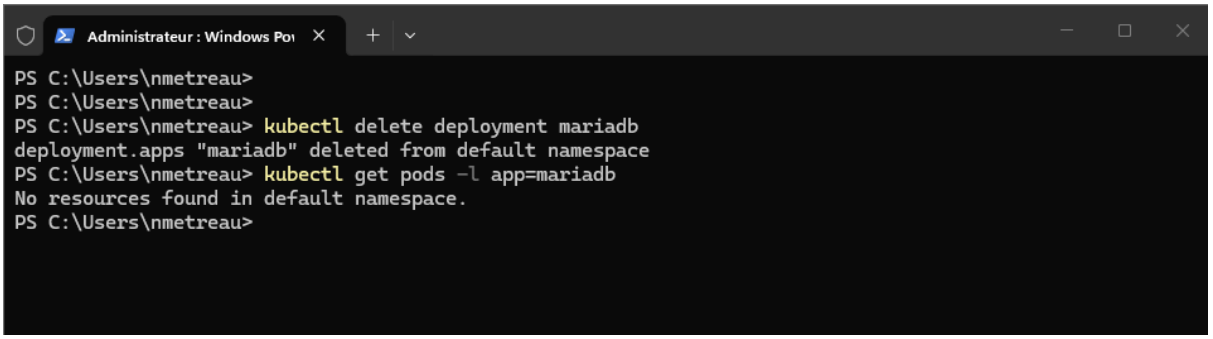
▪ Le nouveau pod n'arrive pas à démarrer.



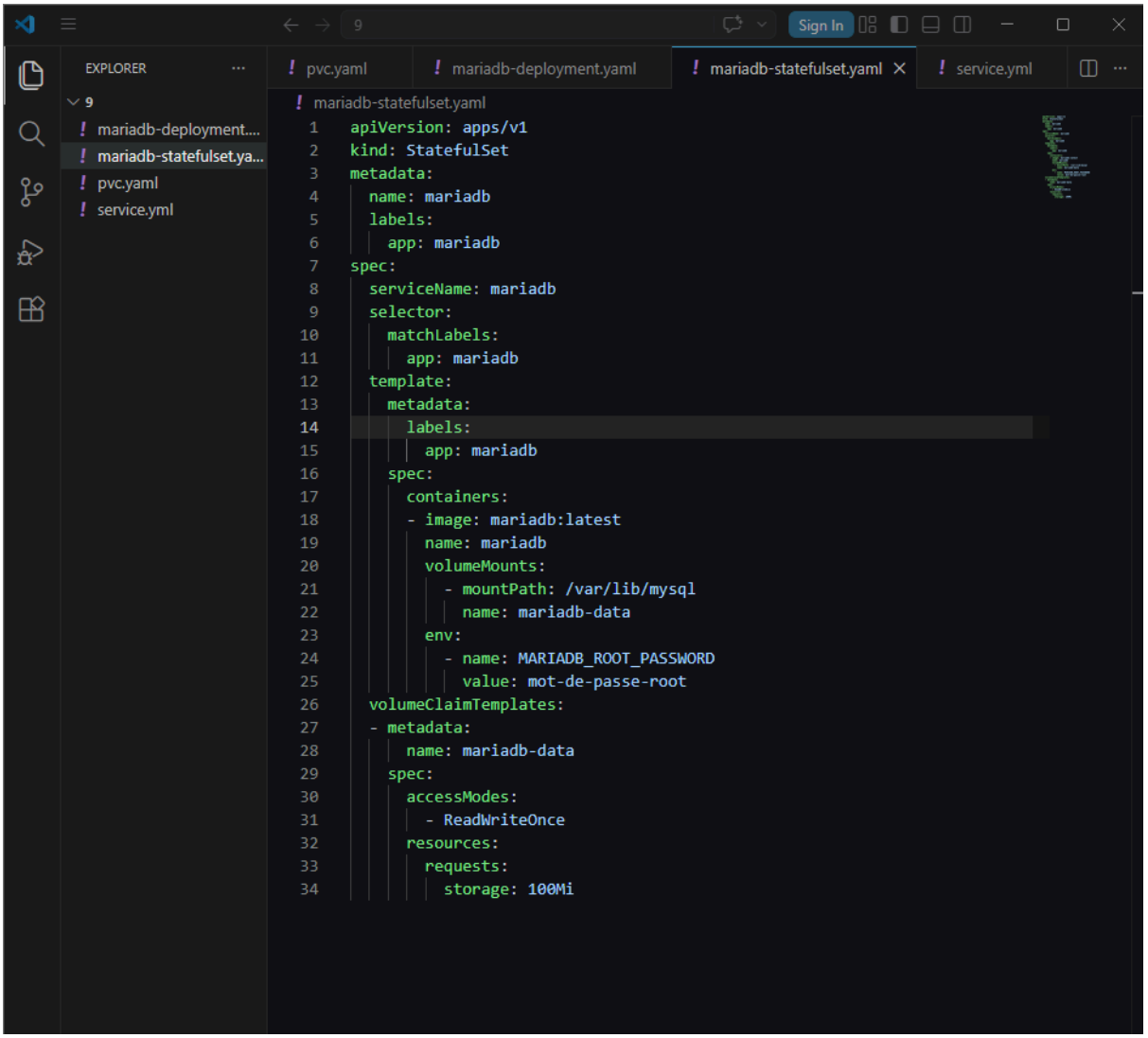
```
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       False
  ContainersReady             False
  PodScheduled                True
Volumes:
  mariadb-data:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  mariadb-data
    ReadOnly:   false
  kube-api-access-h7q7s:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    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  2m10s             default-scheduler  Successfully assigned default/mariadb-7b5b
7797dd-6ktvc to minikube
  Normal   Pulled     2m9s              kubelet            Successfully pulled image "mariadb:latest"
 in 866ms (866ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     97s               kubelet            Successfully pulled image "mariadb:latest"
 in 872ms (872ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     34s               kubelet            Successfully pulled image "mariadb:latest"
 in 847ms (847ms including waiting). Image size: 335568219 bytes.
  Warning  Unhealthy  10s (x12 over 2m)  kubelet            Startup probe failed: mariadb-admin: conne
ct to server at 'localhost' failed
error: 'Can't connect to local server through socket '/run/mysqld/mysqld.sock' (2)'
Check that mariadbd is running and that the socket: '/run/mysqld/mysqld.sock' exists!
  Normal  Killing  10s (x4 over 100s)  kubelet  Container mariadb failed startup probe, will be restar
ted
  Normal  Pulling  3s (x5 over 2m9s)   kubelet  Pulling image "mariadb:latest"
  Normal  Created  2s (x5 over 2m8s)   kubelet  Container created
  Normal  Started  2s (x5 over 2m8s)   kubelet  Container started
  Normal  Pulled   2s (x2 over 65s)    kubelet  Successfully pulled image "mariadb:latest" in 841ms (8
41ms including waiting). Image size: 335568219 bytes.
PS C:\Users\nmetreau>
```



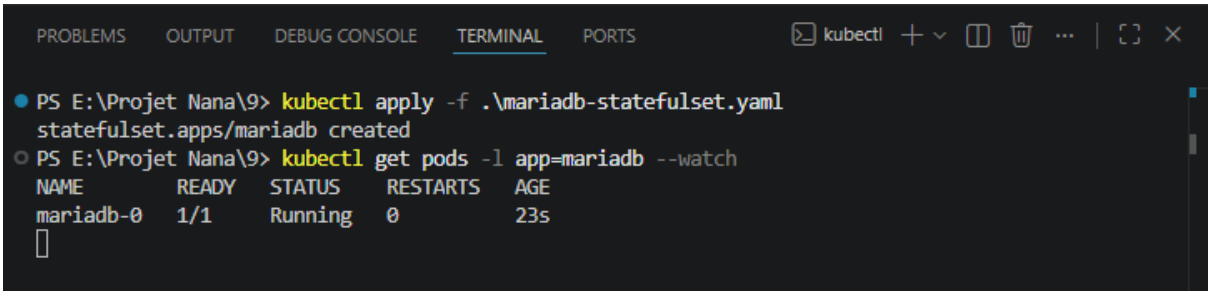
▪ Purge de l’ancien déploiement de MariaDB :



▪ Modification du fichier mariadb-deployment.yaml et sauvegarde sous le nom mariadbstatefulset.yaml :

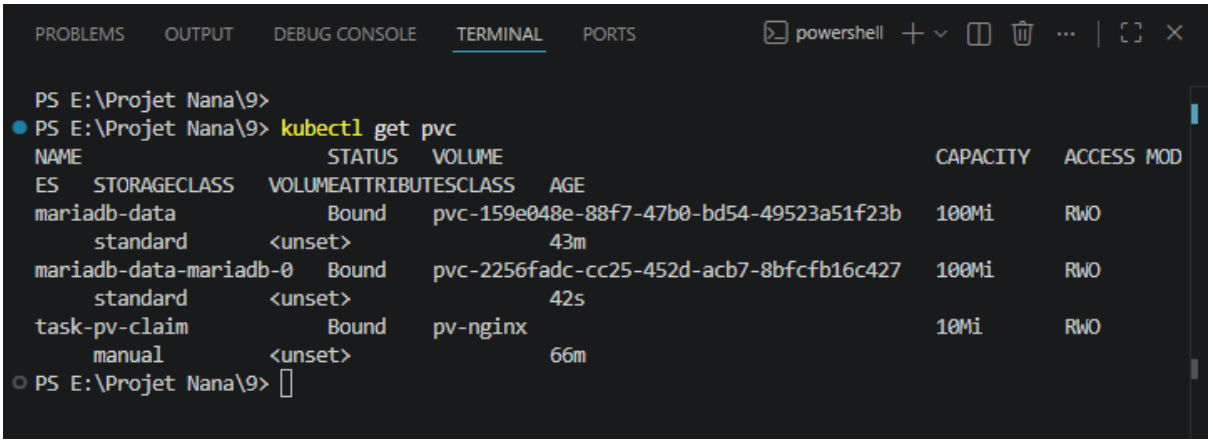


▪ Création du StatefulSet et consultation de l'état des pods associés au Statefulset de MariaDB :

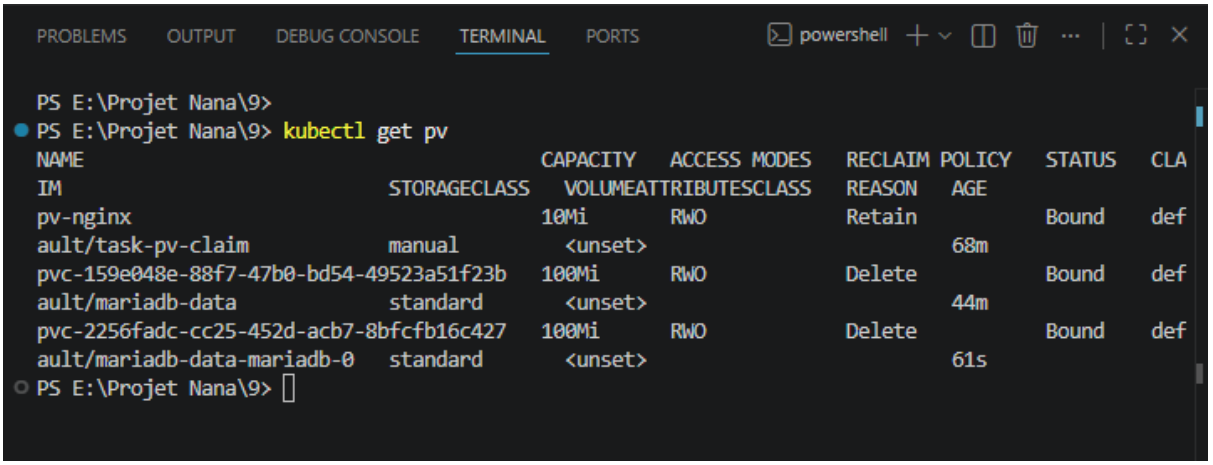


▪ La base est démarrée et prête à fonctionner.

- Consultation de l'état des demandes de volumes persistants :

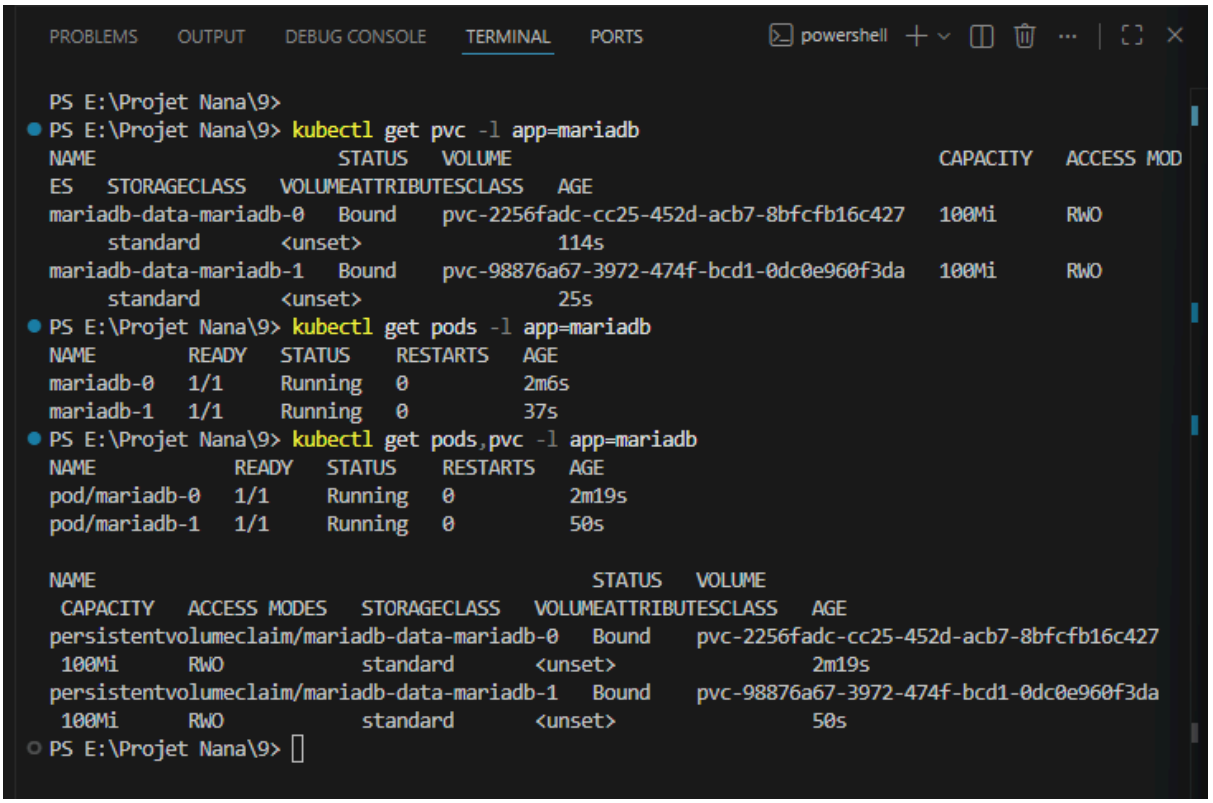


▪ Consultation de l'état des volumes persistants :

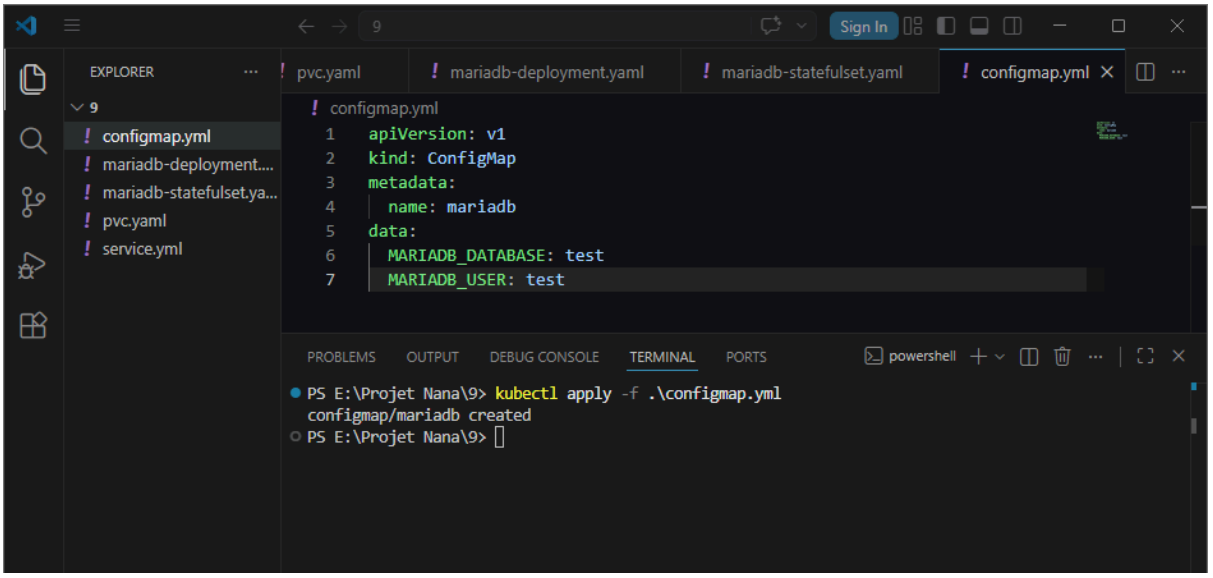


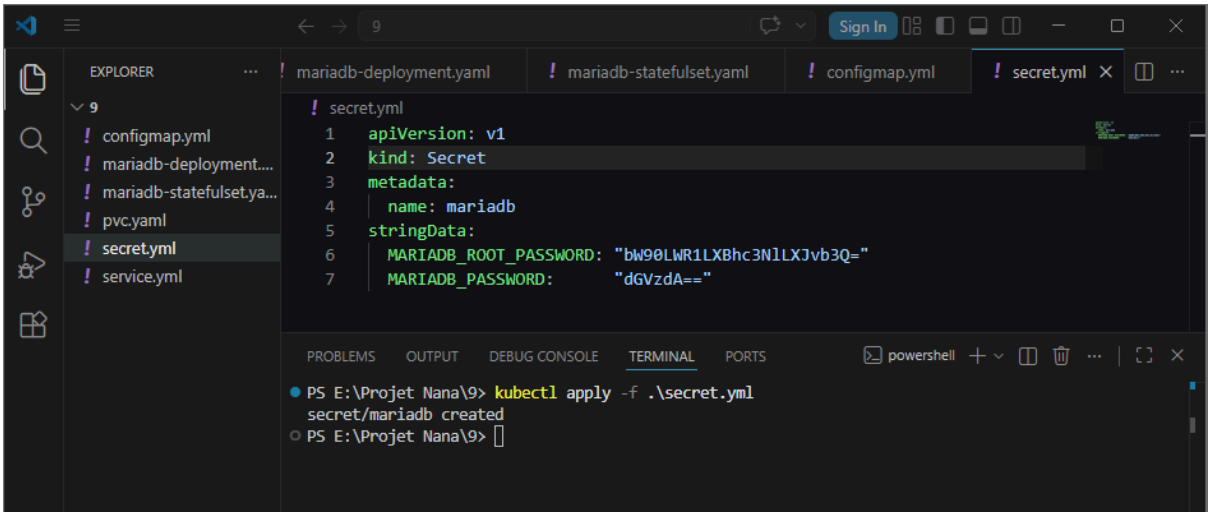
▪ Scalabilité de l'objet StatefulSet :

▪ Consultation des demandes de volumes persistants et des pods :



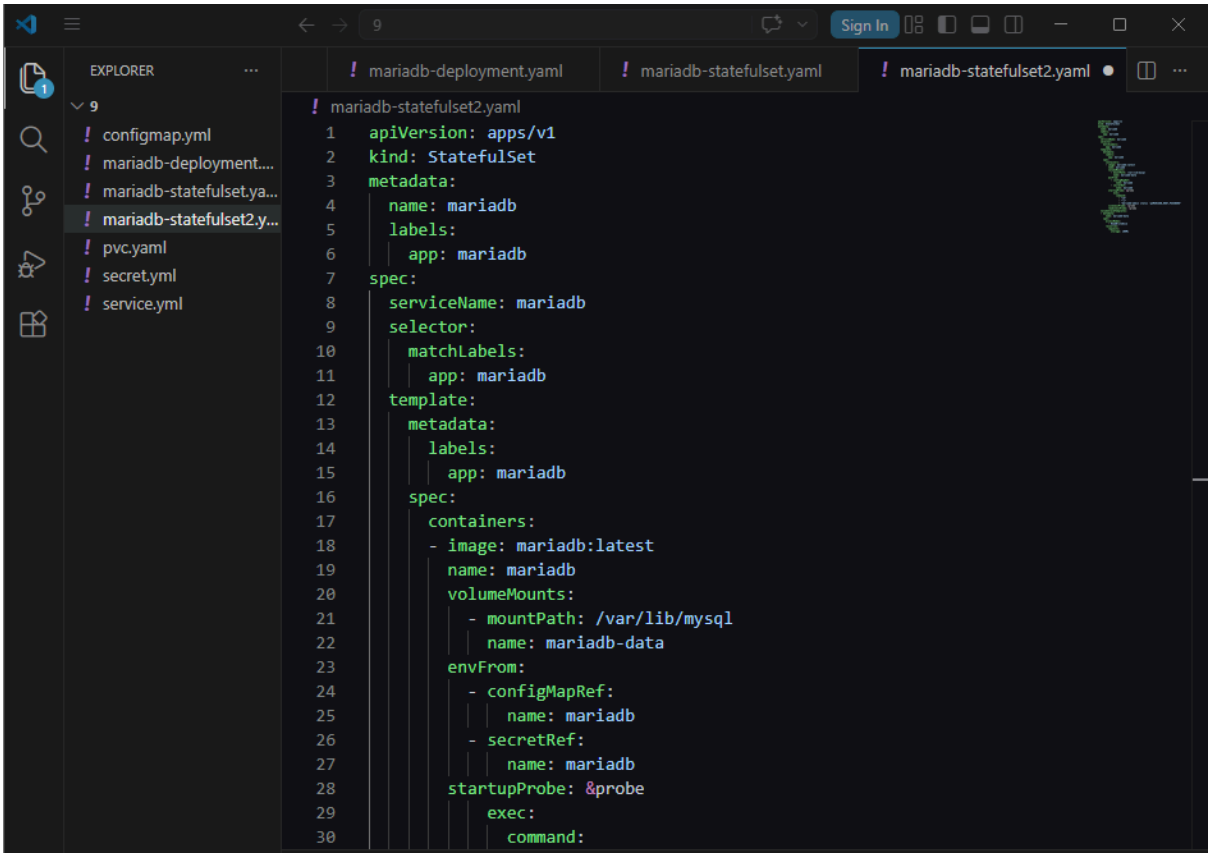
▪ Afin de tester la réplication au paragraphe 12, une base et un compte de test vont être créés.

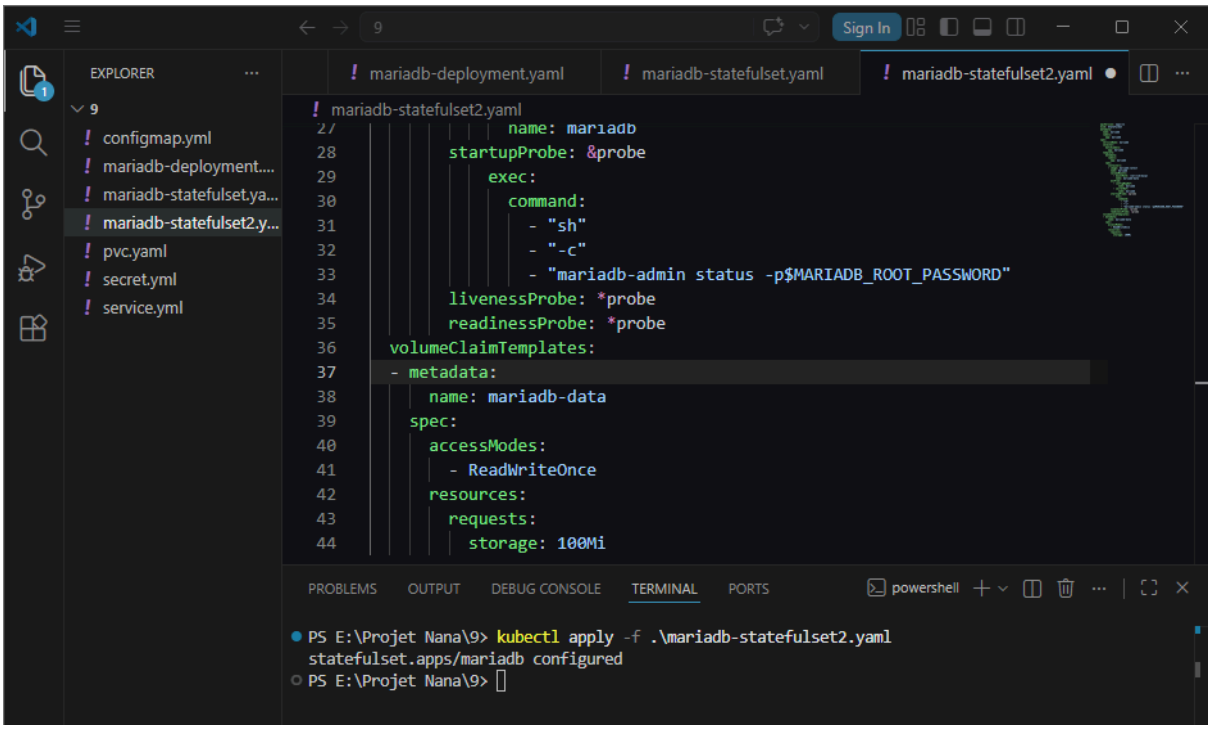




▪ Rattachement au conteneur :

- Le référencement se fait à l'aide du champ envFrom.





▪ La base est maintenant configurée.

12. Mises en place d’une réplication entre pods