

## BTS 2 - Services Informatiques aux Organisations

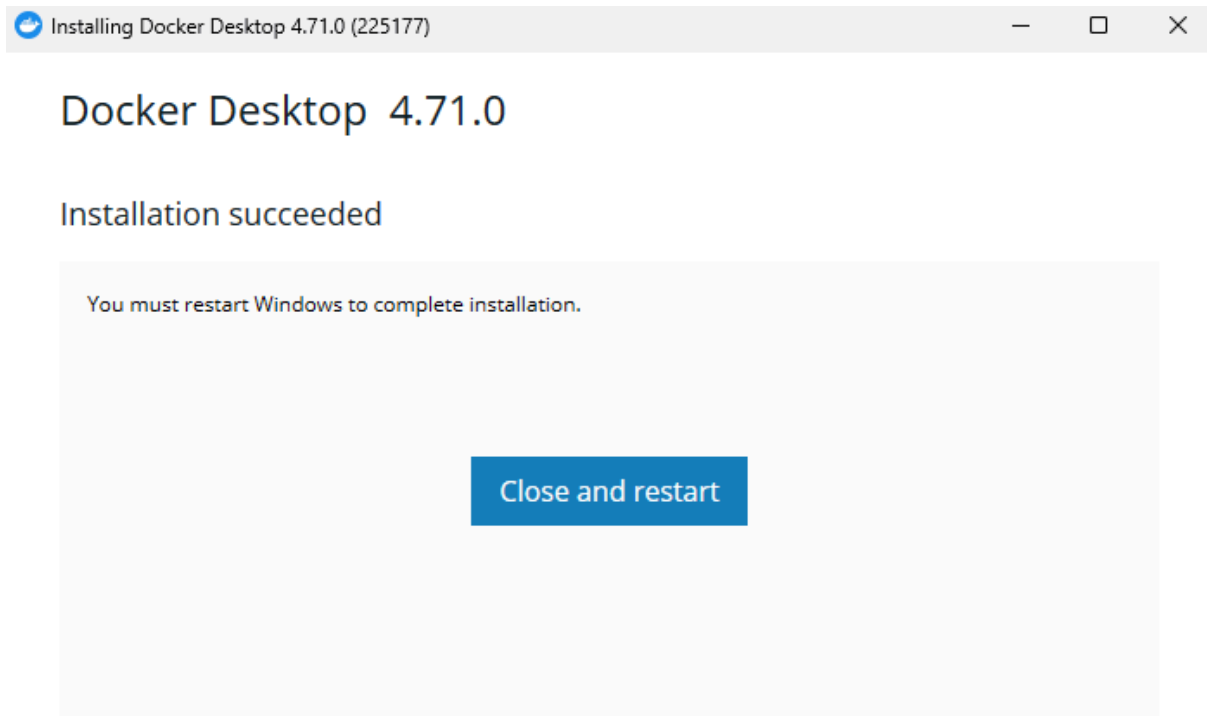


### BTS SIO SISR – Chapitre 5 – Kubernetes : plateforme de déploiement des applications conteneurisées

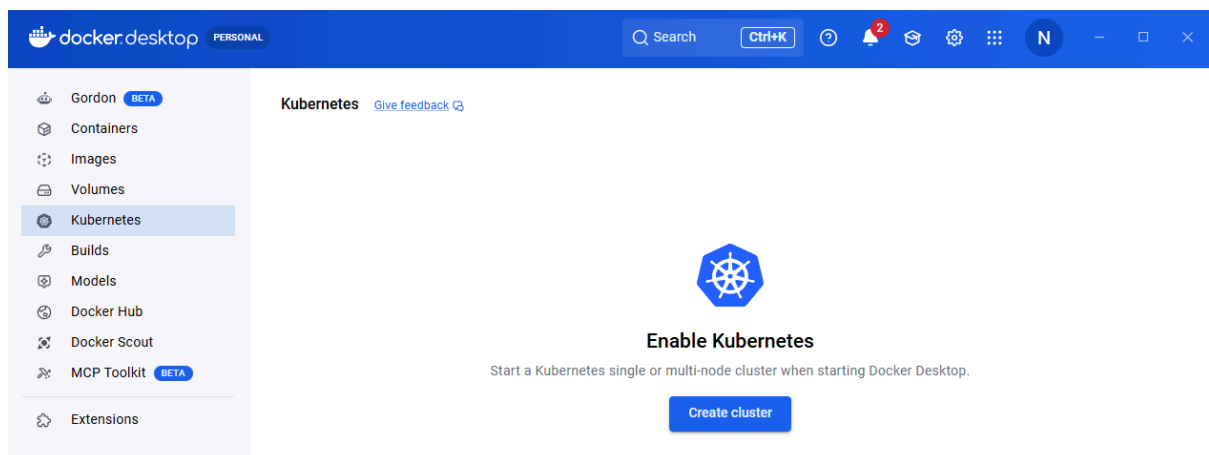
|   |    |
|---|----|
| 1. Kubernetes avec Docker Desktop.....                                      | 2  |
| 2. Installation de Minikube.....  | 7  |
| 3. Le tableau de bord de Kubernetes.....                                    | 15 |
| 4. Créer un déploiement à l'aide de l'outil kubectl.....                    | 25 |
| 5. Automatisation de déploiement par fichier YAML.....                      | 35 |
| 6. Application.....   | 45 |
| 7. Suite Ingress (Mailpit) : hôtes virtuels et nom de domaine nip.io.....   | 59 |
| 8. Hôtes virtuels et nom de domaine nip.io : autre exemple.....             | 61 |
| 9. Cycle de vie d'un conteneur dans Kubernetes.....                         | 63 |
| 10. Persistence des données.....  | 65 |
| 1) Volume de type emptyDir.....   | 65 |
| 2) Volume de type PV (PersistentVolume) et PVC (PersistentVolumeClaim)..... | 69 |
| 11. Hébergement d'un groupe de pods fonctionnant en cluster.....            | 75 |

## 1. Kubernetes avec Docker Desktop

Installation de Docker Desktop :



Activation de Kubernetes depuis Docker Desktop :



Création d'un cluster Kubernetes de type Kind avec 2 nodes :

### 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

**Kubeadm**  
Create a single-node cluster with kubeadm.  
Version: v1.34.1

#### Advanced Settings

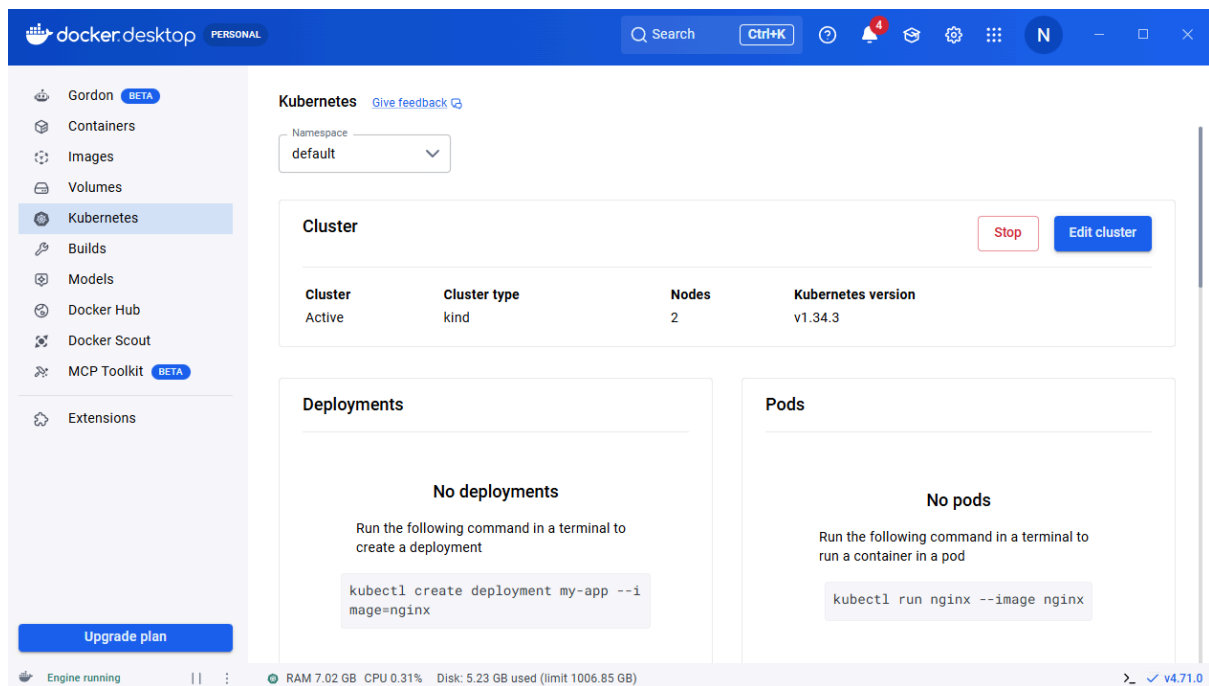
Show system containers (advanced)  
Show Kubernetes internal containers when using Docker commands.

Installation du cluster Kubernetes :

### Kubernetes Cluster Installation

Installation takes a few minutes and requires an internet connection.

Interface de Kubernetes dans Docker Desktop :



La commande **kubectl get nodes** permet de vérifier l'état des 3 nodes (noeuds) du cluster.

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

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> kubectl get nodes
NAME                    STATUS    ROLES    AGE   VERSION
desktop-control-plane   Ready    control-plane   2m10s   v1.34.3
desktop-worker          Ready    <none>        119s   v1.34.3
PS C:\WINDOWS\system32>
```

Configuration du cluster Kubernetes dans Docker Desktop (type, nœuds et version) :

### Modify Kubernetes Cluster



#### Cluster Type

kind  
Create a cluster containing one or more nodes with kind. Requires the [containerd image store](#)

Node(s): 3

Changing the number of nodes resets the cluster. All stacks and resources are deleted.



Version: 1.34.3

Changing the Kubernetes version resets your cluster. All stacks and resources are deleted.

Kubernetes version

Kubeadm  
Create a single-node cluster with kubeadm.  
Version: v1.34.1

#### Advanced Settings

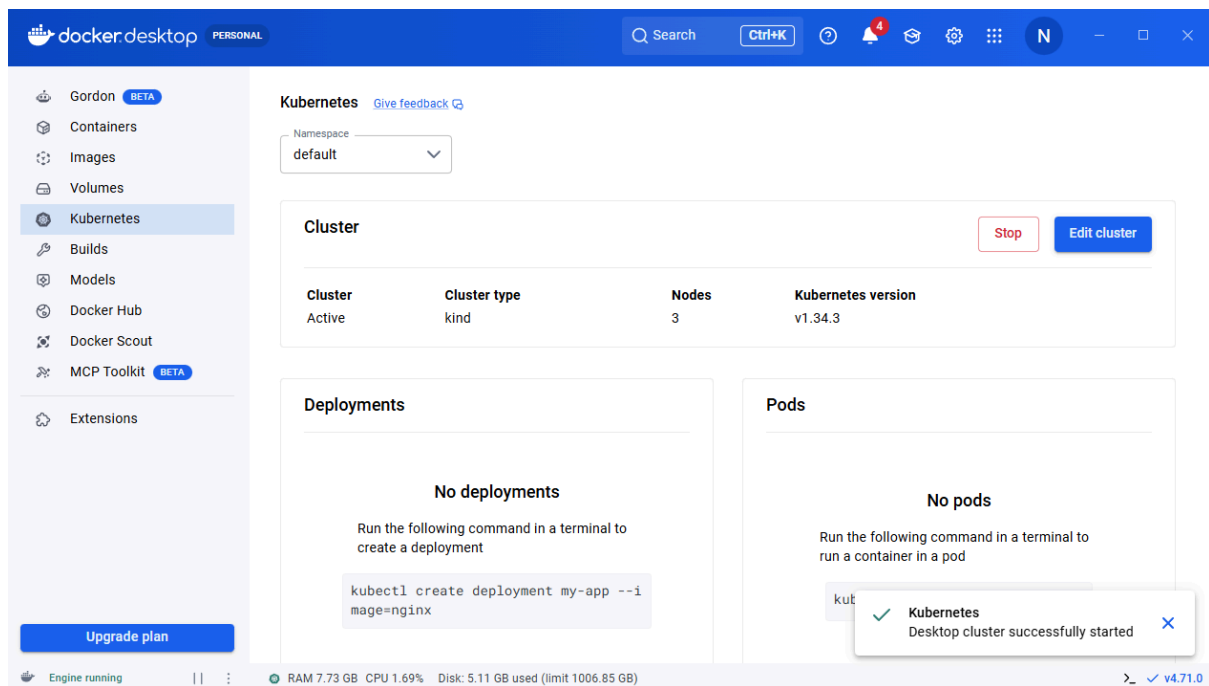
Show system containers (advanced)  
Show Kubernetes internal containers when using Docker commands.

Confirmation de modification du cluster Kubernetes :

### Change Kubernetes cluster configuration?

Changing the number of nodes deletes the current cluster and creates a new one.

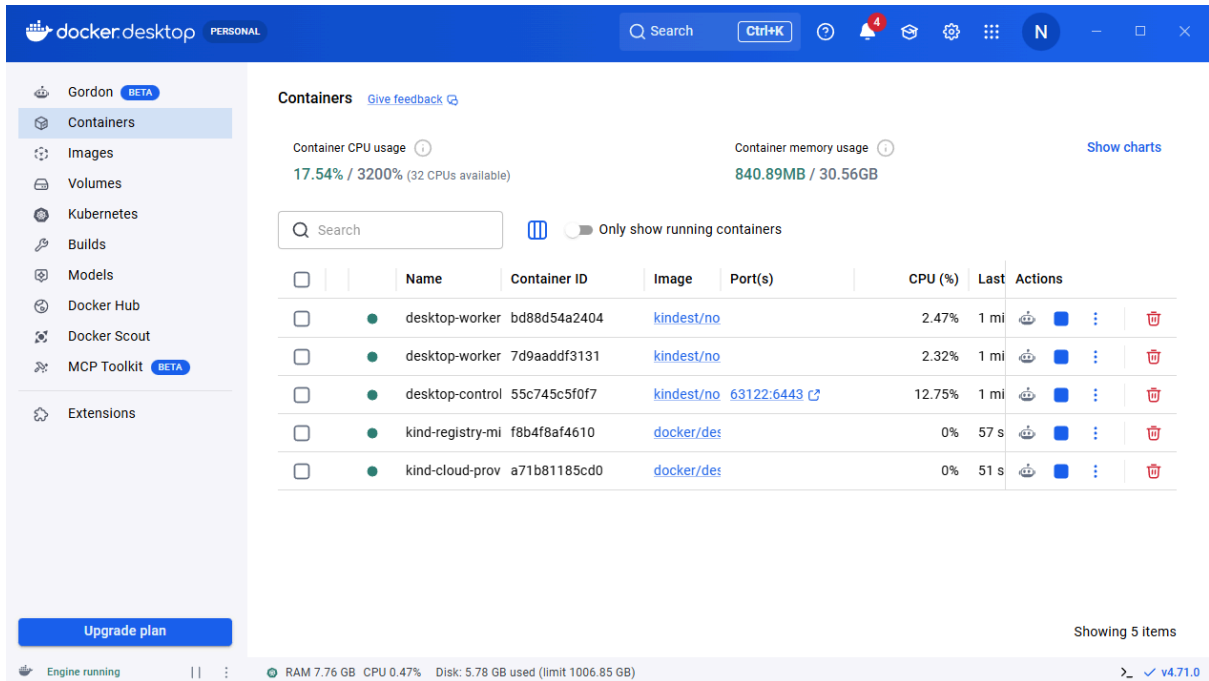
Démarrage du cluster Kubernetes dans Docker Desktop :



La commande `kubectl get nodes` permet de vérifier l'état des 3 nodes (noeuds) du cluster.

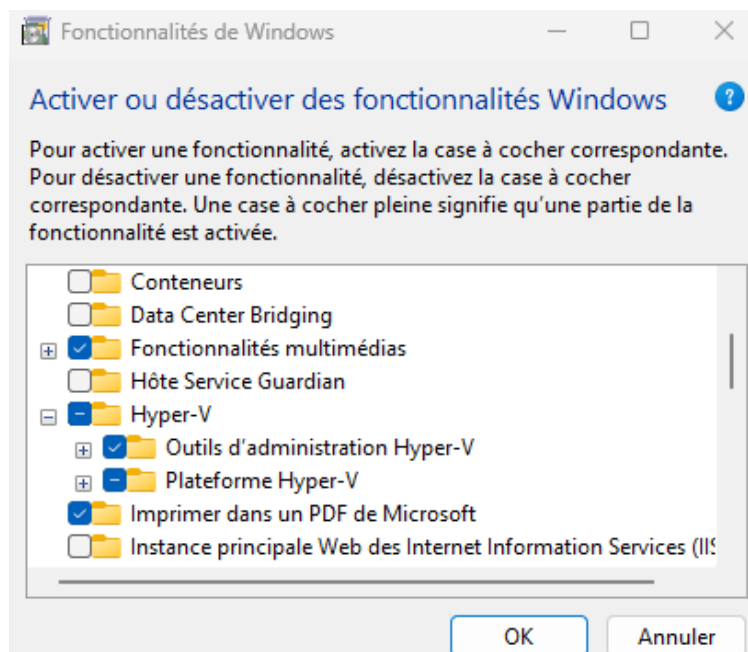
```
PS C:\WINDOWS\system32> kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
desktop-control-plane Ready    control-plane 56s   v1.34.3
desktop-worker      Ready    <none>    47s   v1.34.3
desktop-worker2     Ready    <none>    47s   v1.34.3
PS C:\WINDOWS\system32>
```

Visualisation de la liste des conteneurs Kubernetes en cours d'exécution dans Docker Desktop.

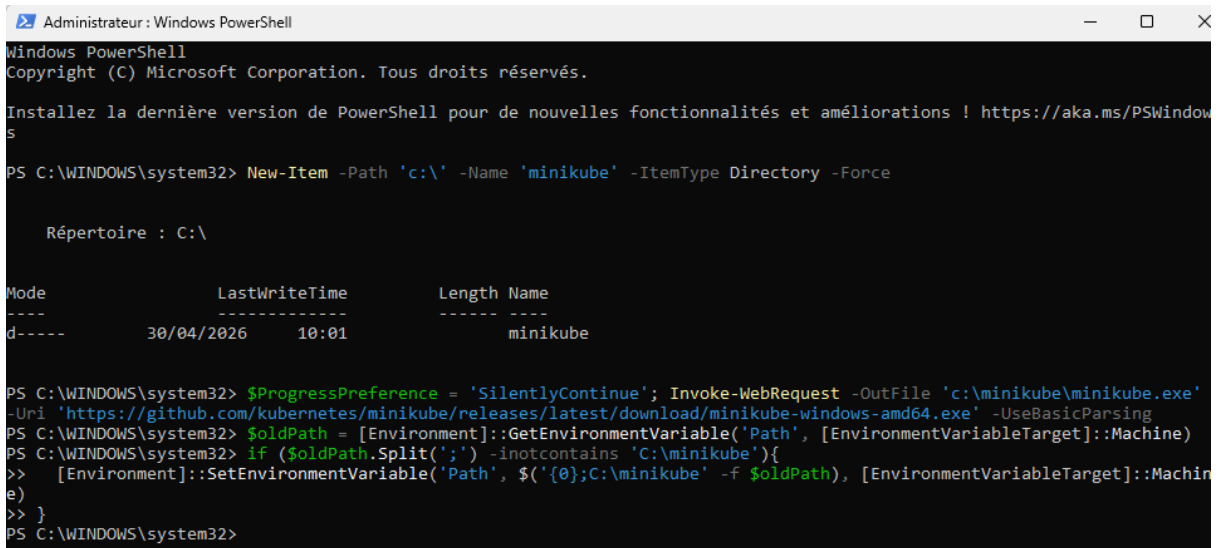


## 2. Installation de Minikube

Activation de Hyper-V permet d'utiliser des machines virtuelles pour exécuter Minikube.



L'installation de Minikube permet de créer un cluster Kubernetes local.



```

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

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows
PS C:\WINDOWS\system32> New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force

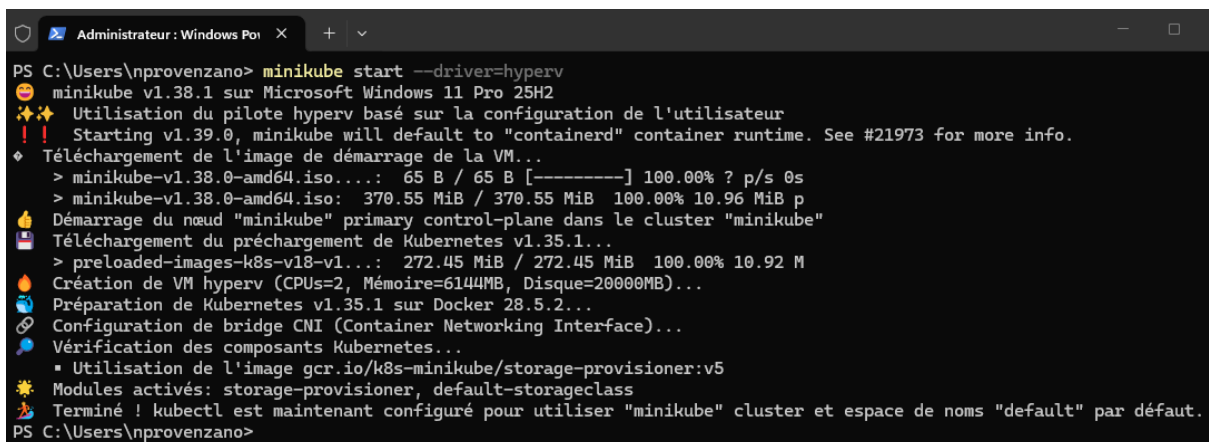
Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d-----          30/04/2026   10:01         minikube

PS C:\WINDOWS\system32> $ProgressPreference = 'SilentlyContinue'; Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe'
-Uri 'https://github.com/kubernetes/minikube/releases/latest/download/minikube-windows-amd64.exe' -UseBasicParsing
PS C:\WINDOWS\system32> $oldPath = [Environment]::GetEnvironmentVariable('Path', [EnvironmentVariableTarget]::Machine)
PS C:\WINDOWS\system32> if ($oldPath.Split(';') -notcontains 'C:\minikube'){
>> [Environment]::SetEnvironmentVariable('Path', $('{0};C:\minikube' -f $oldPath), [EnvironmentVariableTarget]::Machin
e)
>> }
PS C:\WINDOWS\system32>

```

La commande **minikube start** permet de démarrer le cluster Kubernetes local.

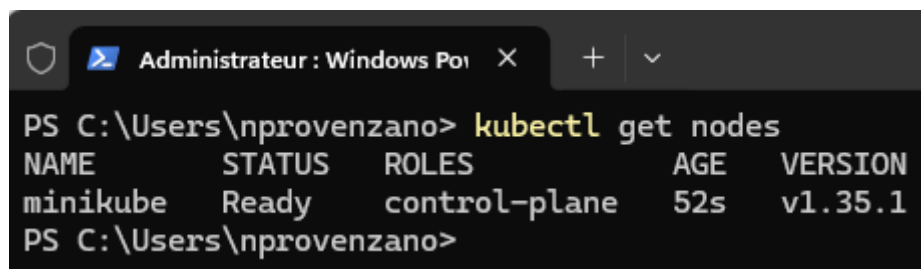


```

Administrateur : Windows Po
PS C:\Users\nprovenzano> minikube start --driver=hyperv
🐼 minikube v1.38.1 sur Microsoft Windows 11 Pro 25H2
🌟🌟 Utilisation du pilote hyperv basé sur la configuration de l'utilisateur
!! 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% 10.96 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% 10.92 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.
PS C:\Users\nprovenzano>

```

La commande **kubectl get nodes** permet de vérifier que le cluster est opérationnel.

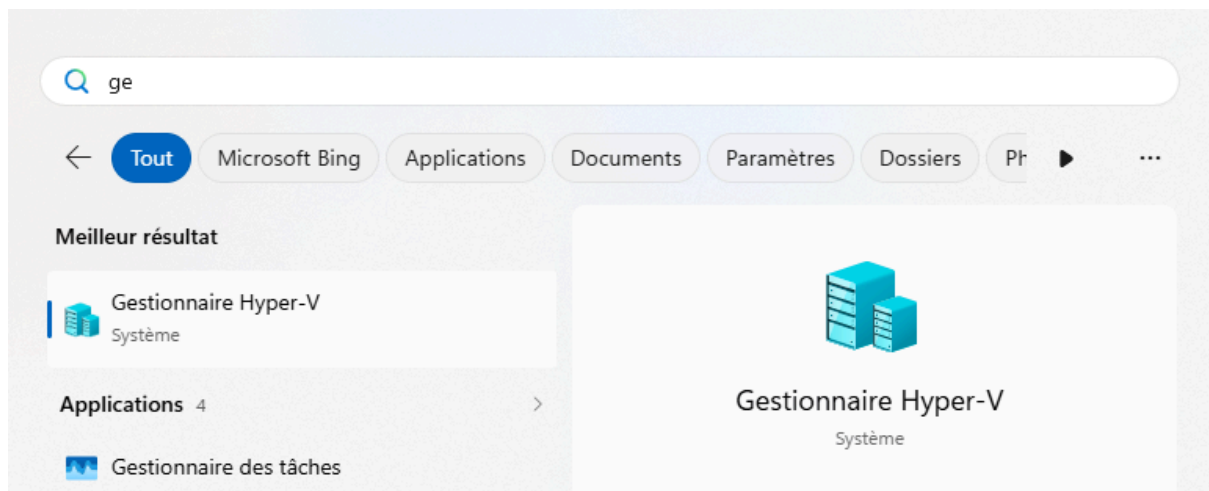


```

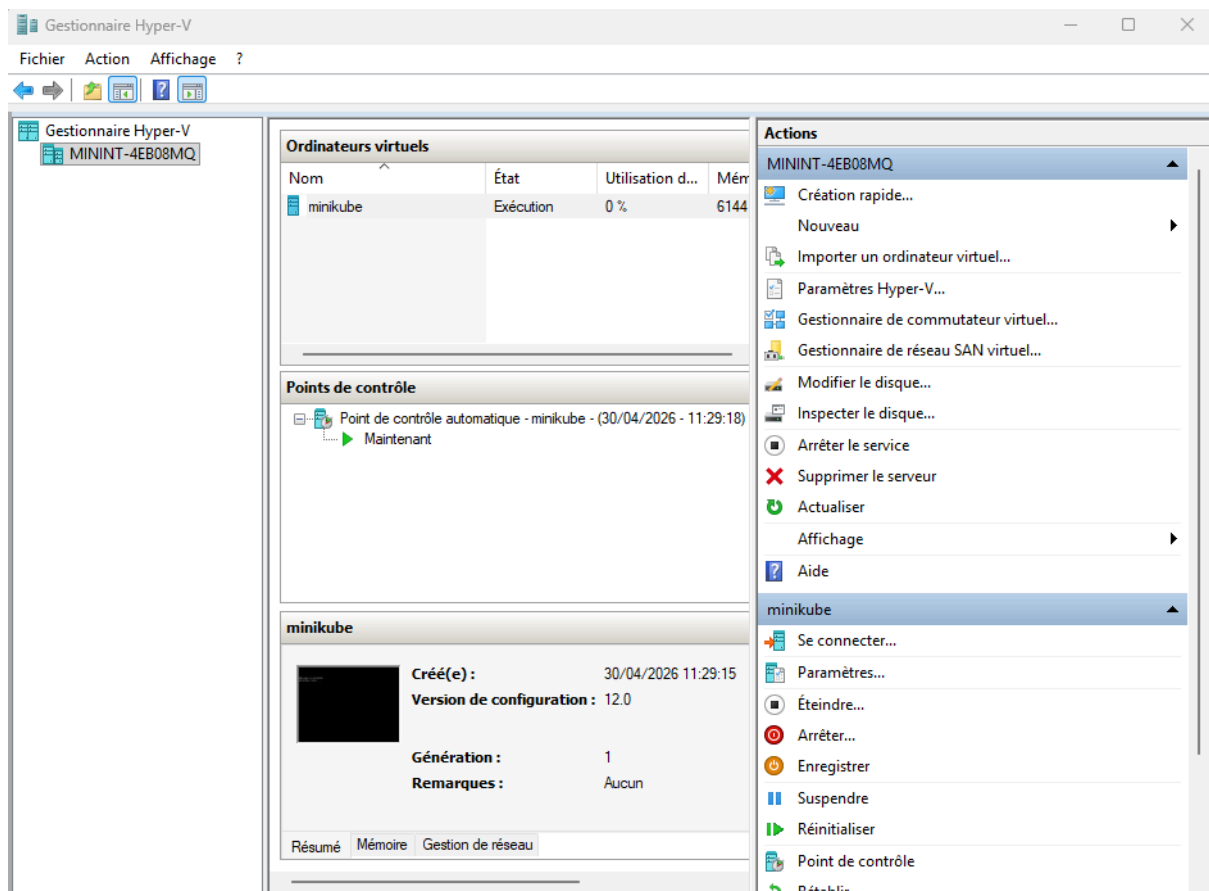
Administrateur : Windows Po
PS C:\Users\nprovenzano> kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
minikube      Ready    control-plane   52s   v1.35.1
PS C:\Users\nprovenzano>

```

Ouverture du Gestionnaire Hyper-V :



Le Gestionnaire Hyper-V permet de vérifier que la machine virtuelle Minikube est créée et en cours d'exécution.



La commande **kubectl get all** permet d'afficher tous les objets du cluster.

```

Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl get all
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes  ClusterIP     10.96.0.1     <none>         443/TCP    92s
PS C:\Users\nprovenzano>

```

La commande **kubectl run** permet de créer un pod à partir d'une image Docker.

La commande **kubectl get pods** permet de lister les pods en cours d'exécution.

```

Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl run monpod --image=nginx:latest
pod/monpod created
PS C:\Users\nprovenzano> kubectl get pods
NAME      READY   STATUS              RESTARTS   AGE
monpod    0/1     ContainerCreating   0           6s
PS C:\Users\nprovenzano> kubectl get all
NAME                READY   STATUS    RESTARTS   AGE
pod/monpod          1/1     Running   0           16s

NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes  ClusterIP     10.96.0.1     <none>         443/TCP    6m23s
PS C:\Users\nprovenzano>

```

La commande **kubectl logs** permet d'afficher les journaux d'un pod.

```

Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> 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 09:36:06 [notice] 1#1: using the "epoll" event method
2026/04/30 09:36:06 [notice] 1#1: nginx/1.29.8
2026/04/30 09:36:06 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2026/04/30 09:36:06 [notice] 1#1: OS: Linux 6.6.95
2026/04/30 09:36:06 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2026/04/30 09:36:06 [notice] 1#1: start worker processes
2026/04/30 09:36:06 [notice] 1#1: start worker process 29
2026/04/30 09:36:06 [notice] 1#1: start worker process 30
PS C:\Users\nprovenzano>

```

La commande **kubectl get namespace** permet d’afficher les espaces de noms.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl get namespace
NAME          STATUS    AGE
default       Active    7m38s
kube-node-lease Active    7m38s
kube-public   Active    7m38s
kube-system   Active    7m38s
PS C:\Users\nprovenzano>
```

La commande **kubectl get pods** permet de lister les pods et de vérifier leur état. Le pod “**monpod**” est en statut Running, le conteneur est en cours d’exécution ce qui indique que le déploiement est fonctionnel.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
monpod    1/1     Running   0           117s
PS C:\Users\nprovenzano> kubectl get pod
NAME      READY   STATUS    RESTARTS   AGE
monpod    1/1     Running   0           2m
PS C:\Users\nprovenzano> |
```

La commande **kubectl get nodes** permet de lister les nœuds du cluster et de vérifier leur état. Le résultat montre que le nœud “minikube” est en statut Ready avec le rôle control-plane.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl get nodes
NAME        STATUS    ROLES    AGE     VERSION
minikube    Ready    control-plane  8m36s  v1.35.1
PS C:\Users\nprovenzano> kubectl get node
NAME        STATUS    ROLES    AGE     VERSION
minikube    Ready    control-plane  8m39s  v1.35.1
PS C:\Users\nprovenzano>
```

La commande `kubectl get node -o wide` permet d’afficher des informations détaillées sur les nœuds du cluster. Le résultat montre que le nœud “minikube” est en statut Ready, avec son adresse IP interne, sa version Kubernetes et son système d’exploitation.

```

Administrateur : Windows Poi x + v
PS C:\Users\nprovenzano> kubectl get node -o wide
NAME          STATUS    ROLES    AGE   VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE           KERNEL-VERSION
CONTAINER-RUNTIME
minikube      Ready    control-plane   9m3s   v1.35.1   172.22.204.133 <none>        Buildroot 2025.02   6.6.95
docker://28.5.2
PS C:\Users\nprovenzano>

```

La commande `minikube ssh` permet d’accéder à la machine virtuelle Minikube.

```

Administrateur : Windows Poi x + v
PS C:\Users\nprovenzano> minikube ssh

  ____  _
 / ___|| | | |
| |___| |_| |
|___| \___|___|

$ sudo ctr namespace ls
NAME LABELS
moby
$ sudo ctr -n moby container ls
CONTAINER          IMAGE          RUNTIME
0275cc34ffd7f981093338a59de3b646caa93da97b0b4bbf3fd06faf40fe3b19 - io.containerd.runc.v2
0b3acbbde0fe895b1055f7e8c3253069d011247eea44abcbcb0c18f19fc131f9 - io.containerd.runc.v2
3a156f852d3dee10925a7bf23220cb4156d7487b2f2f4977c567986d21bd5bf0 - io.containerd.runc.v2
3b7026adbbf9fe76bcb4677dcb151ce876a13a6fa3e3b3180fbefd3e01524ec - io.containerd.runc.v2
556f769de87ce3ca5303d8cc632fc00f9b9554d679824f2f95ad60c54fff70d6 - io.containerd.runc.v2
58219ce6db5adf21ae1d8d1d4c37d782f0062fe5a8a442bdb87824b807db847a - io.containerd.runc.v2
65665a3f4109ac593167784b961b0925415cf26b0e66054e2ac3c2945a4c3e0c - io.containerd.runc.v2
7e6f04e2b6712fd6af4c6617c13c1e1dbebcfc0bab0eeca65cd86a7de64935cb - io.containerd.runc.v2
7fc4d148b2baff3e257e7a8569e0d9708be9cbffedc4cfcecc97fb86794d40bd - io.containerd.runc.v2
94ffa3b6b0c87b1eec1dff255350b2e2a7621e1f7f087296606da9f14227ebd5 - io.containerd.runc.v2
9914d5caaed85a447f808626c9afc7dff67fb5a53a04df651896c3af2958e140 - io.containerd.runc.v2
a3d28f797fec884af645332013e1c5cb65ed9503fa8e7047a7b6fd2940de01c - io.containerd.runc.v2
c18b63523484e3af62ebb3259a6be5c4f9d7602abe2e769a980678ed714da54e - io.containerd.runc.v2
d0c1c94b453f8a9118f99e69e9ad9c8f8f522c6bb15aa1a9b8def92604ba60e5 - io.containerd.runc.v2
d6c64b020a677858e99ca5a4d530ab591eab4af7ac133ae27518b4c50a034882 - io.containerd.runc.v2
d7a337c131da8fd6a0733c376047c93ca41eaf6c9f65ae365c73484b96f98972 - io.containerd.runc.v2
$ exit
logout
PS C:\Users\nprovenzano>

```

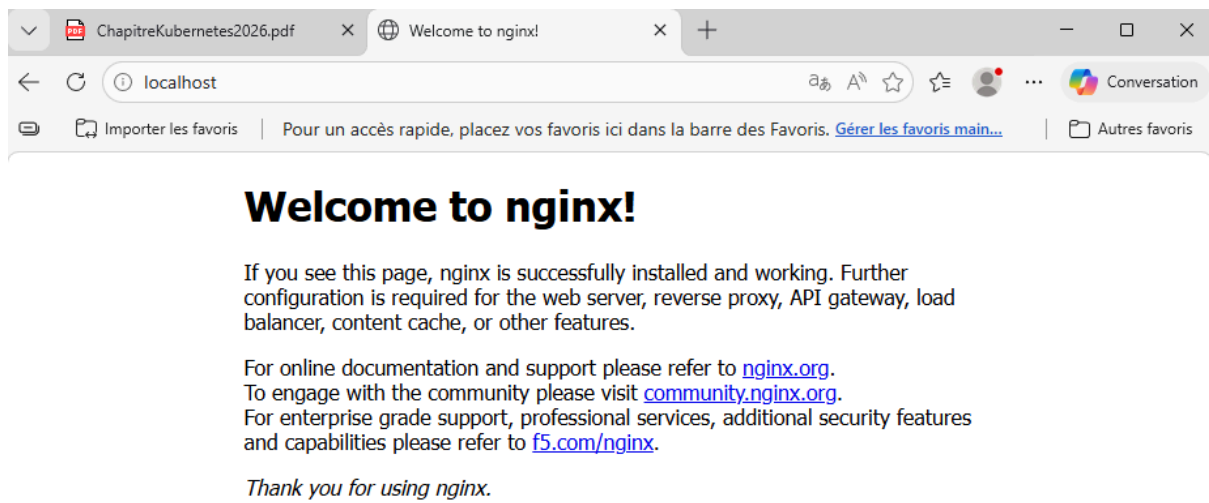
La commande **kubectl describe pod monpod** permet d'obtenir les détails d'un pod, sur son statut, sa configuration, ses conteneurs et les événements supplémentaires.

```
Administrateur : Windows Poi x + v
PS C:\Users\nprovenzano> kubectl describe pod/monpod
Name:          monpod
Namespace:    default
Priority:      0
Service Account: default
Node:         minikube/172.22.204.133
Start Time:   Thu, 30 Apr 2026 11:35:56 +0200
Labels:       run=monpod
Annotations:  <none>
Status:       Running
IP:           10.244.0.4
IPs:
IP: 10.244.0.4
Containers:
  monpod:
    Container ID:  docker://7fc4d148b2baff3e257e7a8569e0d9708be9cbffedc4cfcecc97fb86794d40bd
    Image:         nginx:latest
    Image ID:      docker-pullable://nginx@sha256:6e23479198b998e5e25921dff8455837c7636a67111a04a635cf1bb363d199dc
    Port:          <none>
    Host Port:     <none>
    State:         Running
      Started:     Thu, 30 Apr 2026 11:36:06 +0200
    Ready:         True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-l6wrk (ro)
Conditions:
  Type                               Status
  PodReadyToStartContainers          True
  Initialized                         True
  Ready                               True
  ContainersReady                    True
  PodScheduled                       True
Volumes:
  kube-api-access-l6wrk:
    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
```

La commande **kubectl port-forward monpod 80** permet de faire un tunnel entre le pod et la machine locale pour accéder à un service interne du cluster via le port 80.

```
Administrateur : Windows Poi x + v
PS C:\Users\nprovenzano> kubectl port-forward monpod 80
Forwarding from 127.0.0.1:80 -> 80
Forwarding from [::1]:80 -> 80
Handling connection for 80
Handling connection for 80
```

L'accès à localhost permet de vérifier que le pod Nginx est bien déployé et que le service est accessible depuis la machine locale.



La commande **kubectl delete pod monpod** permet de supprimer un pod “monpod” du cluster Kubernetes.

```
Windows PowerShell
PS C:\Users\nprovenzano> kubectl delete pod/monpod
pod "monpod" deleted from default namespace
PS C:\Users\nprovenzano> kubectl get pod
No resources found in default namespace.
PS C:\Users\nprovenzano> |
```

### 3. Le tableau de bord de Kubernetes

La commande **minikube addons enable dashboard** permet d'activer l'interface graphique de Kubernetes pour visualiser et gérer les ressources du cluster.

La commande **minikube addons enable metrics-server** permet d'activer le service qui collecte les métriques des ressources (CPU, mémoire) des pods et des nœuds.

```

Administrateur : Windows Pow x Administrateur : Windows Poi x + v
PS C:\Users\nprovenzano> minikube addons enable dashboard
  dashboard est un addon maintenu par Kubernetes. Pour toute question, contactez minikube sur GitHub.
  Vous pouvez consulter la liste des mainteneurs de minikube sur : https://github.com/kubernetes/minikube/blob/master/OWNER
  S
  ▪ Utilisation de l'image docker.io/kubernetesui/dashboard:v2.7.0
  ▪ Utilisation de l'image docker.io/kubernetesui/metrics-scraper:v1.0.8
  Certaines fonctionnalités du tableau de bord nécessitent le module complémentaire metrics-server. Pour activer toutes
  les fonctionnalités, veuillez exécuter :

  minikube addons enable metrics-server

  ✨ Le module 'dashboard' est activé
PS C:\Users\nprovenzano> minikube addons enable metrics-server
  ✨ metrics-server est un addon maintenu par Kubernetes. Pour toute question, contactez minikube sur GitHub.
  Vous pouvez consulter la liste des mainteneurs de minikube sur : https://github.com/kubernetes/minikube/blob/master/OWNER
  S
  ▪ Utilisation de l'image registry.k8s.io/metrics-server/metrics-server:v0.8.1
  ✨ Le module 'metrics-server' est activé
PS C:\Users\nprovenzano>

```

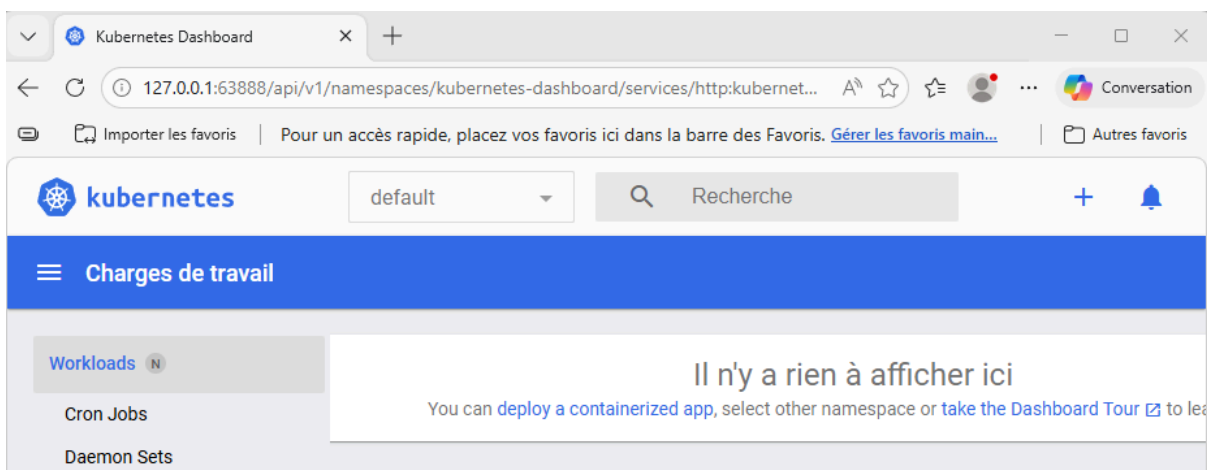
La commande **minikube dashboard** permet de lancer l'interface graphique de Kubernetes.

```

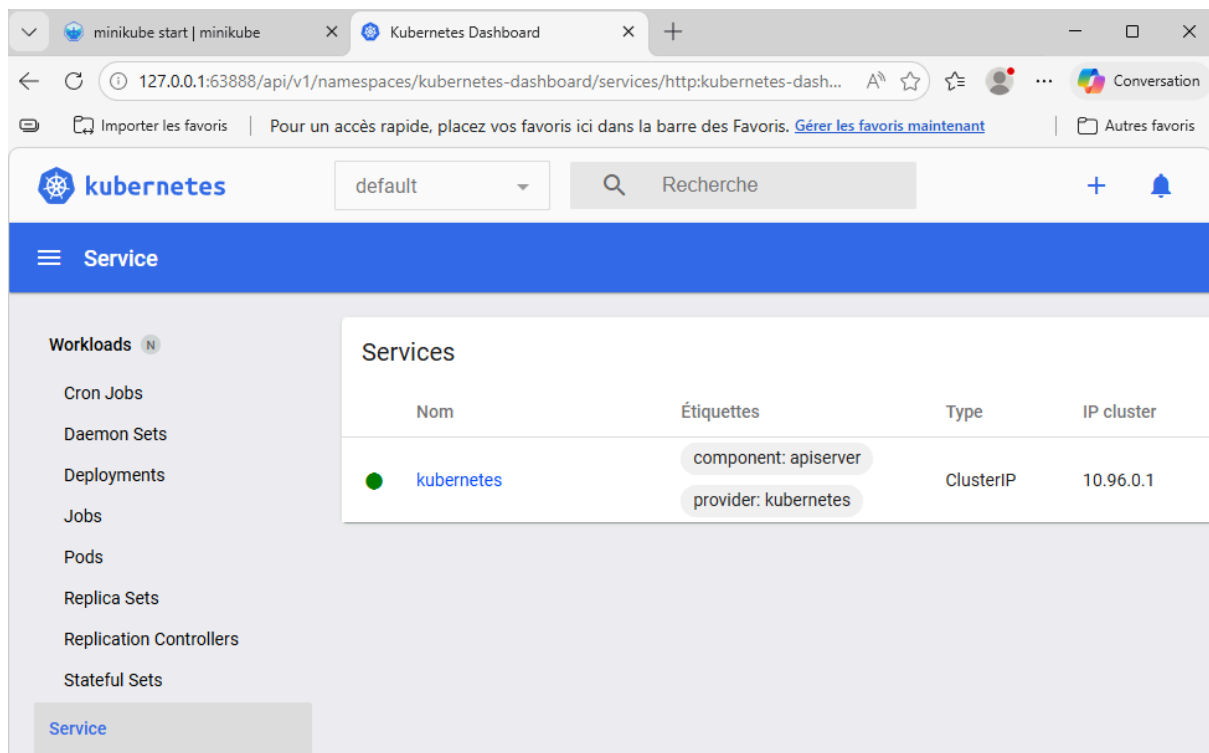
Windows PowerShell x + v
PS C:\Users\nprovenzano> minikube dashboard
  🤖 Vérification de l'état du tableau de bord...
  🚀 Lancement du proxy...
  🤖 Vérification de l'état du proxy...
  🌐 Ouverture de http://127.0.0.1:63888/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy
  / dans votre navigateur par défaut...

```

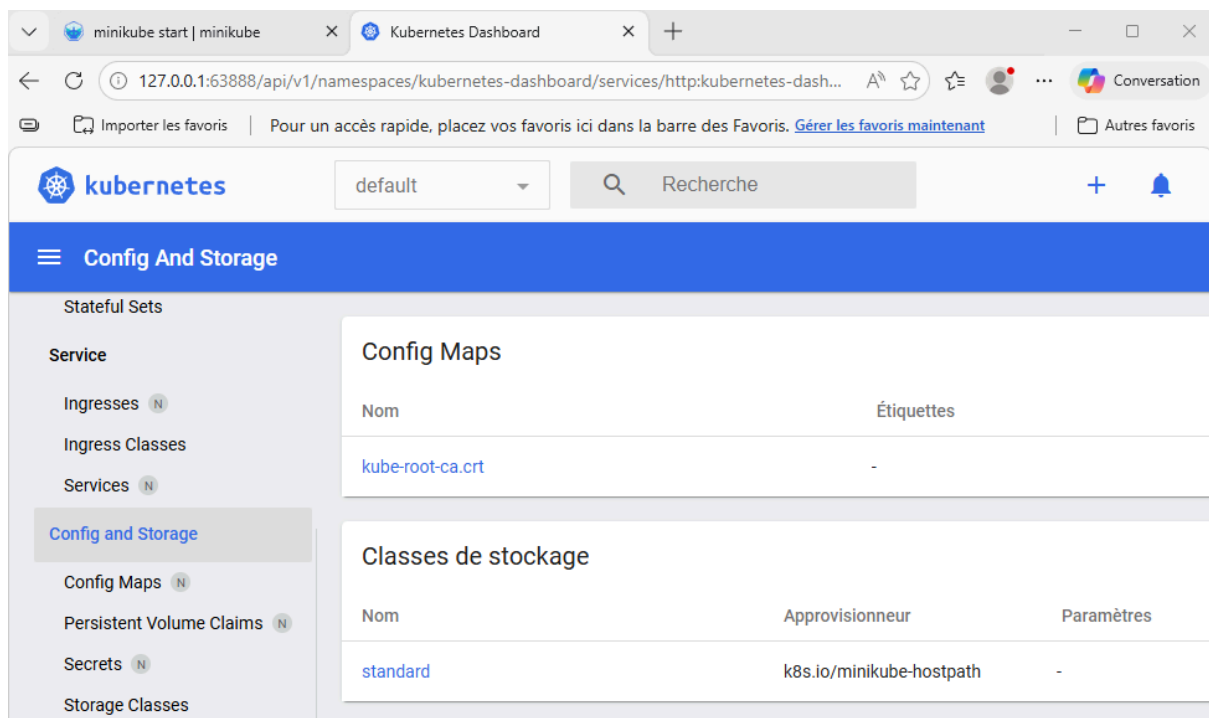
Visualisation du tableau de bord du cluster Kubernetes :



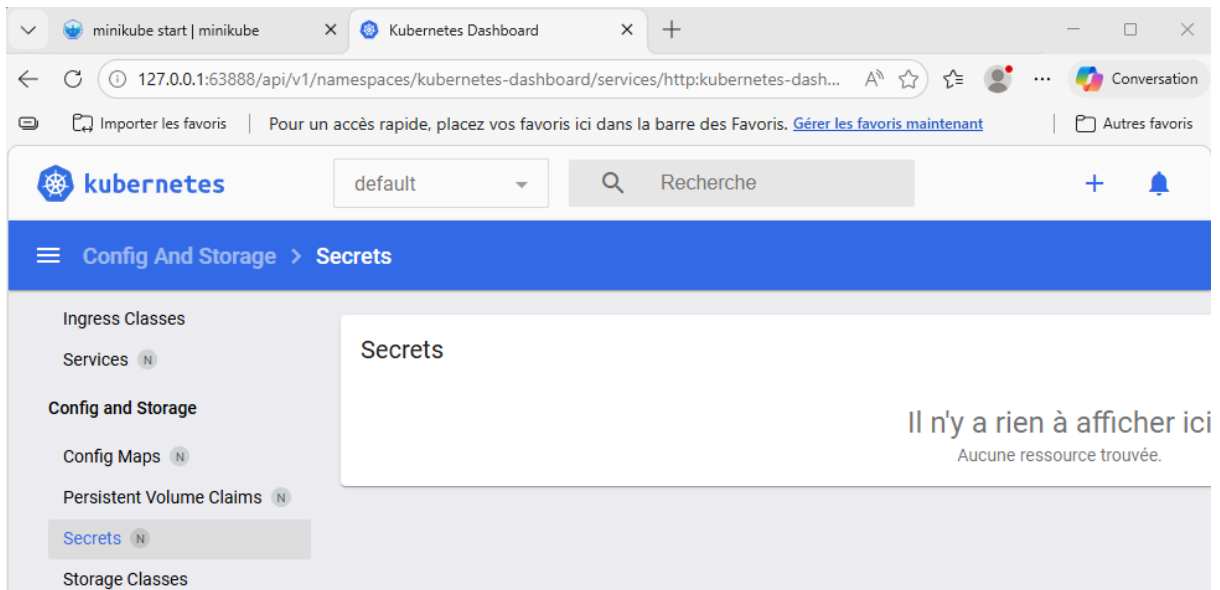
Affichage des Services du cluster :



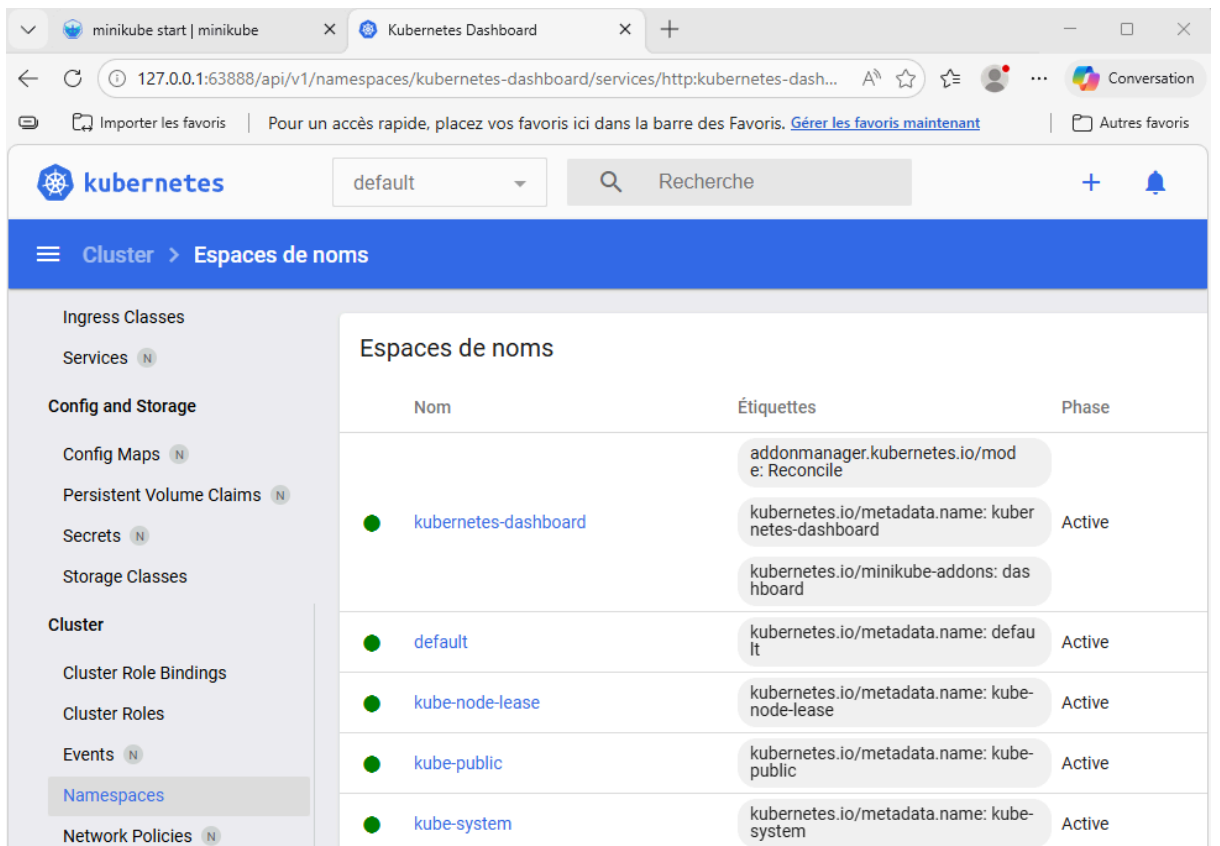
Affichage des ConfigMaps et du stockage :



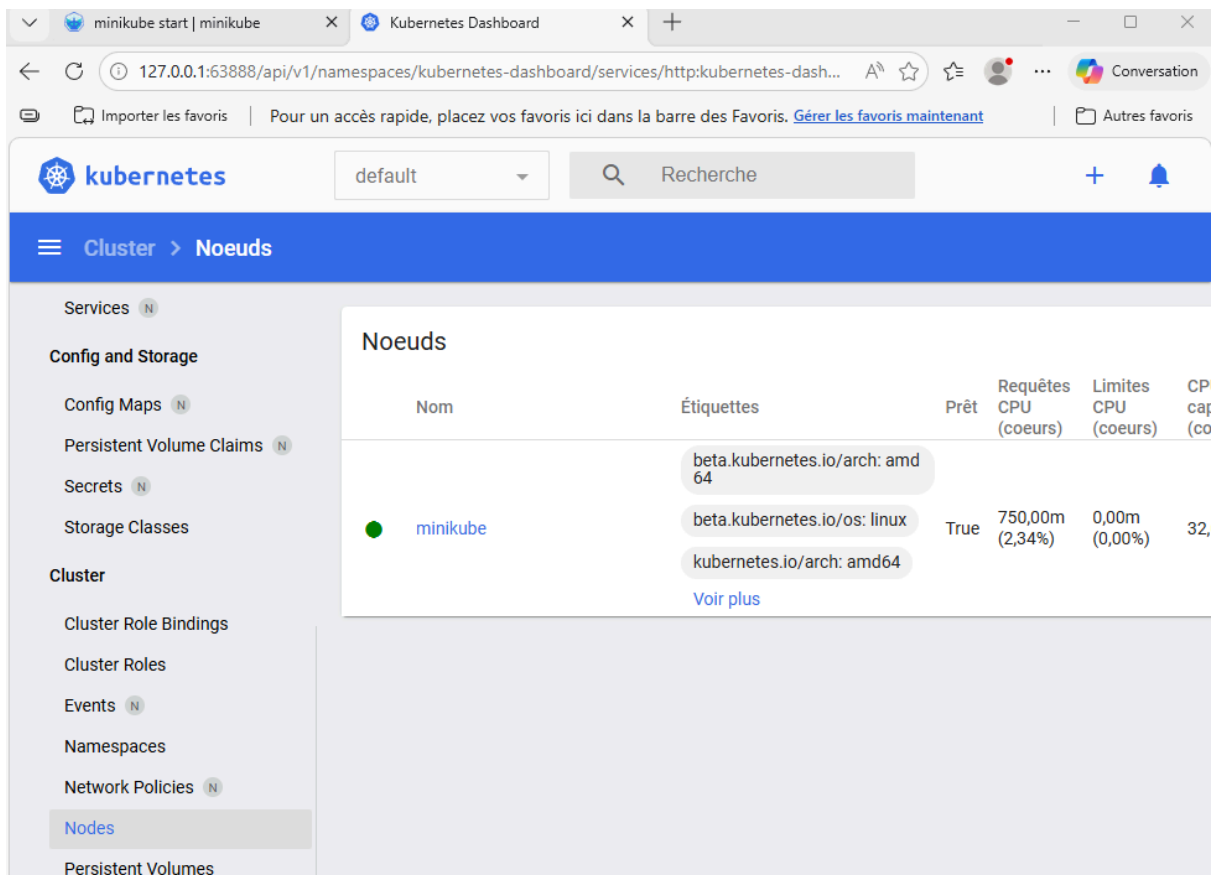
Affichage des Secrets :



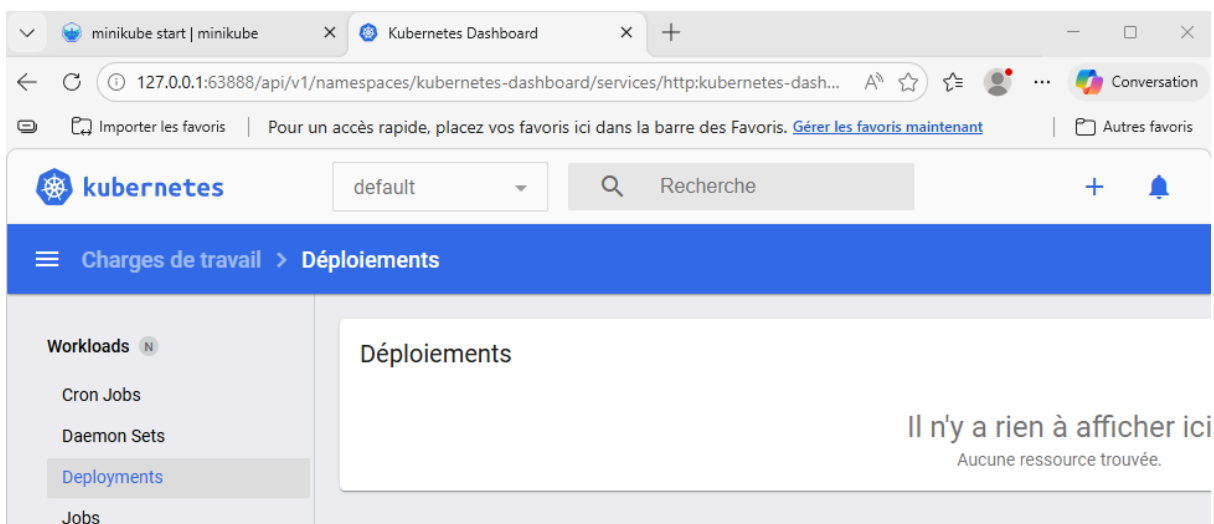
Affichage des Namespaces :



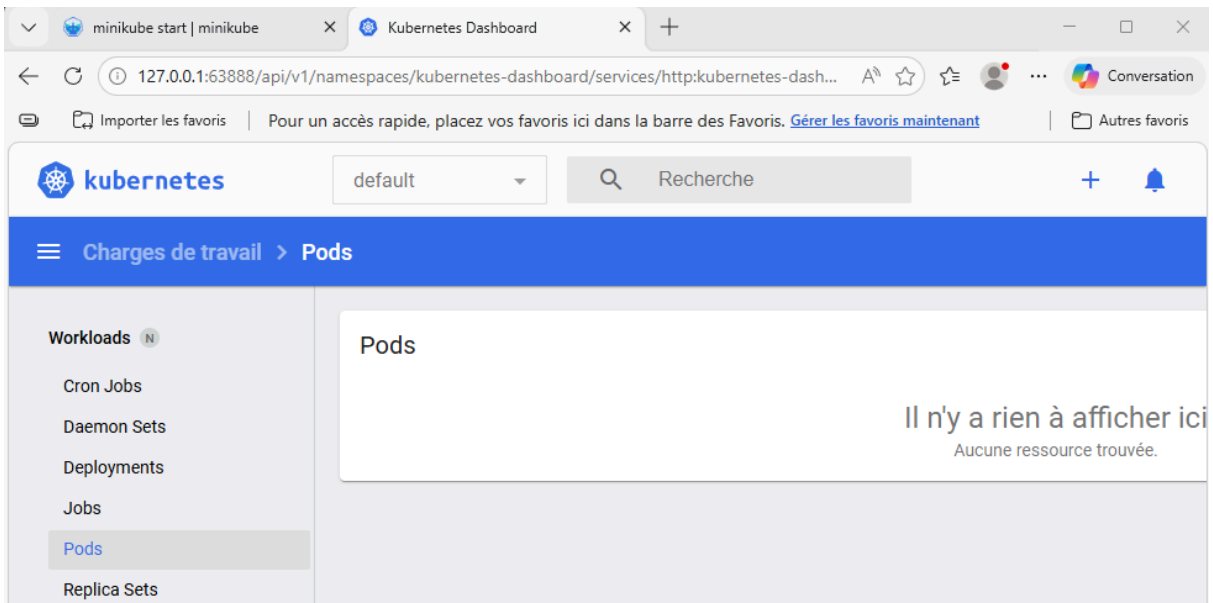
Affichage des Nodes (noeuds) :



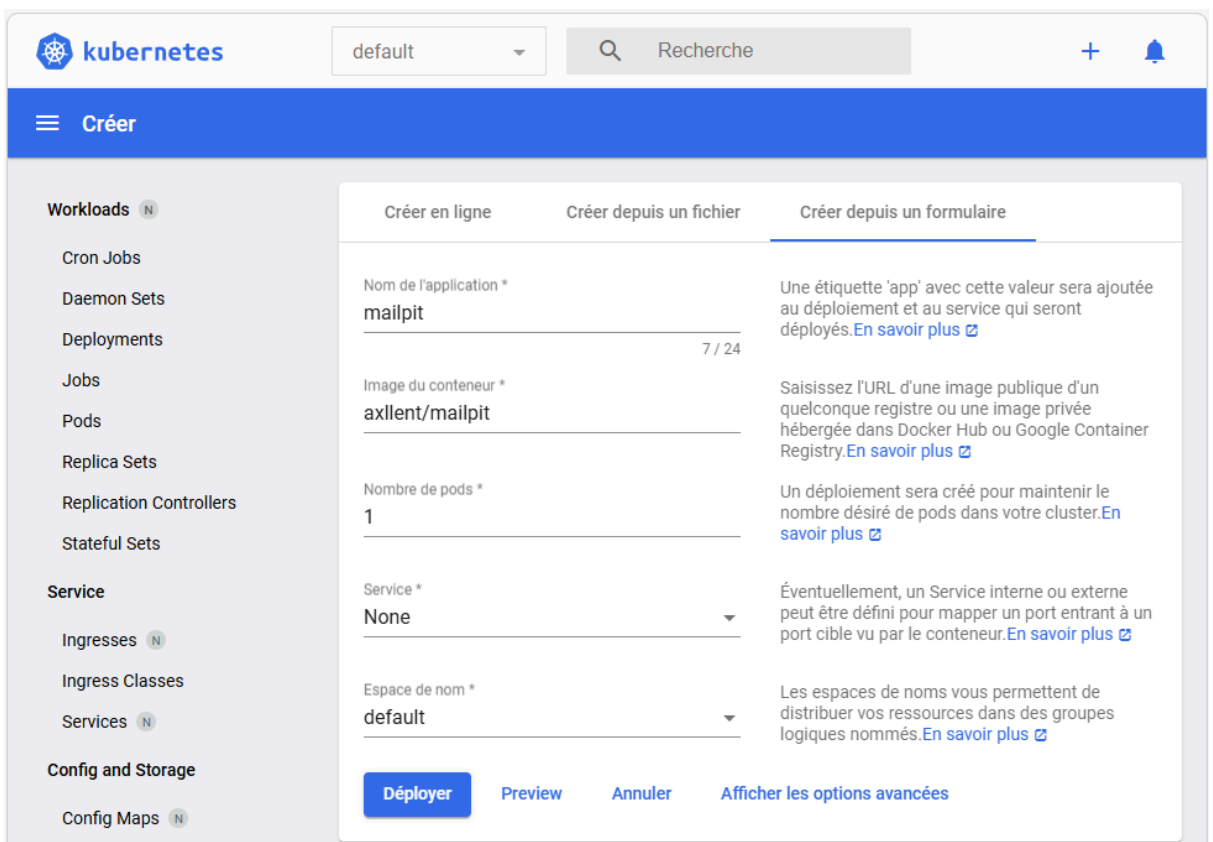
Affichage des Déploiements :



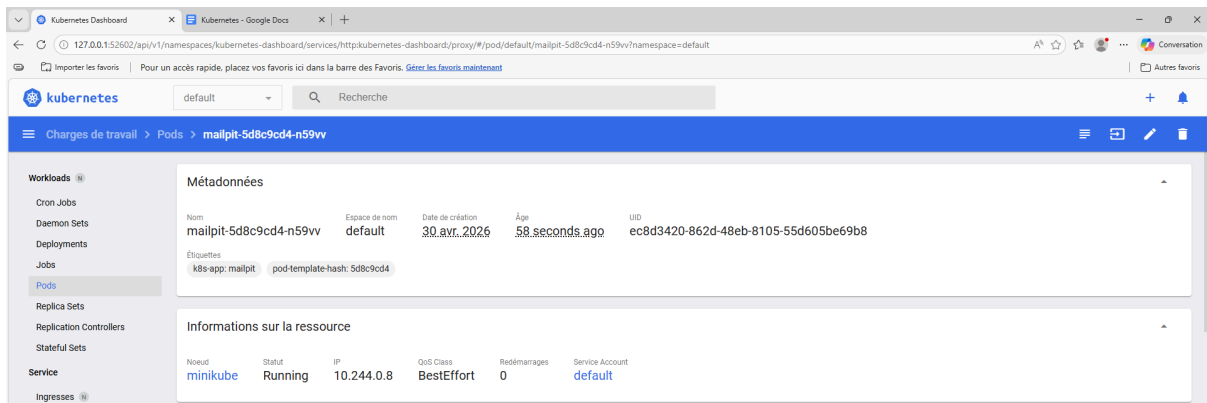
Affichage des Pods :



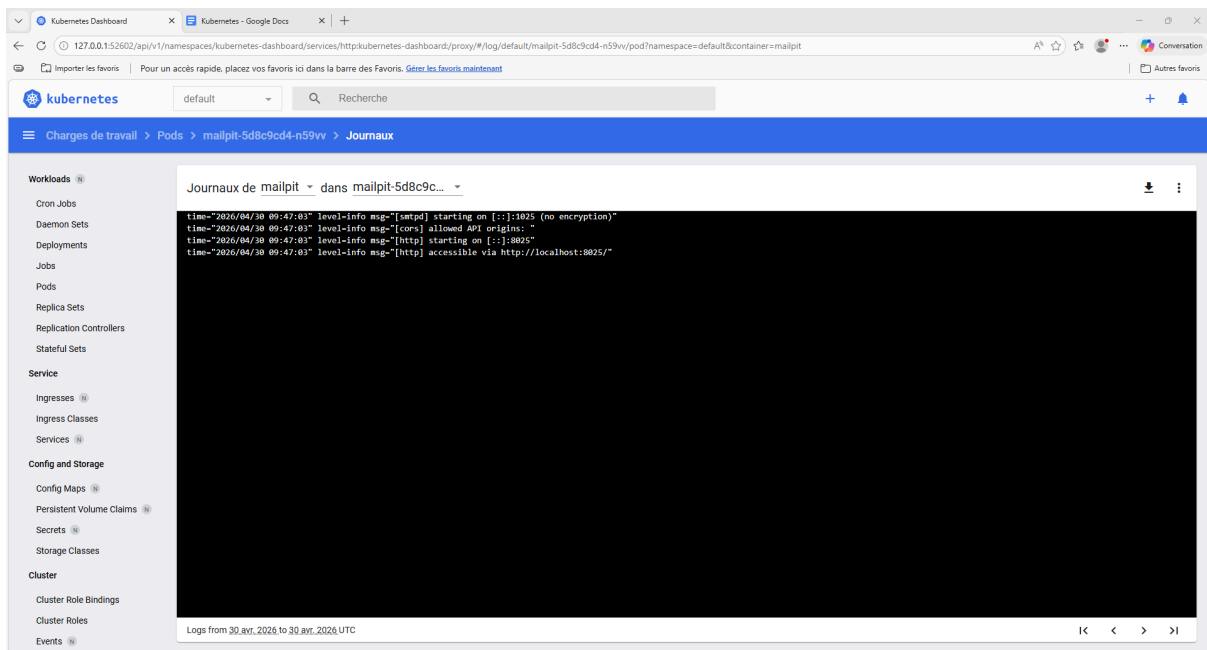
Déploiement de l'application Mailpit :



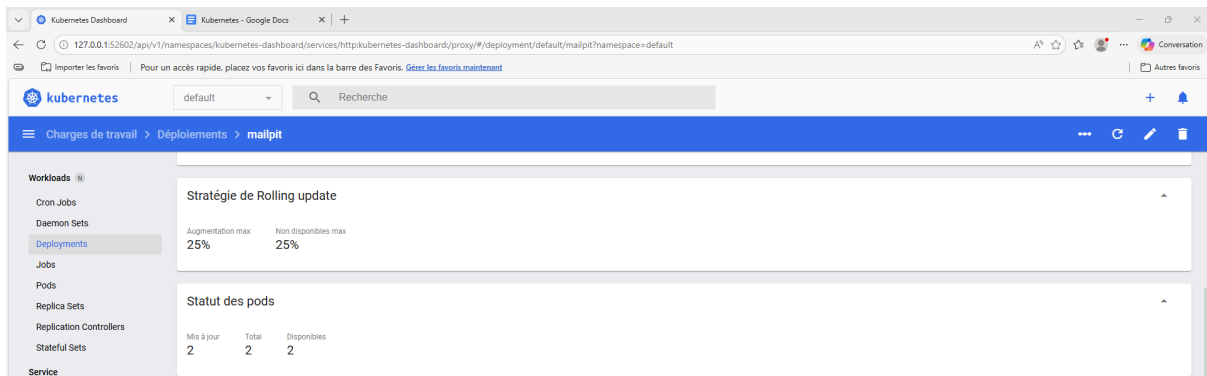
### Affichage des détails du pod Mailpit :



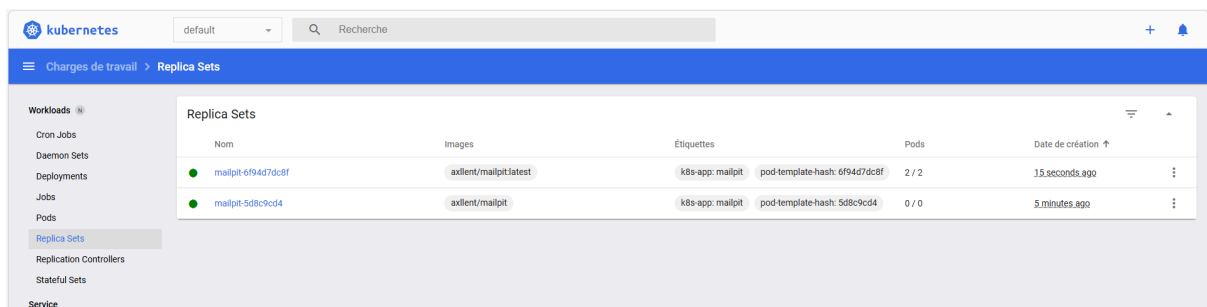
### Affichage des logs du pod Mailpit :



Affichage du déploiement Mailpit :



Affichage des ReplicaSets du cluster :



Mise à l'échelle du déploiement à deux répliques :

Mettre à l'échelle une ressource

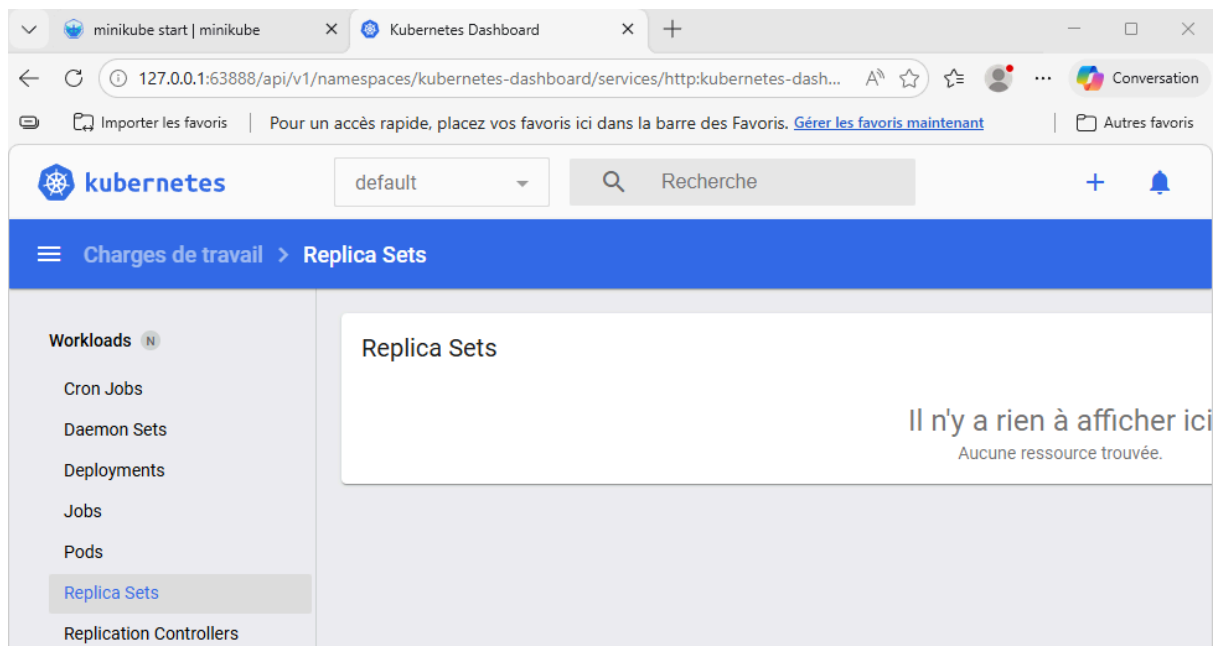
Deployment mailpit will be updated to reflect the desired replicas count.

Répliques désirées \*  Répliques actuelles

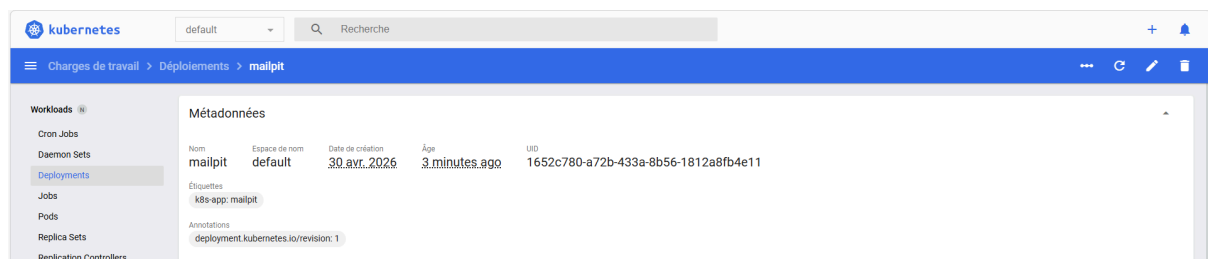
*i* Cette action est équivalente à : `kubectl scale -n default deployment mailpit --replicas=2`

[Mettre à l'échelle](#) [Annuler](#)

Affichage des ReplicaSets du cluster :




Mise à jour du déploiement Mailpit suite à la mise à l'échelle à 2 répliques.



Avant modification du fichier YAML du déploiement Mailpit :

### Éditer une ressource

| YAML | JSON   |
|------|--|
| 1    | <code>kind: Deployment</code>                          |
| 2    | <code>apiVersion: apps/v1</code>                       |
| 3    | <code>metadata:</code>                                 |
| 4    | <code>name: mailpit</code>                             |
| 5    | <code>namespace: default</code>                        |
| 6    | <code>uid: 1652c780-a72b-433a-8b56-1812a8fb4e11</code> |
| 7    | <code>resourceVersion: '1613'</code>                   |
| 8    | <code>generation: 2</code>                             |
| 9    | <code>creationTimestamp: '2026-04-30T09:46:58Z'</code> |
| 10   | <code>labels:</code>                                   |
| 11   | <code>k8s-app: mailpit</code>                          |
| 12   | <code>annotations:</code>                              |
| 13   | <code>deployment.kubernetes.io/revision: '1'</code>    |
| 14   | <code>managedFields:</code>                            |
| 15   | - <code>manager: dashboard</code>                      |
| 16   | <code>operation: Update</code>                         |
| 17   | <code>apiVersion: apps/v1</code>                       |
| 18   | <code>fieldsType: FieldsV1</code>                      |
| 19   | <code>fieldsV1:</code>                                 |
| 20   | <code>f:spec:</code>                                   |
| 21   | <code>f:replicas: {}</code>                            |

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

[Mettre à jour](#)

[Annuler](#)

Après modification du fichier YAML du déploiement Mailpit pour mettre à jour la configuration de l'application.

### Éditer une ressource

YAML

JSON

```

110     name: mailpit
111     labels:
112     |       k8s-app: mailpit
113     spec:
114     |       containers:
115     |       |       - name: mailpit
116     |       |         image: axllent/mailpit:latest
117     |       |         resources: {}
118     |       |         terminationMessagePath: /dev/termination-log
119     |       |         terminationMessagePolicy: File
120     |       |         imagePullPolicy: Always
121     |       |         securityContext:
122     |       |         |         privileged: false
123     |       |         restartPolicy: Always
124     |       |         terminationGracePeriodSeconds: 30
125     |       |         dnsPolicy: ClusterFirst
126     |       |         securityContext: {}
127     |       |         schedulerName: default-scheduler
128     |       strategy:
129     |       |         type: RollingUpdate
130     |       |         rollingUpdate:
131     |       |         |         maxUnavailable: 25%
    
```

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

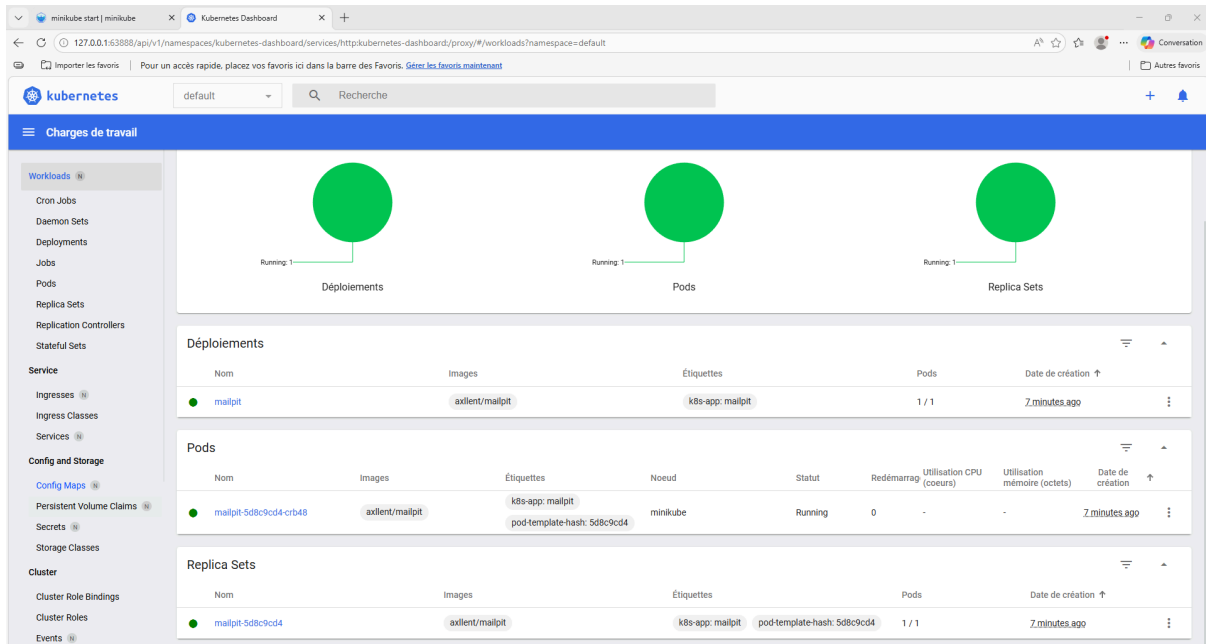
Mettre à jour
Annuler

La mise à jour du déploiement Mailpit crée un nouveau ReplicaSet avec les modifications appliquées.

The screenshot shows the Kubernetes dashboard interface. At the top, there's a navigation bar with 'kubernetes' and a search bar. Below it, the breadcrumb trail reads 'Charges de travail > Déploiements > mailpit'. A table shows the deployment status: 'Progressing', 'True', '38 seconds ago', '6 minutes ago', 'NewReplicaSetAvailable', and 'ReplicaSet "mailpit-6f94d7dc8f" has successfully progressed.'. Below the table, a 'Nouveau Replica Set' card is displayed with the following details:

- Nom: mailpit-6f94d7dc8f
- Espace de nom: default
- Âge: 43 seconds ago
- Pods: 2 / 2
- Étiquettes: k8s-app: mailpit, pod-template-hash: 6f94d7dc8f
- Images: axllent/mailpit:latest

Le tableau de bord permet de visualiser l'état des ressources (déploiements, pods, replicats sets) du cluster Kubernetes.



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

La commande **kubectl get deployment** permet d'afficher les déploiements du cluster et leur état.

```
Administrateur : Windows Po1 X + v
PS C:\Users\nprovenzano> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mailpit   2/2     2            2           7m19s
PS C:\Users\nprovenzano>
```

La commande **kubectl delete deployment mailpit** permet de supprimer le déploiement Mailpit, ce qui entraîne la suppression des pods et des ReplicaSets associés.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl delete deployment mailpit
deployment.apps "mailpit" deleted from default namespace
PS C:\Users\nprovenzano>
```

La commande **kubectl get deployment** permet de confirmer que le déploiement Mailpit a bien été supprimé du cluster.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl get deployment
No resources found in default namespace.
PS C:\Users\nprovenzano>
```

La commande **kubectl create deployment mailpit --image=axllent/mailpit** permet de créer un déploiement Mailpit à partir d'une image Docker.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl create deployment mailpit --image=axllent/mailpit
deployment.apps/mailpit created
PS C:\Users\nprovenzano>
```

La commande **kubectl get deployment -o wide** permet d'afficher les détails du déploiement Mailpit.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mailpit   1/1     1            1           19s
PS C:\Users\nprovenzano> kubectl get deployment -o wide
NAME      READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES           SELECTOR
mailpit   1/1     1            1           24s   mailpit      axllent/mailpit app=mailpit
PS C:\Users\nprovenzano>
```

La commande **kubectl describe deployment mailpit** permet d'afficher les informations détaillées du déploiement Mailpit.

```
Administrateur : Windows Po... x + v
PS C:\Users\nprovenzano> kubectl describe deployment mailpit
Name: mailpit
Namespace: default
CreationTimestamp: Thu, 30 Apr 2026 11:56:25 +0200
Labels: app=mailpit
Annotations: deployment.kubernetes.io/revision: 1
Selector: app=mailpit
Replicas: 1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels: app=mailpit
  Containers:
    mailpit:
      Image: axllent/mailpit
      Port: <none>
      Host Port: <none>
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
      Node-Selectors: <none>
      Tolerations: <none>
Conditions:
  Type           Status  Reason
  ----           -
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets: <none>
NewReplicaSet: mailpit-7b96b48c5d (1/1 replicas created)
Events:
  Type           Reason             Age   From          Message
  ----           -
  Normal        ScalingReplicaSet  55s   deployment-controller  Scaled up replica set mailpit-7b96b48c5d from 0 to 1
PS C:\Users\nprovenzano>
```

La commande **kubectl get replicaset** permet de lister les ReplicaSets, et **kubectl describe rs** permet d'afficher leurs détails comme les pods associés et la configuration.

```

Administrateur : Windows Po...
PS C:\Users\nprovenzano> kubectl get replicaset
NAME                DESIRED   CURRENT   READY   AGE
mailpit-7b96b48c5d  1         1         1       85s
PS C:\Users\nprovenzano> kubectl describe rs mailpit-7b96b48c5d
Name:                mailpit-7b96b48c5d
Namespace:           default
Selector:            app=mailpit,pod-template-hash=7b96b48c5d
Labels:              app=mailpit
                    pod-template-hash=7b96b48c5d
Annotations:         deployment.kubernetes.io/desired-replicas: 1
                    deployment.kubernetes.io/max-replicas: 2
                    deployment.kubernetes.io/revision: 1
Controlled By:       Deployment/mailpit
Replicas:            1 current / 1 desired
Pods Status:         1 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:             app=mailpit
                    pod-template-hash=7b96b48c5d
  Containers:
  mailpit:
    Image:            axllent/mailpit
    Port:             <none>
    Host Port:        <none>
    Environment:      <none>
    Mounts:           <none>
    Volumes:          <none>
    Node-Selectors:   <none>
    Tolerations:      <none>
Events:
  Type      Reason              Age   From                      Message
  ----      -
  Normal    SuccessfulCreate    112s  replicaset-controller     Created pod: mailpit-7b96b48c5d-hhn7j
PS C:\Users\nprovenzano>

```

La commande **kubectl get pods** permet de lister les pods, et **kubectl describe pod** permet d'afficher les détails comme l'état, l'IP et la configuration.

```

Administrateur : Windows Po...
PS C:\Users\nprovenzano> kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-hhn7j  1/1     Running   0           2m18s
PS C:\Users\nprovenzano> kubectl describe pods mailpit-7b96b48c5d
Name:                mailpit-7b96b48c5d-hhn7j
Namespace:           default
Priority:             0
Service Account:     default
Node:                minikube/172.22.204.133
Start Time:          Thu, 30 Apr 2026 11:56:25 +0200
Labels:              app=mailpit
                    pod-template-hash=7b96b48c5d
Annotations:         <none>
Status:              Running
IP:                  10.244.0.12
IPs:
  IP:                10.244.0.12
Controlled By:       ReplicaSet/mailpit-7b96b48c5d
Containers:
  mailpit:
    Container ID:     docker://21d4e17239734488b024761d7d05ee3fe5ad0b2ff680fd3fc83242d08c876fbc
    Image:            axllent/mailpit
    Image ID:         docker-pullable://axllent/mailpit@sha256:757f22b56c1da03570afdb3d259effe5091018008a81bbec8158cee7e16

```

Suite de l'image :

```
Containers:
  mailpit:
    Container ID:   docker://21d4e17239734488b024761d7d05ee3fe5ad0b2ff680fd3fc83242d08c876fbc
    Image:          axllent/mailpit
    Image ID:       docker-pullable://axllent/mailpit@sha256:757f22b56c1da03570afdb3d259effe5091018008a81bbedc8158cee7e16fdbc
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 30 Apr 2026 11:56:26 +0200
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-wtrk7 (ro)
Conditions:
  Type                                     Status
  PodReadyToStartContainers              True
  Initialized                             True
  Ready                                   True
  ContainersReady                         True
  PodScheduled                            True
Volumes:
  kube-api-access-wtrk7:
```

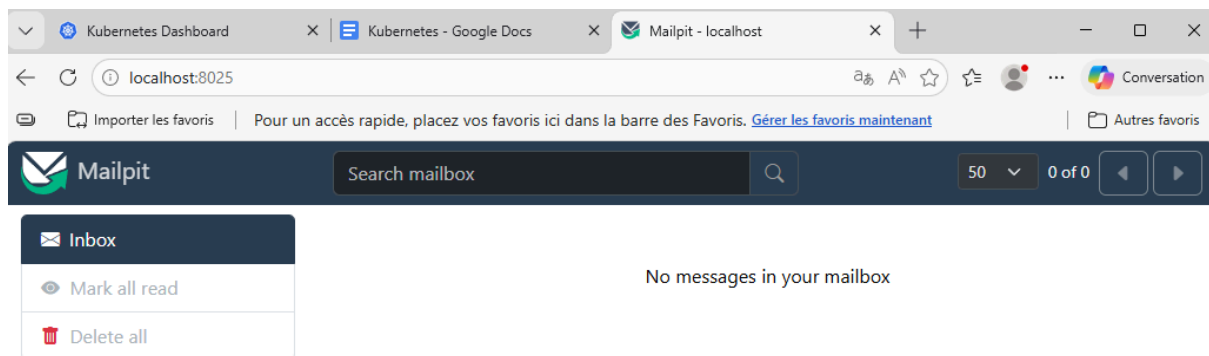
La commande **kubectl logs** permet de consulter les logs du pod Mailpit pour vérifier le démarrage et le fonctionnement du service.

```
Administrateur : Windows Po... x + v
PS C:\Users\nprovenzano> kubectl logs mailpit-7b96b48c5d-hhn7j
time="2026/04/30 09:56:26" level=info msg="[smtpd] starting on [::]:1025 (no encryption)"
time="2026/04/30 09:56:26" level=info msg="[cors] allowed API origins: "
time="2026/04/30 09:56:26" level=info msg="[http] starting on [::]:8025"
time="2026/04/30 09:56:26" level=info msg="[http] accessible via http://localhost:8025/"
PS C:\Users\nprovenzano> kubectl logs mailpit-7b96b48c5d-hhn7j -c mailpit
time="2026/04/30 09:56:26" level=info msg="[smtpd] starting on [::]:1025 (no encryption)"
time="2026/04/30 09:56:26" level=info msg="[cors] allowed API origins: "
time="2026/04/30 09:56:26" level=info msg="[http] starting on [::]:8025"
time="2026/04/30 09:56:26" level=info msg="[http] accessible via http://localhost:8025/"
PS C:\Users\nprovenzano> |
```

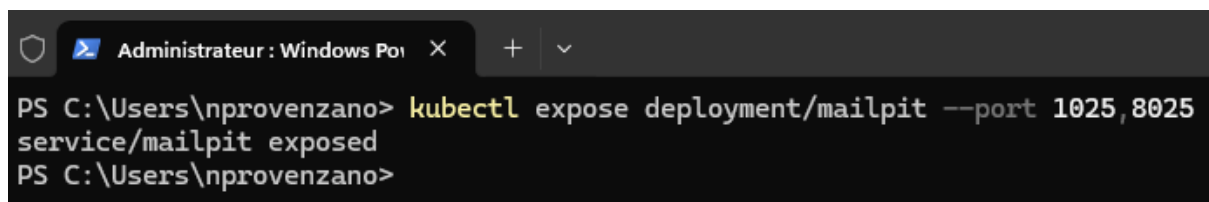
La commande **kubectl port-forward** permet d'accéder au service Mailpit en redirigeant le port du pod vers localhost.

```
Administrateur : Windows Po... x + v
PS C:\Users\nprovenzano> kubectl port-forward mailpit-7b96b48c5d-hhn7j 8025
Forwarding from 127.0.0.1:8025 -> 8025
Forwarding from [::1]:8025 -> 8025
Handling connection for 8025
Handling connection for 8025
Handling connection for 8025
```

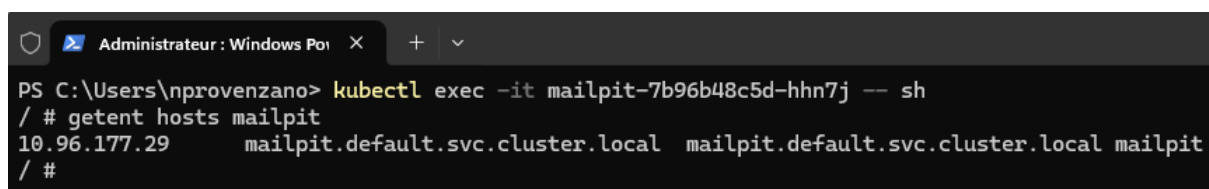
L'accès à **localhost:8025** permet de vérifier l'interface web de Mailpit.



La commande **kubectl expose deployment mailpit** permet de créer un service pour exposer l'application Mailpit sur des ports accessibles.



La commande **kubectl exec -it ... -- sh** permet d'accéder au terminal du conteneur pour exécuter des commandes à l'intérieur du pod.



La commande **kubectl run** permet de créer un pod temporaire pour tester la résolution DNS du service Mailpit.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl run -it --rm pod-test --image=alpine sh
All commands and output from this session will be recorded in container logs, including credentials and sensitive information passed through the command prompt.
If you don't see a command prompt, try pressing enter.
/ # nslookup mailpit
Server:      10.96.0.10
Address:    10.96.0.10:53

** server can't find mailpit.cluster.local: NXDOMAIN

Name:      mailpit.default.svc.cluster.local
Address: 10.96.177.29

** server can't find mailpit.svc.cluster.local: NXDOMAIN

** server can't find mailpit.cluster.local: NXDOMAIN

** server can't find mailpit.svc.cluster.local: NXDOMAIN

/ # exit
Session ended, resume using 'kubectl attach pod-test -c pod-test -i -t' command when the pod is running
pod "pod-test" deleted from default namespace
PS C:\Users\nprovenzano>
```

La commande **kubectl scale deployment mailpit --replicas=2** permet de mettre à l'échelle le déploiement en créant 2 répliques de l'application.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl scale deployment mailpit --replicas=2
deployment.apps/mailpit scaled
PS C:\Users\nprovenzano>
```

Les commandes **kubectl get deployment** et **kubectl get pods** permettent de vérifier que le déploiement Mailpit possède bien deux répliques en fonctionnement.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl get deployment mailpit
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mailpit   2/2     2            2           14m
PS C:\Users\nprovenzano> kubectl get pods -l app=mailpit
NAME                                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-hhn7j            1/1     Running   0           14m
mailpit-7b96b48c5d-sphd4            1/1     Running   0           42s
PS C:\Users\nprovenzano>
```

La commande **minikube addons enable ingress** permet d'activer le contrôleur Ingress dans le cluster Kubernetes.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> minikube addons enable ingress
⚠ ingress est un addon maintenu par Kubernetes. Pour toute question, contactez minikube sur GitHub.
Vous pouvez consulter la liste des mainteneurs de minikube sur : https://github.com/kubernetes/minikube/blob/master/OWNER
S
  ▪ Utilisation de l'image registry.k8s.io/ingress-nginx/controller:v1.14.3
  ▪ Utilisation de l'image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.6.7
  ▪ Utilisation de l'image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.6.7
  🔵 Vérification du module ingress...
  🌟 Le module 'ingress' est activé
PS C:\Users\nprovenzano>
```

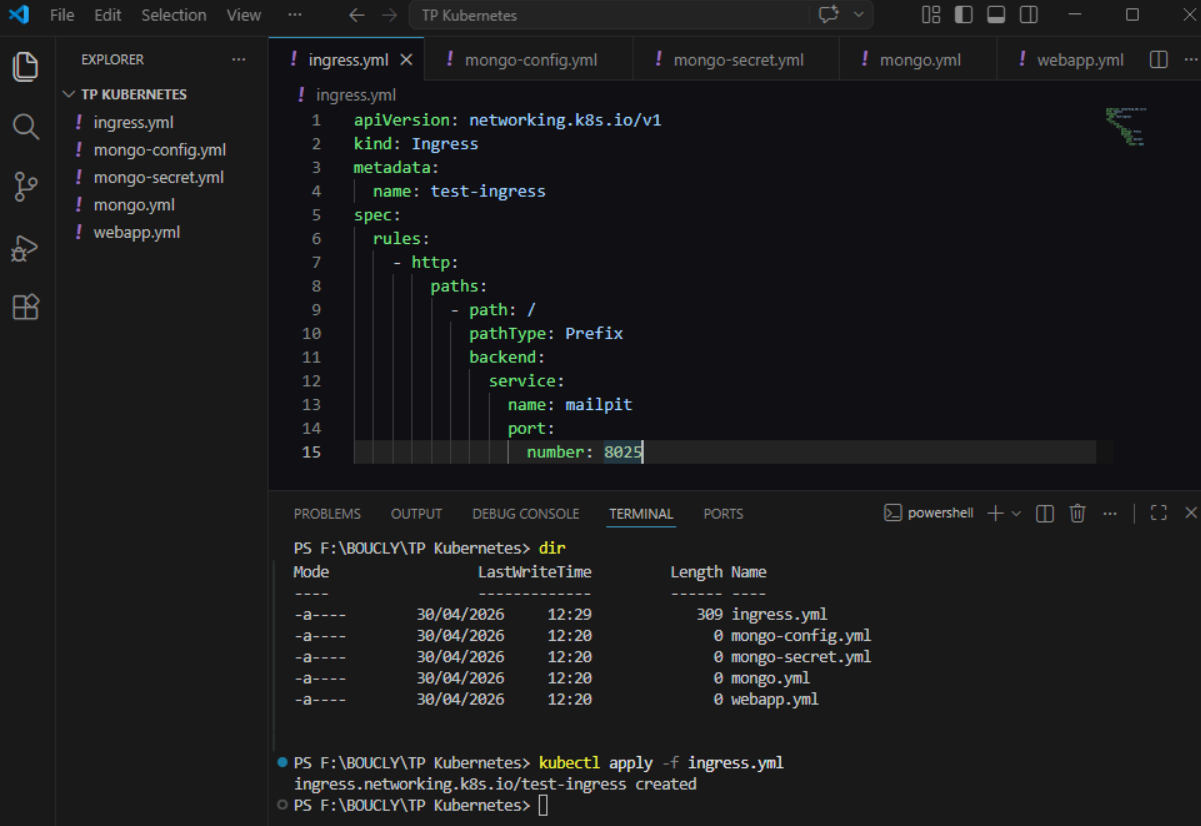
La commande **kubectl get namespace** permet d'afficher les différents espaces de noms du cluster Kubernetes.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl get namespace
NAME                STATUS    AGE
default             Active   42m
ingress-nginx       Active   54s
kube-node-lease     Active   42m
kube-public         Active   42m
kube-system         Active   42m
kubernetes-dashboard Active   28m
PS C:\Users\nprovenzano>
```

La commande **kubectl -n ingress-nginx get pods** permet de vérifier le fonctionnement des pods du contrôleur Ingress. Trois pods sont présents : deux terminés (Completed) et un pod principal en état Running, indique que le contrôleur Ingress est actif.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl -n ingress-nginx get pods -l app.kubernetes.io/name
NAME                                READY   STATUS    RESTARTS   AGE
ingress-nginx-admission-create-vkmth 0/1     Completed 0           2m9s
ingress-nginx-admission-patch-42hjc  0/1     Completed 0           2m9s
ingress-nginx-controller-596f8778bc-nqd9v 1/1     Running   0           2m9s
PS C:\Users\nprovenzano>
```

Le fichier **ingress.yml** permet de définir une règle Ingress pour rediriger les requêtes vers le service Mailpit.



The screenshot shows a Visual Studio Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a folder named 'TP KUBERNETES' containing several YAML files: ingress.yml, mongo-config.yml, mongo-secret.yml, mongo.yml, and webapp.yml. The main editor window displays the content of 'ingress.yml' with the following code:

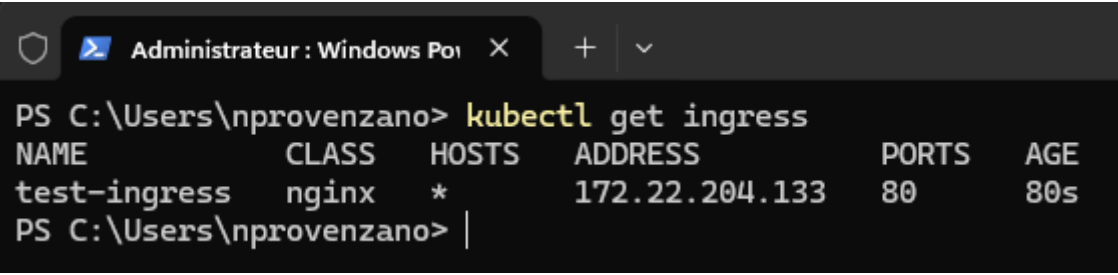
```
1 apiVersion: networking.k8s.io/v1
2 kind: Ingress
3 metadata:
4   name: test-ingress
5 spec:
6   rules:
7     - http:
8       paths:
9         - path: /
10           pathType: Prefix
11             backend:
12               service:
13                 name: mailpit
14                 port:
15                   number: 8025
```

The terminal window shows the following commands and output:

```
PS F:\BOUCLY\TP Kubernetes> dir
Mode                LastWriteTime         Length Name
----                -
-a----             30/04/2026   12:29           309 ingress.yml
-a----             30/04/2026   12:20             0 mongo-config.yml
-a----             30/04/2026   12:20             0 mongo-secret.yml
-a----             30/04/2026   12:20             0 mongo.yml
-a----             30/04/2026   12:20             0 webapp.yml

PS F:\BOUCLY\TP Kubernetes> kubectl apply -f ingress.yml
ingress.networking.k8s.io/test-ingress created
PS F:\BOUCLY\TP Kubernetes>
```

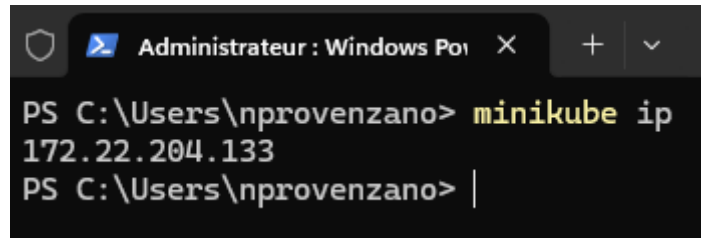
La commande **kubectl get ingress** permet de vérifier que la ressource Ingress a bien été créée et d'obtenir l'adresse d'accès à l'application.



The screenshot shows a Windows PowerShell terminal window with the following output:

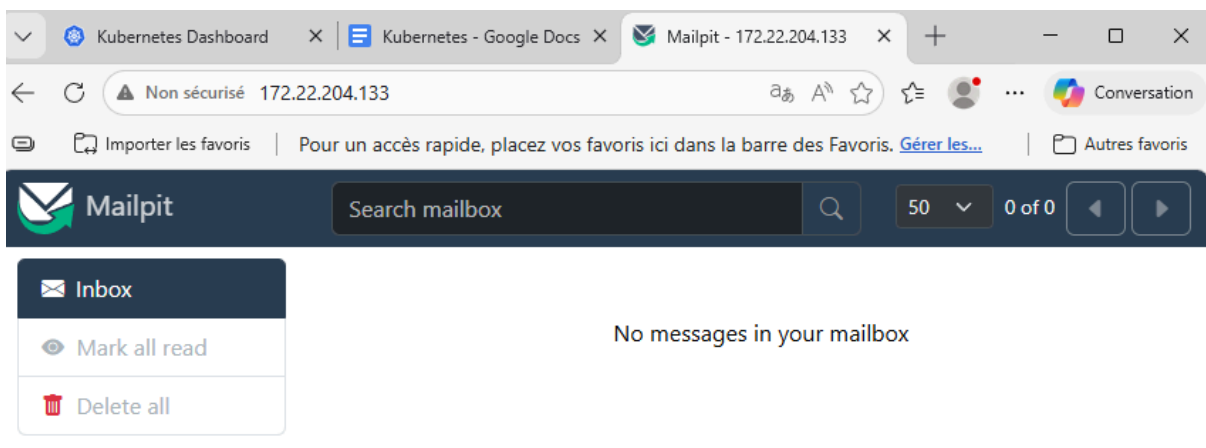
```
PS C:\Users\nprovenzano> kubectl get ingress
NAME          CLASS  HOSTS      ADDRESS          PORTS  AGE
test-ingress  nginx  *          172.22.204.133  80     80s
PS C:\Users\nprovenzano> |
```

La commande **minikube ip** permet d'afficher l'adresse IP du cluster Minikube pour accéder à l'interface de Mailpit.



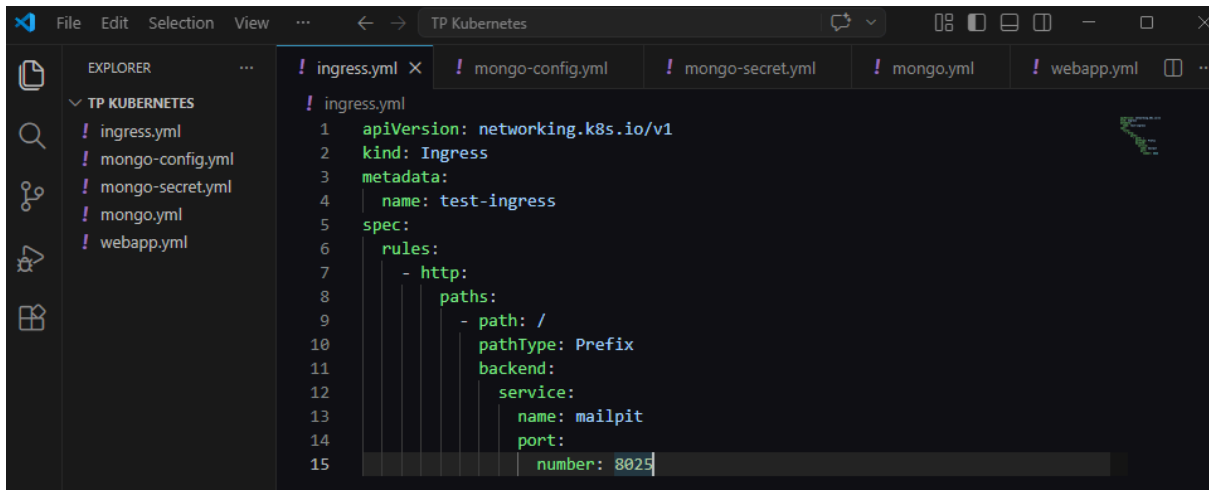
```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> minikube ip
172.22.204.133
PS C:\Users\nprovenzano> |
```

L'accès à l'adresse IP du cluster Minikube dans le navigateur permet de confirmer que l'application Mailpit est bien accessible depuis l'extérieur grâce à Ingress.



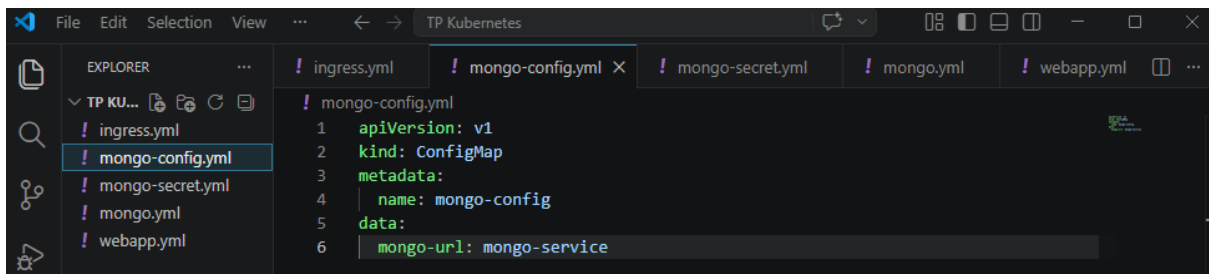
## 5. Automatisation de déploiement par fichier YAML

Le fichier **ingress.yml** permet de définir une règle Ingress redirigeant les requêtes vers le service Mailpit.



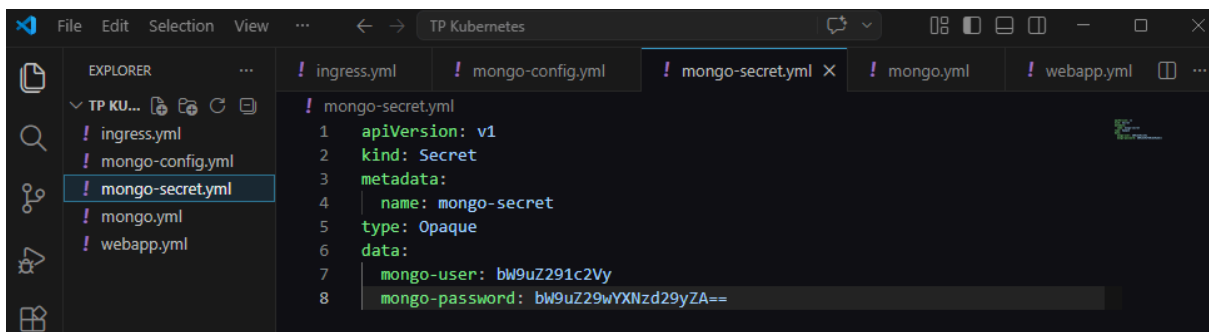
```
! ingress.yml
1  apiVersion: networking.k8s.io/v1
2  kind: Ingress
3  metadata:
4    name: test-ingress
5  spec:
6    rules:
7      - http:
8          paths:
9            - path: /
10             pathType: Prefix
11             backend:
12               service:
13                 name: mailpit
14                 port:
15                   number: 8025
```

Le fichier **mongo-config.yml** permet de créer une ConfigMap contenant les paramètres de configuration de MongoDB.



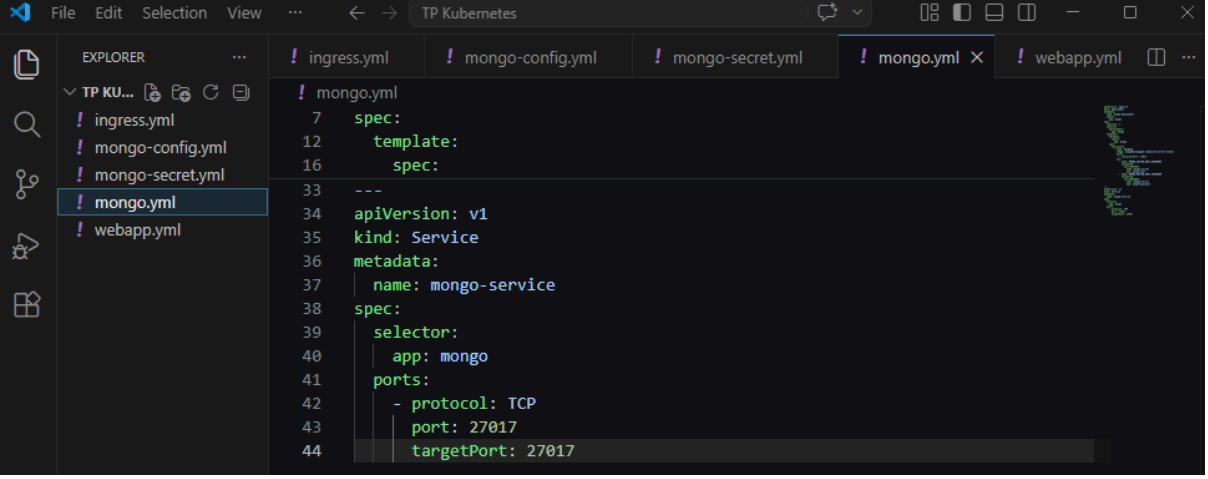
```
! mongo-config.yml
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: mongo-config
5  data:
6    mongo-url: mongo-service
```

Le fichier **mongo-secret.yml** permet de créer un Secret afin de stocker de manière sécurisée les identifiants et mots de passe de MongoDB.



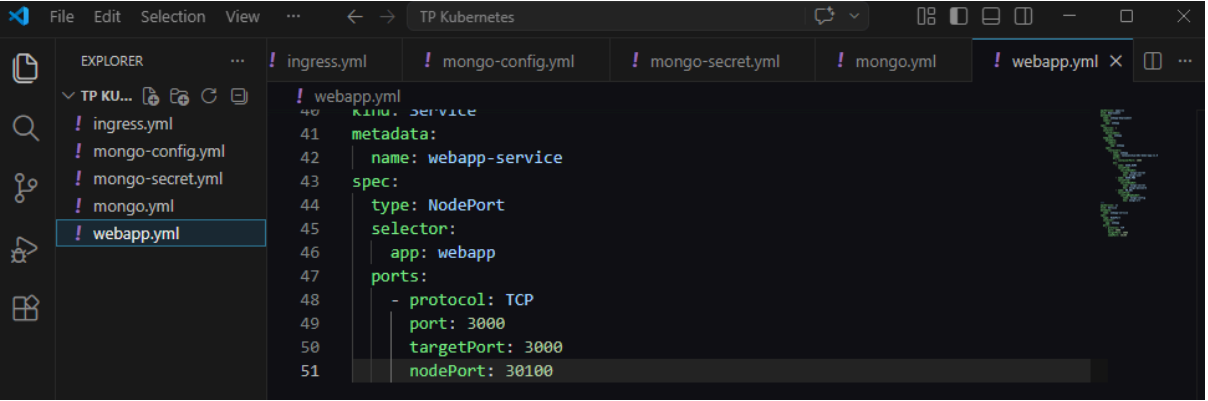
```
! mongo-secret.yml
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: mongo-secret
5  type: Opaque
6  data:
7    mongo-user: bW9uZ291c2Vy
8    mongo-password: bW9uZ29wYXNzd29yZA==
```

Le fichier **mongo.yml** permet de créer un déploiement MongoDB et un service associé pour permettre la communication avec les autres applications.



```
File Edit Selection View ... TP Kubernetes
! ingress.yml ! mongo-config.yml ! mongo-secret.yml ! mongo.yml x ! webapp.yml ...
TP KU...
! ingress.yml
! mongo-config.yml
! mongo-secret.yml
! mongo.yml
! webapp.yml
! mongo.yml
7 spec:
12   template:
16     spec:
33   ---
34   apiVersion: v1
35   kind: Service
36   metadata:
37     name: mongo-service
38   spec:
39     selector:
40       app: mongo
41     ports:
42     - protocol: TCP
43       port: 27017
44       targetPort: 27017
```

Le fichier **webapp.yml** permet de créer le déploiement de l'application web et un service NodePort afin de rendre l'application accessible depuis l'extérieur du cluster.



```
File Edit Selection View ... TP Kubernetes
! ingress.yml ! mongo-config.yml ! mongo-secret.yml ! mongo.yml ! webapp.yml x ...
TP KU...
! ingress.yml
! mongo-config.yml
! mongo-secret.yml
! mongo.yml
! webapp.yml
! webapp.yml
40 kind: Service
41 metadata:
42   name: webapp-service
43   spec:
44     type: NodePort
45     selector:
46       app: webapp
47     ports:
48     - protocol: TCP
49       port: 3000
50       targetPort: 3000
51       nodePort: 30100
```

La commande **kubectl apply -f** permet de déployer les composants de l'application à partir des fichiers YAML.

```

! webapp.yml
40 kind: Service
41 metadata:
42   name: webapp-service
43 spec:
44   type: NodePort
45   selector:
46     app: webapp
47   ports:
48   - protocol: TCP
49     port: 3000
50     targetPort: 3000
51     nodePort: 30100

PS F:\BOUCLY\TP Kubernetes> kubectl apply -f mongo-config.yml
configmap/mongo-config created
PS F:\BOUCLY\TP Kubernetes> kubectl apply -f mongo-secret.yml
secret/mongo-secret created
PS F:\BOUCLY\TP Kubernetes> kubectl apply -f mongo.yml
deployment.apps/mongo-deployment created
service/mongo-service created
PS F:\BOUCLY\TP Kubernetes> kubectl apply -f webapp.yml
error: no objects passed to apply
PS F:\BOUCLY\TP Kubernetes> kubectl apply -f webapp.yml
deployment.apps/webapp-deployment created
service/webapp-service created
PS F:\BOUCLY\TP Kubernetes>

```

La commande **kubectl get all** permet d'afficher l'ensemble des ressources du cluster (pods, services, déploiements et ReplicaSets).

```

PS F:\BOUCLY\TP Kubernetes> kubectl get all

pod/mailpit-7b96b48c5d-hhn7j      1/1      Running      0          63m
pod/mailpit-7b96b48c5d-sphd4     1/1      Running      0          49m
pod/mongo-deployment-744864fdd7-n5dj1  1/1      Running      0          3m19s
pod/webapp-deployment-6d88954d4c-kxmdg  0/1      ImagePullBackOff  0          2m58s

NAME                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/kubernetes  ClusterIP     10.96.0.1        <none>            443/TCP          89m
service/mailpit     ClusterIP     10.96.177.29    <none>            1025/TCP,8025/TCP  52m
service/mongo-service ClusterIP     10.100.26.88    <none>            27017/TCP        3m19s
service/webapp-service NodePort      10.107.105.109  <none>            3000:30100/TCP   2m58s

NAME                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mailpit  2/2     2             2           63m
deployment.apps/mongo-deployment  1/1     1             1           3m19s
deployment.apps/webapp-deployment  0/1     1             0           2m58s

NAME                DESIRED   CURRENT   READY   AGE
replicaset.apps/mailpit-7b96b48c5d  2         2         2       63m
replicaset.apps/mongo-deployment-744864fdd7  1         1         1       3m19s
replicaset.apps/webapp-deployment-6d88954d4c  1         1         0       2m58s
PS F:\BOUCLY\TP Kubernetes>

```

Les commandes **kubectl get configmap** et **kubectl get secret** permettent de vérifier la présence des configurations et des données sensibles utilisées par l'application.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + - [ ] [ ] ... [ ] [ ] X
● PS F:\BOUCLY\TP Kubernetes> kubectl get configmap
NAME          DATA  AGE
kube-root-ca.crt  1     90m
mongo-config    1     4m29s
● PS F:\BOUCLY\TP Kubernetes> kubectl get secret
NAME          TYPE  DATA  AGE
mongo-secret  Opaque  2     4m25s
PS F:\BOUCLY\TP Kubernetes>
```

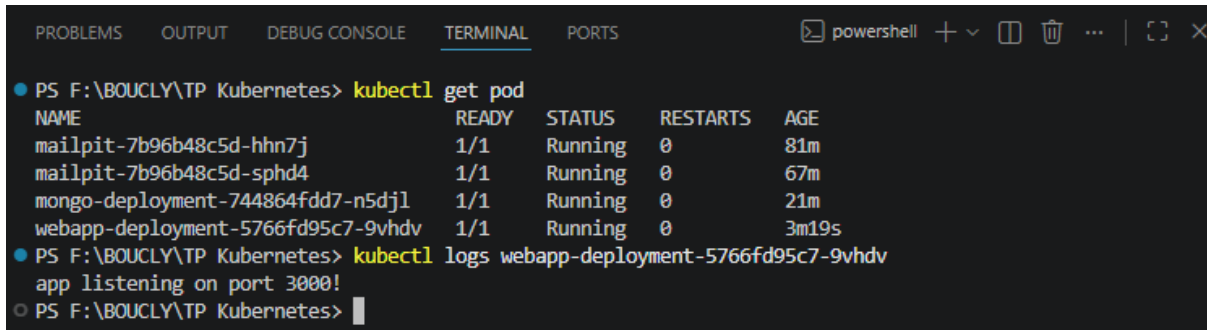
La commande **kubectl describe service webapp-service** permet d'afficher la configuration du service, notamment les ports, l'adresse IP et le type NodePort.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + - [ ] [ ] ... [ ] [ ] X
● PS F:\BOUCLY\TP Kubernetes> kubectl describe service webapp-service
Name:          webapp-service
Namespace:     default
Labels:        <none>
Annotations:   <none>
Selector:      app=webapp
Type:          NodePort
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.107.105.109
IPs:           10.107.105.109
Port:          <unset> 3000/TCP
TargetPort:    3000/TCP
NodePort:      <unset> 30100/TCP
Endpoints:
Session Affinity: None
External Traffic Policy: Cluster
Internal Traffic Policy: Cluster
Events:        <none>
○ PS F:\BOUCLY\TP Kubernetes> |
```

La commande **kubectl describe pod** permet d'afficher les détails du pod et de détecter une erreur de récupération de l'image empêchant son démarrage.

```
PS F:\BOUCLY\TP Kubernetes> kubectl describe pod webapp-deployment-6d88954d4c-kxmdg
Priority: 0
Service Account: default
Node: minikube/172.22.204.133
Start Time: Thu, 30 Apr 2026 12:56:30 +0200
Labels: app=webapp
        pod-template-hash=6d88954d4c
Annotations: <none>
Status: Pending
IP: 10.244.0.19
IPs:
  IP: 10.244.0.19
Controlled By: ReplicaSet/webapp-deployment-6d88954d4c
Containers:
  webapp:
    Container ID:
    Image: nanajanshja/k8s-demo-app:v1.0
    Image ID:
    Port: 3000/TCP
    Host Port: 0/TCP
    State: Waiting
      Reason: ImagePullBackOff
    Ready: False
    Restart Count: 0
    Environment:
      USER_NAME: <set to the key 'mongo-user' in secret 'mongo-secret'> Optional: false
      USER_PWD: <set to the key 'mongo-password' in secret 'mongo-secret'> Optional: false
      DB_URL: <set to the key 'mongo-url' of config map 'mongo-config'> Optional: false
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-tpjz (ro)
Conditions:
  Type                               Status
  PodReadyToStartContainers          True
  Initialized                         True
  Ready                              False
  ContainersReady                   False
  PodScheduled                       True
Volumes:
  kube-api-access-tpjz:
    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
```

La commande **kubectl describe pod** permet d'analyser l'état du pod webapp et d'identifier une erreur de type ImagePullBackOff empêchant le démarrage du conteneur.

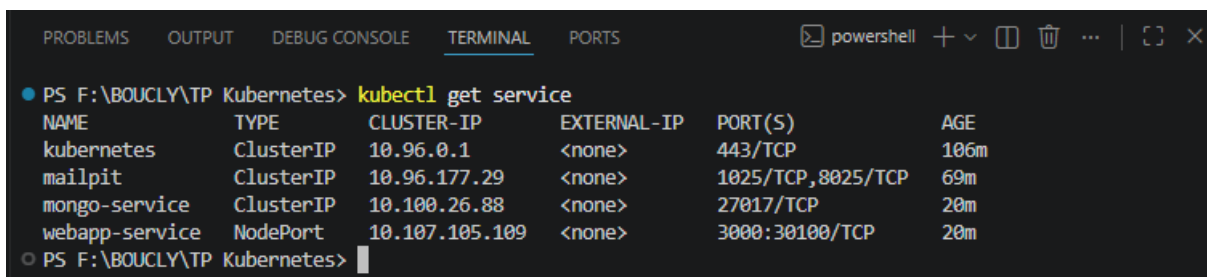


```

PS F:\BOUCLY\TP Kubernetes> kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-hhn7j           1/1     Running   0           81m
mailpit-7b96b48c5d-sphd4           1/1     Running   0           67m
mongo-deployment-744864fdd7-n5djl  1/1     Running   0           21m
webapp-deployment-5766fd95c7-9vhdv 1/1     Running   0           3m19s
PS F:\BOUCLY\TP Kubernetes> kubectl logs webapp-deployment-5766fd95c7-9vhdv
app listening on port 3000!
PS F:\BOUCLY\TP Kubernetes>

```

La commande **kubectl get service** permet de lister les services et de vérifier leurs types, adresses IP et ports.

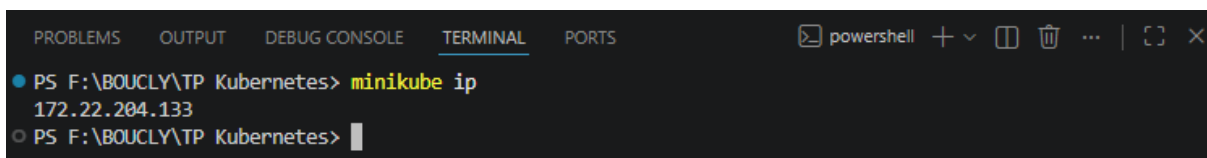


```

PS F:\BOUCLY\TP Kubernetes> kubectl get service
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes     ClusterIP      10.96.0.1      <none>        443/TCP          106m
mailpit        ClusterIP      10.96.177.29   <none>        1025/TCP,8025/TCP 69m
mongo-service  ClusterIP      10.100.26.88   <none>        27017/TCP        20m
webapp-service NodePort        10.107.105.109 <none>        3000:30100/TCP   20m
PS F:\BOUCLY\TP Kubernetes>

```

La commande **minikube ip** permet de récupérer l'adresse IP du cluster afin d'accéder aux services déployés.

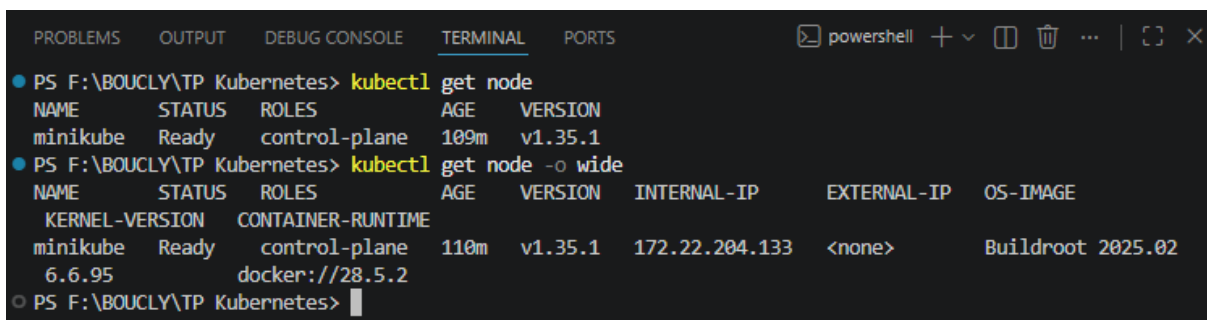


```

PS F:\BOUCLY\TP Kubernetes> minikube ip
172.22.204.133
PS F:\BOUCLY\TP Kubernetes>

```

Les commandes **kubectl get node** et **kubectl get node -o wide** permettent de vérifier l'état du nœud ainsi que des informations détaillées comme l'adresse IP.



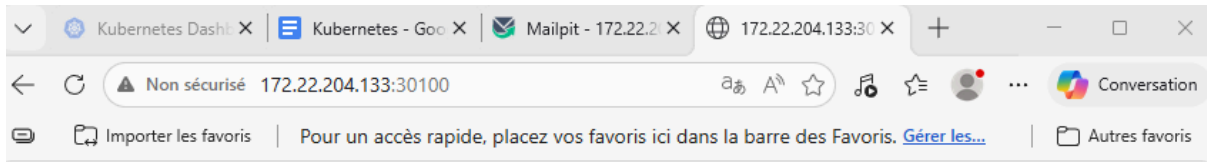
```

PS F:\BOUCLY\TP Kubernetes> kubectl get node
NAME           STATUS    ROLES    AGE   VERSION
minikube       Ready    control-plane  109m  v1.35.1
PS F:\BOUCLY\TP Kubernetes> kubectl get node -o wide
NAME           STATUS    ROLES    AGE   VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE
KERNEL-VERSION CONTAINER-RUNTIME
minikube       Ready    control-plane  110m  v1.35.1   172.22.204.133 <none>        Buildroot 2025.02
6.6.95         docker://28.5.2
PS F:\BOUCLY\TP Kubernetes>

```



L'accès à l'adresse IP et au port du service permet d'afficher l'interface de l'application web.  
> Edit Profile



## User profile



Name: **Anna Smith**

---

Email: **anna.smith@example.com**

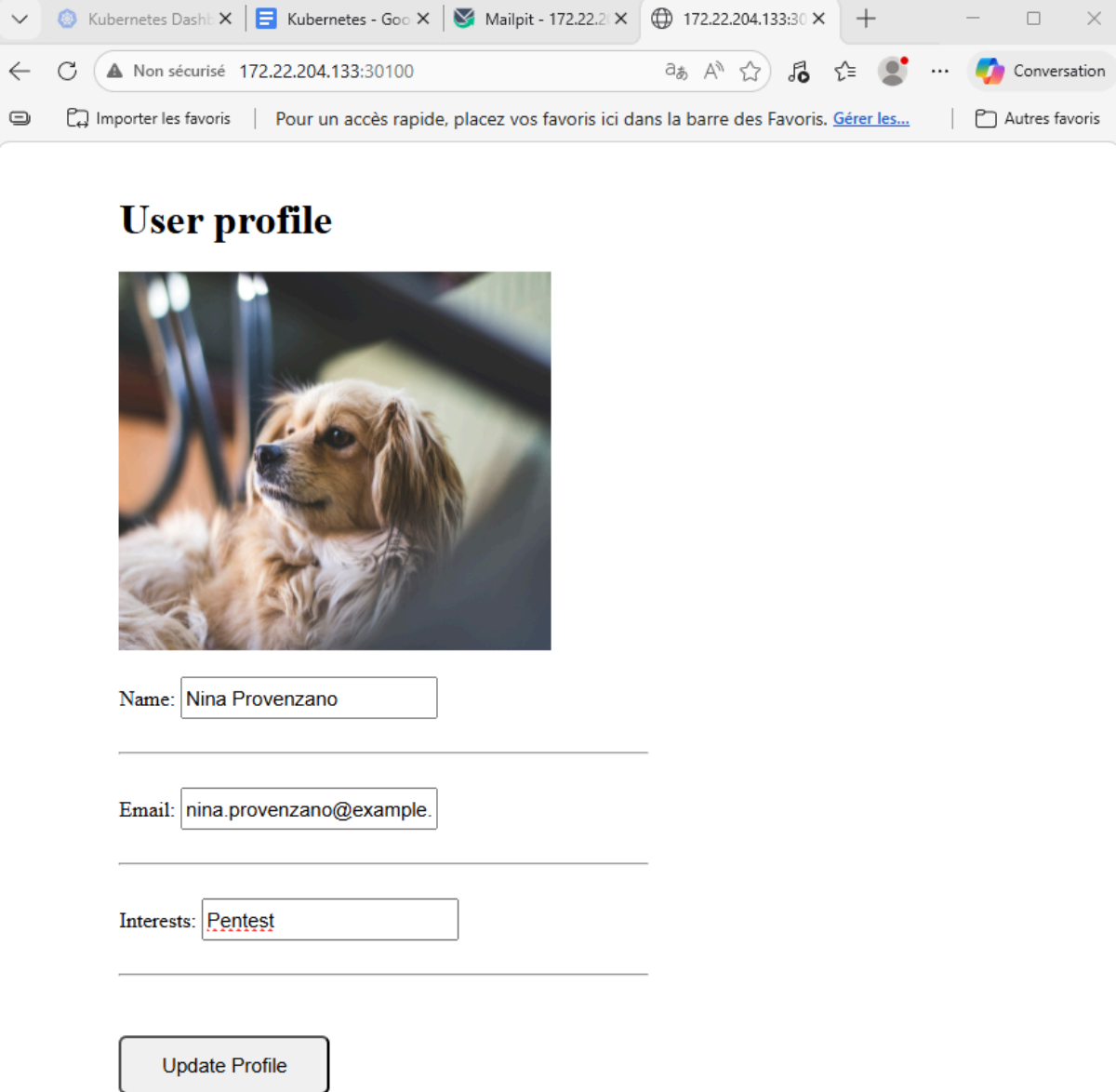
---

Interests: **coding**


---

Edit Profile

La modification du profil permet de vérifier que les données de l'application web peuvent être mises à jour.



**User profile**



Name:

---

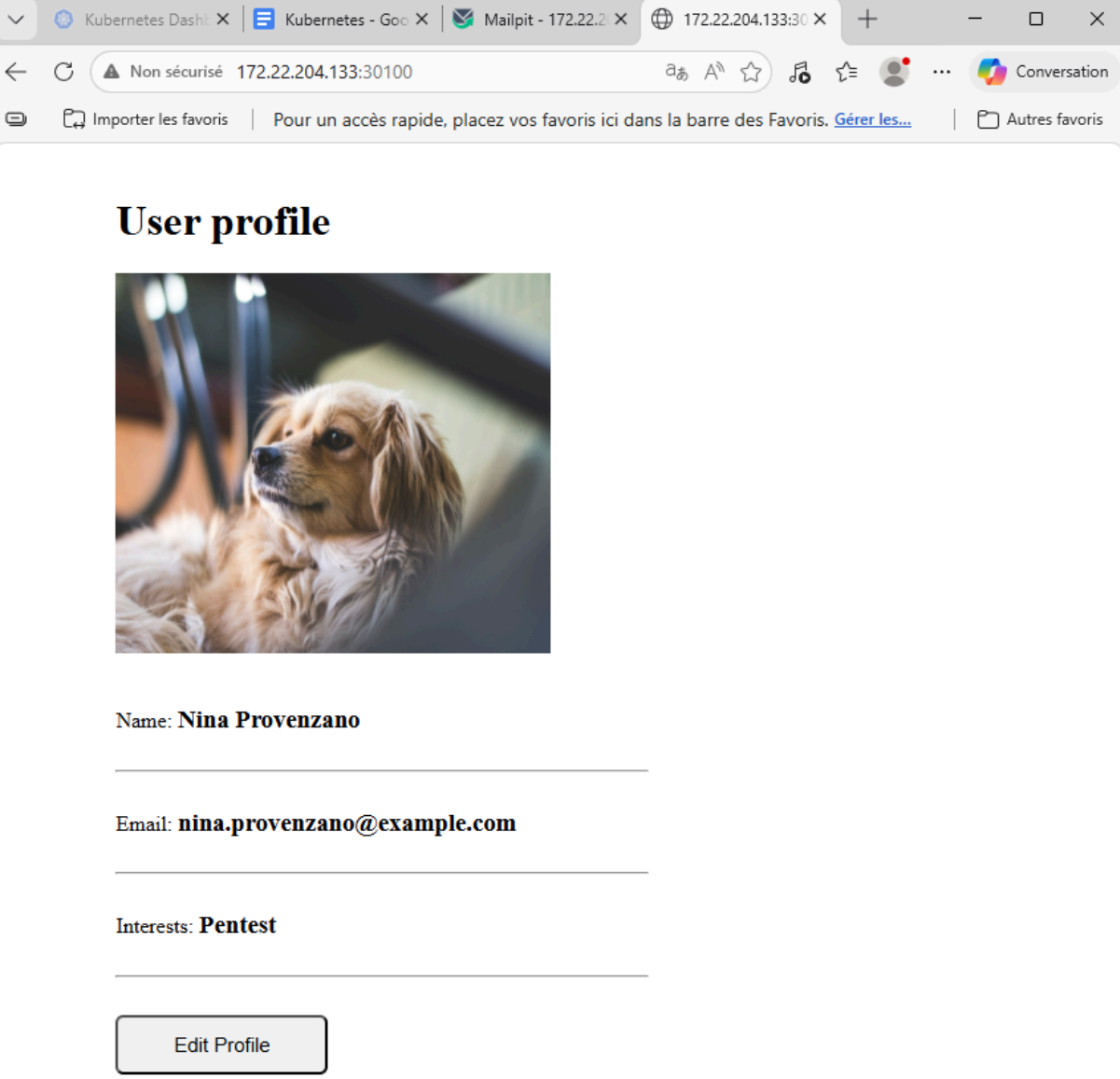
Email:

---


Interests:

---

L'affichage du profil mis à jour permet de vérifier que les modifications ont bien été enregistrées dans l'application.



**User profile**



Name: **Nina Provenzano**

---

Email: **nina.provenzano@example.com**

---

Interests: **Pentest**

---

[Edit Profile](#)



La commande **kubectl get all** permet de vérifier que toutes les ressources de l'application WordPress et MySQL sont correctement déployées.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS F:\BOUCLY\TP Kubernetes\wordpress-k8s> kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/mailpit-7b96b48c5d-hhn7j           1/1     Running   0           100m
pod/mailpit-7b96b48c5d-sphd4           1/1     Running   0           87m
pod/mongo-deployment-744864fdd7-n5djl   1/1     Running   0           40m
pod/mysql-66f86944b7-hd5b6             1/1     Running   0           4m22s
pod/webapp-deployment-5766fd95c7-9vhdv  1/1     Running   0           22m
pod/wordpress-59fcc67747-v119f         1/1     Running   0           4m20s

NAME                                     TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/kubernetes                       ClusterIP     10.96.0.1       <none>           443/TCP          127m
service/mailpit                           ClusterIP     10.96.177.29   <none>           1025/TCP,8025/TCP 90m
service/mongo-service                     ClusterIP     10.100.26.88   <none>           27017/TCP        40m
service/mysql                              ClusterIP     10.107.152.182 <none>           3306/TCP         4m22s
service/webapp-service                     NodePort     10.107.105.109 <none>           3000:30100/TCP   40m
service/wordpress                         NodePort     10.108.202.173 <none>           80:30080/TCP     4m22s

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mailpit                  2/2     2             2           100m
deployment.apps/mongo-deployment         1/1     1             1           40m
deployment.apps/mysql                     1/1     1             1           4m22s
deployment.apps/webapp-deployment        1/1     1             1           40m
deployment.apps/wordpress                 1/1     1             1           4m20s

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/mailpit-7b96b48c5d       2         2         2       100m
replicaset.apps/mongo-deployment-744864fdd7 1         1         1       40m
replicaset.apps/mysql-66f86944b7         1         1         1       4m22s
replicaset.apps/webapp-deployment-5766fd95c7 1         1         1       22m
replicaset.apps/webapp-deployment-6d88954d4c 0         0         0       40m
replicaset.apps/wordpress-59fcc67747     1         1         1       4m20s
○ PS F:\BOUCLY\TP Kubernetes\wordpress-k8s>
    
```

La commande **kubectl get pvc** permet d'afficher les volumes persistants utilisés par l'application.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell - wordpress-k8s
● PS F:\BOUCLY\TP Kubernetes\wordpress-k8s> kubectl get pvc
NAME                                     CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM                STORAGECLASS  VOLUMEATTRIBUTESCLASS  REASON  AGE
mysql-pv                                2Gi       RWO            Retain           Available  default/wordpress-pvc  standard      <unset>                 <unset>  4m43s
pvc-15b23d24-7525-4ae4-895a-b57dd439c6e9 2Gi       RWO            Delete           Bound     default/wordpress-pvc  standard      <unset>                 <unset>  4m43s
pvc-5f49d2b5-9cad-49b2-bd00-3da2e09ac5c8 2Gi       RWO            Delete           Bound     default/mysql-pvc      standard      <unset>                 <unset>  4m43s
wordpress-pv                             2Gi       RWO            Retain           Available  <none>                 <unset>                 <unset>  4m43s
○ PS F:\BOUCLY\TP Kubernetes\wordpress-k8s>
    
```



L'interface d'installation de WordPress permet de renseigner les informations nécessaires à la création du site.

**Bienvenue**

Bienvenue dans la très célèbre installation en 5 minutes de WordPress ! Vous n'avez qu'à remplir les informations demandées ci-dessous et vous serez prêt à utiliser la plus extensible et puissante plateforme de publication de contenu au monde.

### Informations nécessaires

Veuillez renseigner les informations suivantes. Ne vous inquiétez pas, vous pourrez les modifier plus tard.

**Titre du site**

**Identifiant**

Les identifiants ne peuvent utiliser que des caractères alphanumériques, des espaces, des tirets bas ("\_"), des traits d'union ("-"), des points et le symbole @.

**Mot de passe**

Moyenne

**Important :** Vous aurez besoin de ce mot de passe pour vous connecter. Pensez à le stocker dans un lieu sûr.

**Votre e-mail**

Vérifiez bien cette adresse e-mail avant de continuer.

**Visibilité par les moteurs de recherche**  Demander aux moteurs de recherche de ne pas indexer ce site

Certains moteurs de recherche peuvent décider de l'indexer malgré tout.

Activer Windows  
Accédez aux paramètres pour activer Windows.

Saisie des informations pour configurer les paramètres administrateur du site WordPress.

**Bienvenue**

Bienvenue dans la très célèbre installation en 5 minutes de WordPress ! Vous n'avez qu'à remplir les informations demandées ci-dessous et vous serez prêt à utiliser la plus extensible et puissante plateforme de publication de contenu au monde.

### Informations nécessaires

Veillez renseigner les informations suivantes. Ne vous inquiétez pas, vous pourrez les modifier plus tard.

**Titre du site**

**Identifiant**   
Les identifiants ne peuvent utiliser que des caractères alphanumériques, des espaces, des tirets bas ("\_"), des traits d'union ("-"), des points et le symbole @.

**Mot de passe**    
Moyenne

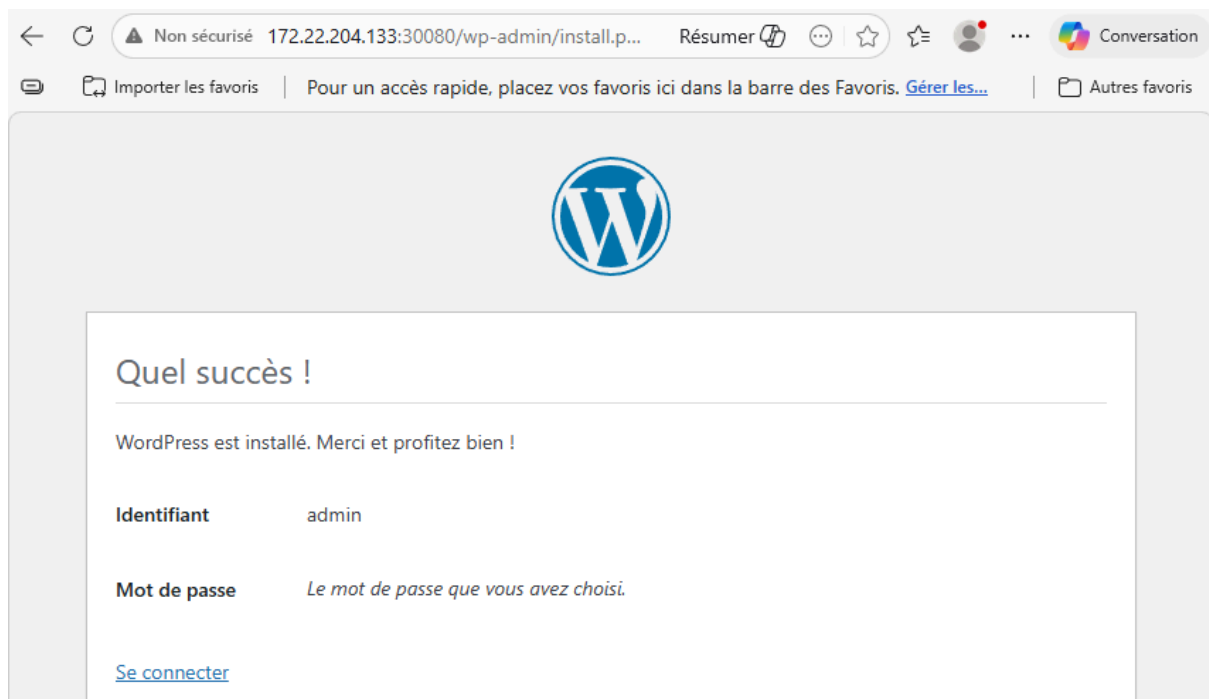
**Important** : Vous aurez besoin de ce mot de passe pour vous connecter. Pensez à le stocker dans un lieu sûr.

**Votre e-mail**   
Vérifiez bien cette adresse e-mail avant de continuer.

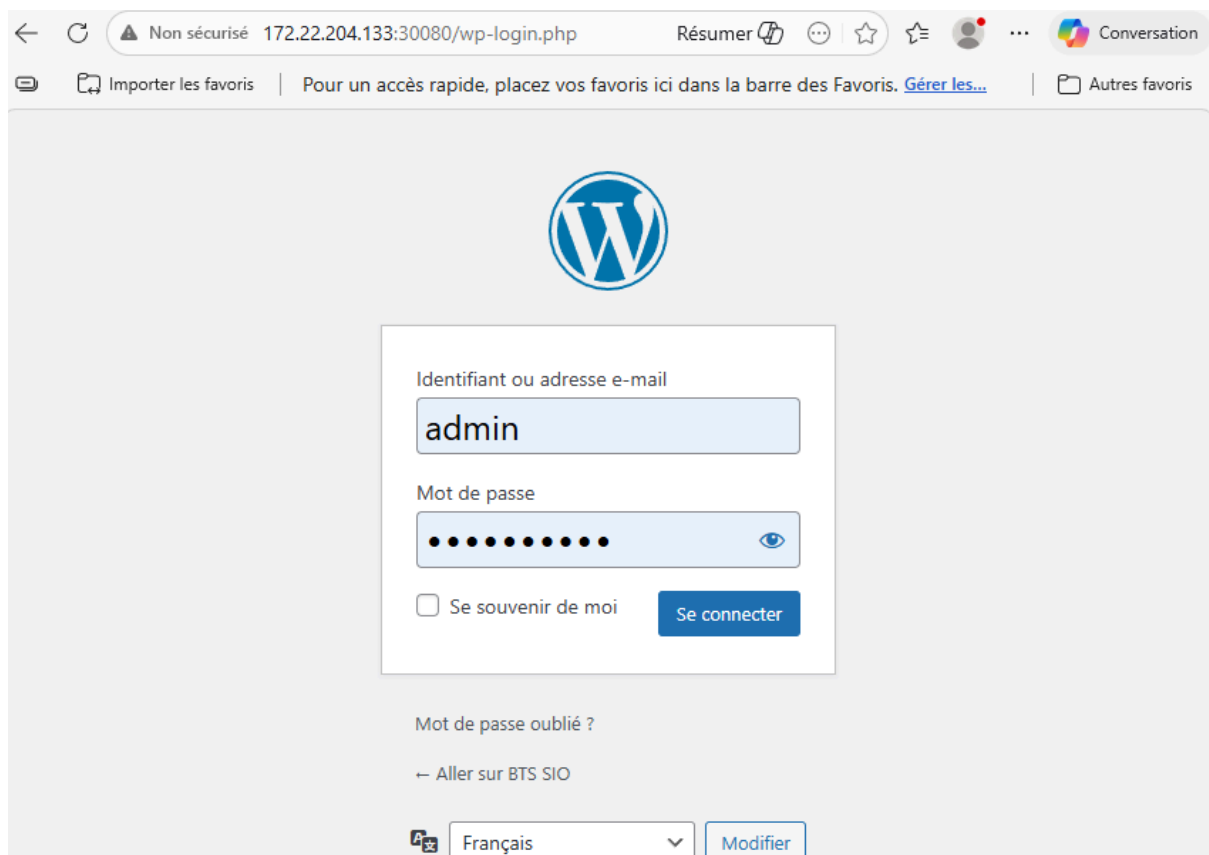
**Visibilité par les moteurs de recherche**  Demander aux moteurs de recherche de ne pas indexer ce site  
Certains moteurs de recherche peuvent décider de l'indexer malgré tout.

Activer Windows  
Accédez aux paramètres pour activer Windows.

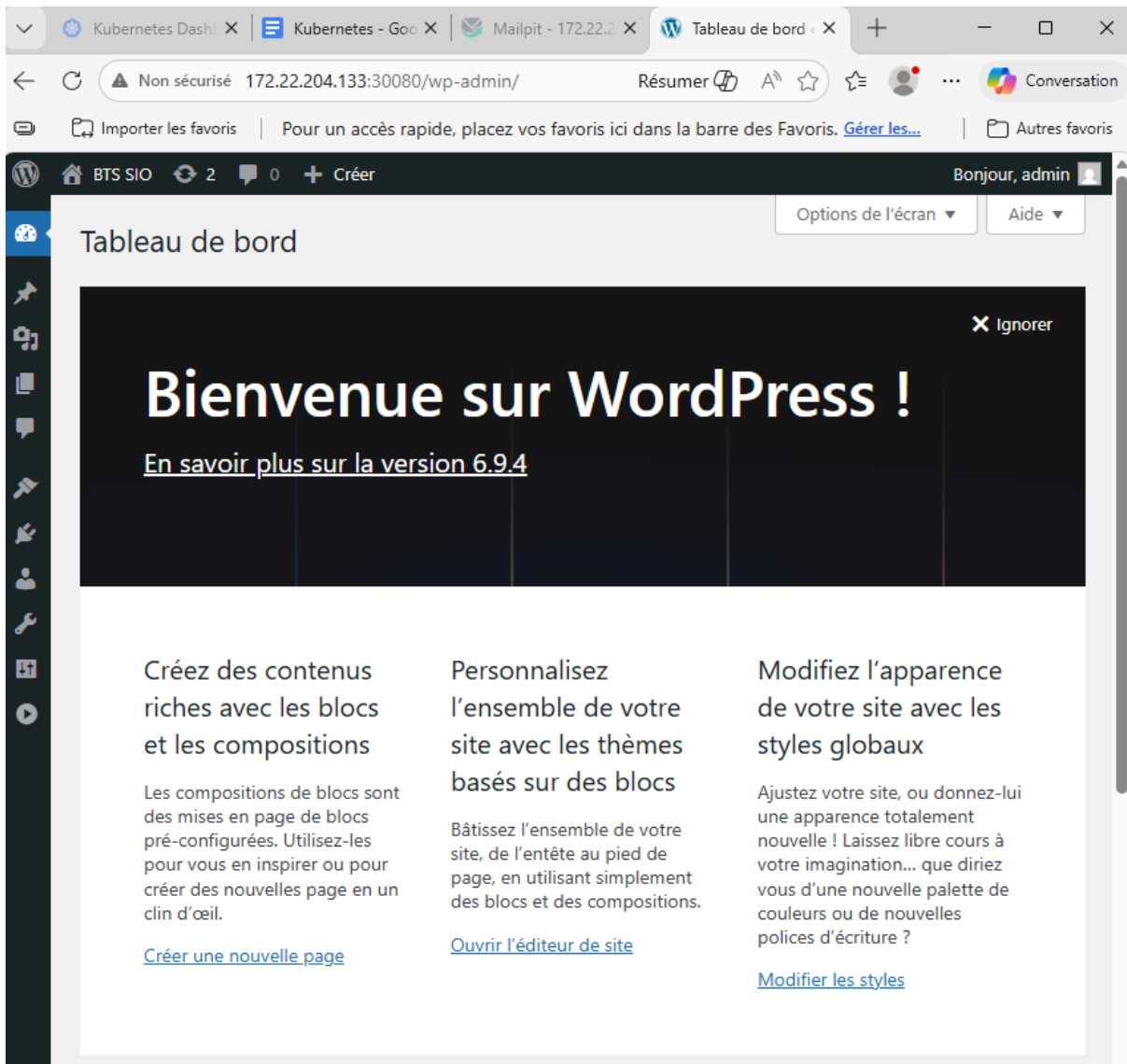
Confirmation que l'installation de WordPress a été réalisée avec succès.



Page de connexion de l'administrateur :



L'accès au tableau de bord WordPress permet d'administrer et de gérer le contenu du site.



La commande `kubectl delete pod -l app=wordpress` permet de supprimer les pods WordPress.

```
Administrateur : Windows Pow  Administrateur : Windows Poi  +  v
PS C:\Users\nprovenzano> kubectl delete pod -l app=wordpress
pod "wordpress-59fcc67747-vll9f" deleted from default namespace
PS C:\Users\nprovenzano> |
```

La commande **kubectl get all** permet de confirmer que les pods WordPress ont été recréés automatiquement par le déploiement, assurant la continuité du service.

```

PS C:\Users\nprovenzano> kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/mailpit-7b96b48c5d-hhn7j           1/1     Running   0           109m
pod/mailpit-7b96b48c5d-sphd4           1/1     Running   0           95m
pod/mongo-deployment-744864fdd7-n5djl  1/1     Running   0           49m
pod/mysql-66f86944b7-hd5b6            1/1     Running   0           12m
pod/webapp-deployment-5766fd95c7-9vhdv 1/1     Running   0           30m
pod/wordpress-59fcc67747-td5h2        1/1     Running   0           53s

NAME                                     TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/kubernetes                      ClusterIP      10.96.0.1       <none>           443/TCP          135m
service/mailpit                         ClusterIP      10.96.177.29   <none>           1025/TCP,8025/TCP 98m
service/mongo-service                   ClusterIP      10.100.26.88   <none>           27017/TCP        49m
service/mysql                           ClusterIP      10.107.152.182 <none>           3306/TCP         12m
service/webapp-service                   NodePort       10.107.105.109 <none>           3000:30100/TCP   49m
service/wordpress                       NodePort       10.108.202.173 <none>           80:30080/TCP     12m

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mailpit                 2/2     2             2           109m
deployment.apps/mongo-deployment        1/1     1             1           49m
deployment.apps/mysql                   1/1     1             1           12m
deployment.apps/webapp-deployment        1/1     1             1           49m
deployment.apps/wordpress               1/1     1             1           12m

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/mailpit-7b96b48c5d      2         2         2       109m
replicaset.apps/mongo-deployment-744864fdd7 1         1         1       49m
replicaset.apps/mysql-66f86944b7        1         1         1       12m
replicaset.apps/webapp-deployment-5766fd95c7 1         1         1       30m
replicaset.apps/webapp-deployment-6d88954d4c 0         0         0       49m
replicaset.apps/wordpress-59fcc67747    1         1         1       12m
PS C:\Users\nprovenzano>

```

La commande **kubectl get pv** permet de confirmer que les données sont conservées grâce aux volumes persistants malgré la suppression et la recréation des pods.

```

PS C:\Users\nprovenzano> kubectl get pv
NAME          CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
mysql-pv      2Gi        RWO            Retain           Available   default/wordpress-pvc  standard         <unset>           13m
pvc-15b23d24-7525-4ae4-895a-b57dd439c6e9 2Gi        RWO            Delete           Bound      default/wordpress-pvc  standard         <unset>           13m
pvc-5f49d2b5-9cad-49b2-bd60-3da2e09ac5c8 2Gi        RWO            Delete           Bound      default/mysql-pvc      standard         <unset>           13m
wordpress-pv 2Gi        RWO            Retain           Available   default/mysql-pvc      standard         <unset>           13m
PS C:\Users\nprovenzano>

```

La commande **kubectl get pvc** permet de confirmer que les PersistentVolumeClaims sont toujours actifs et associés aux volumes de stockage.

```

PS C:\Users\nprovenzano> kubectl get pvc
NAME          STATUS   VOLUME          CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mysql-pvc     Bound   pvc-5f49d2b5-9cad-49b2-bd60-3da2e09ac5c8 2Gi        RWO            standard       <unset>           13m
wordpress-pvc Bound   pvc-15b23d24-7525-4ae4-895a-b57dd439c6e9 2Gi        RWO            standard       <unset>           13m
PS C:\Users\nprovenzano>

```

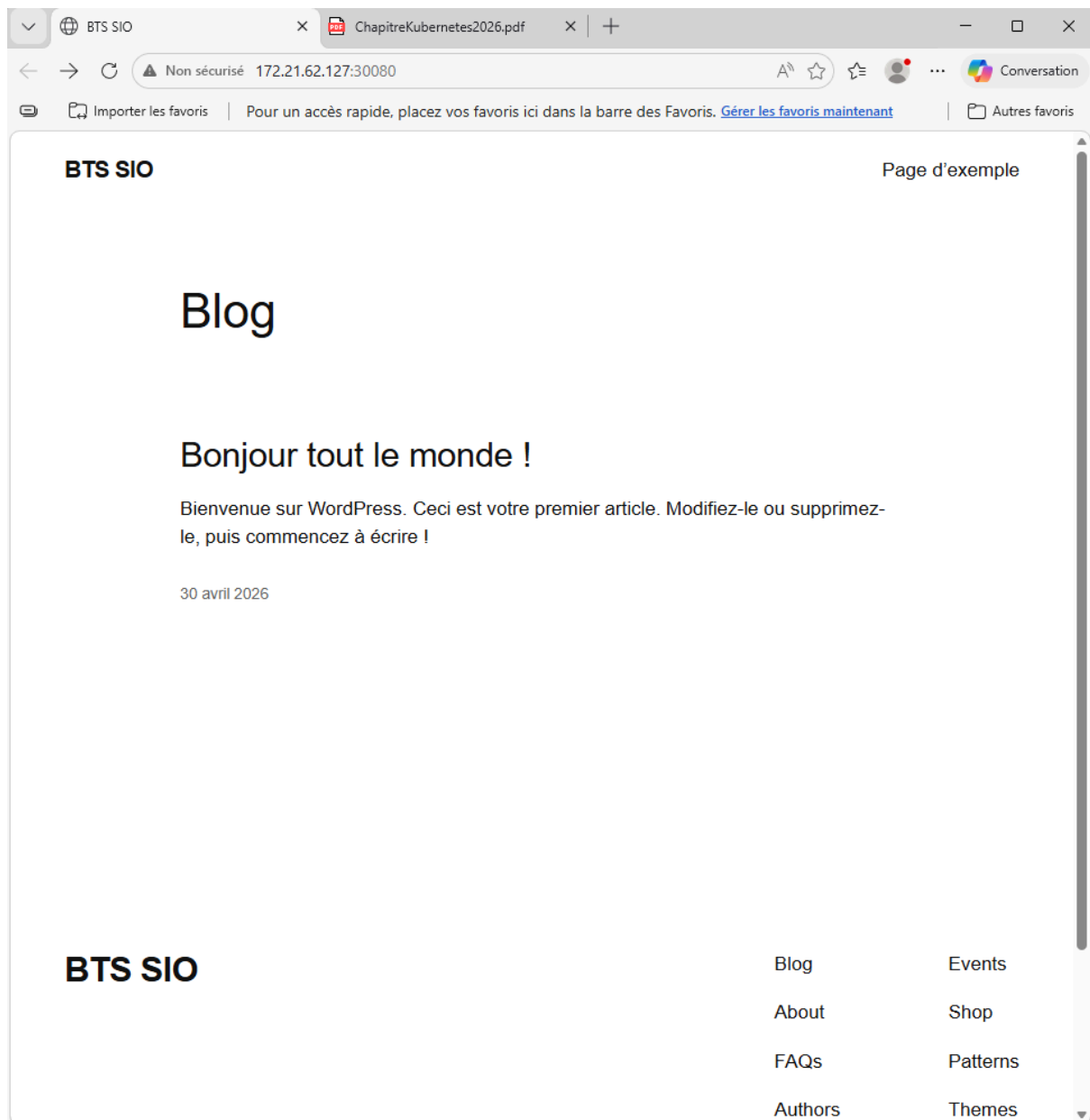
Les commandes `kubectl delete pod -l app=wordpress` et `kubectl delete pod -l app=mysql` permettent de supprimer les pods pour tester leur recréation automatique par les déploiements.

```
Administrateur : Windows Pow  Administrateur : Windows Poi  +  v
PS C:\Users\nprovenzano> kubectl delete pod -l app=wordpress
pod "wordpress-59fcc67747-kgtcl" deleted from default namespace
PS C:\Users\nprovenzano> kubectl delete pod -l app=mysql
pod "mysql-66f86944b7-jbp2v" deleted from default namespace
PS C:\Users\nprovenzano>
```

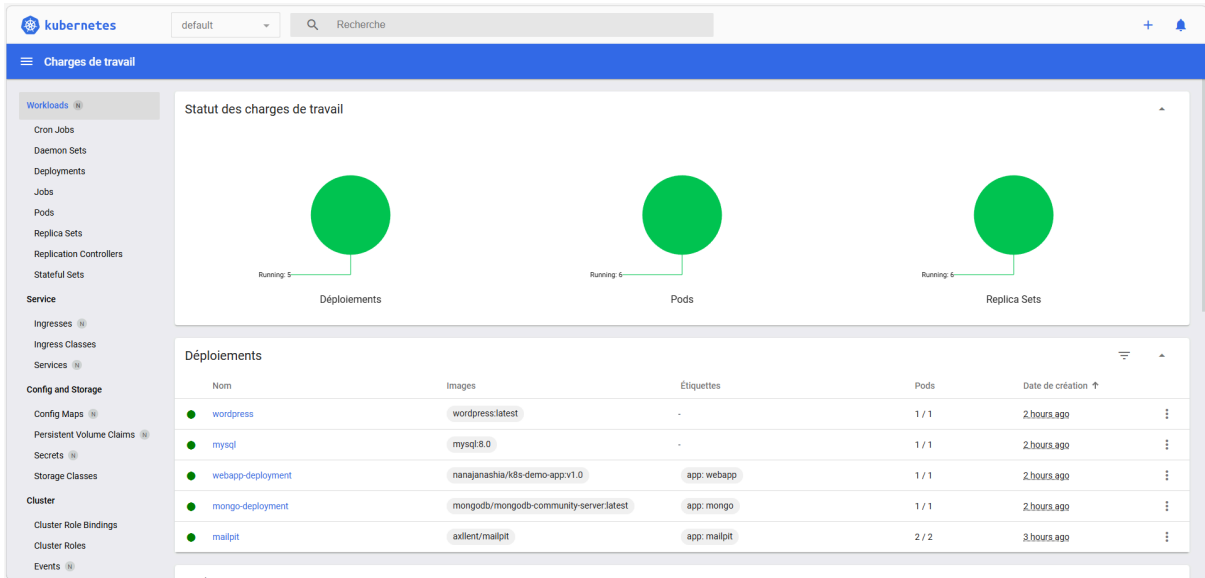
La commande `kubectl get pods` permet de confirmer que les pods WordPress et MySQL sont de nouveau en état Running grâce aux déploiements.

```
Administrateur : Windows Pow  Administrateur : Windows Poi  +  v
PS C:\Users\nprovenzano> kubectl delete pod -l app=wordpress
pod "wordpress-59fcc67747-kgtcl" deleted from default namespace
PS C:\Users\nprovenzano> kubectl delete pod -l app=mysql
pod "mysql-66f86944b7-jbp2v" deleted from default namespace
PS C:\Users\nprovenzano> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-hhn7j           1/1     Running   2 (30m ago) 3h34m
mailpit-7b96b48c5d-sphd4           1/1     Running   2 (30m ago) 3h20m
mongo-deployment-744864fdd7-n5dj1  1/1     Running   2 (29m ago) 154m
mysql-66f86944b7-bhq7              1/1     Running   0           2m17s
webapp-deployment-5766fd95c7-9vhdv 1/1     Running   2 (29m ago) 135m
wordpress-59fcc67747-qkwgt         1/1     Running   0           2m20s
PS C:\Users\nprovenzano>
```

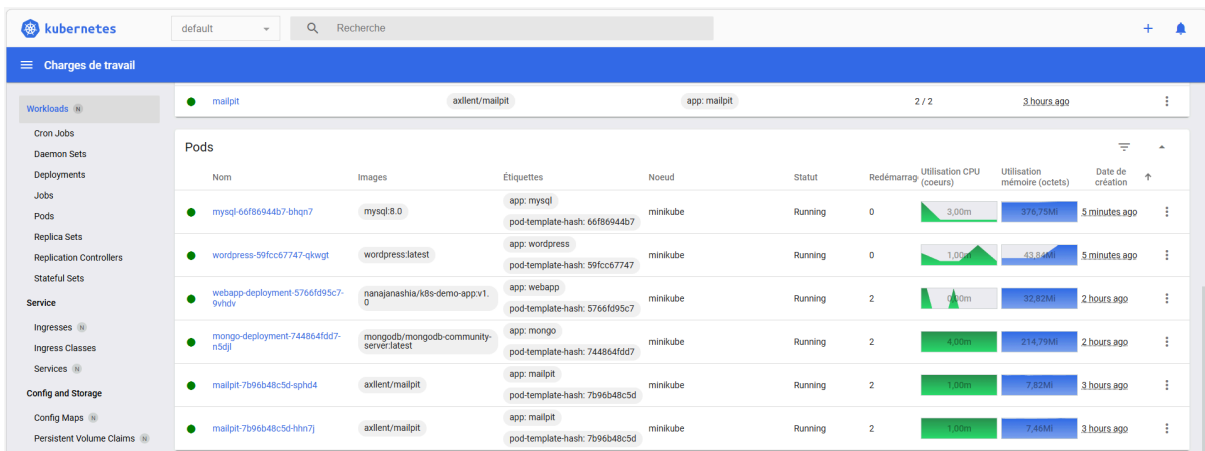
Vérification que le site WordPress est accessible depuis le navigateur et que l'application fonctionne avec l'affichage du contenu.



Le tableau de bord Kubernetes permet de vérifier que les déploiements, les pods et les ReplicaSets sont en état Running.



Affichage des pods et de leurs états :



Affichage des Replicats Sets et de leurs états :

The screenshot displays the 'Replica Sets' section in the Kubernetes dashboard. The top part shows a list of Replica Sets with columns for Name, Images, Labels, Pods, and Date de création. Below this, a detailed view of the 'wordpress-59fcc67747' Replica Set is shown, including its configuration and pod status.

| Nom                          | Images                                  | Étiquettes                                   | Pods  | Date de création |
|------------------------------|---|--|-------|------------------|
| wordpress-59fcc67747         | wordpress:latest                        | app: wordpress pod-template-hash: 59fcc67747 | 1 / 1 | 2 hours ago      |
| mysql-66f86944b7             | mysql:8.0                               | app: mysql pod-template-hash: 66f86944b7     | 1 / 1 | 2 hours ago      |
| webapp-deployment-5766f95c7  | nanajanashia/k8s-demo-app:v1.0          | app: webapp pod-template-hash: 5766f95c7     | 1 / 1 | 2 hours ago      |
| webapp-deployment-6d88954d4c | nanajanashia/k8s-demo-app:v1.0          | app: webapp pod-template-hash: 6d88954d4c    | 0 / 0 | 2 hours ago      |
| mongo-deployment-744864fd7   | mongodb/mongodb-community-server:latest | app: mongo pod-template-hash: 744864fd7      | 1 / 1 | 2 hours ago      |
| mailpit-7b96b48c5d           | axllent/mailpit                         | app: mailpit pod-template-hash: 7b96b48c5d   | 2 / 2 | 3 hours ago      |

Affichage des Services :

The screenshot displays the 'Services' section in the Kubernetes dashboard. The top part shows a list of Services with columns for Name, Labels, Type, IP cluster, Terminaisons internes, Terminaisons externes, and Date de création. Below this, a detailed view of the 'mysql' service is shown, including its configuration and pod status.

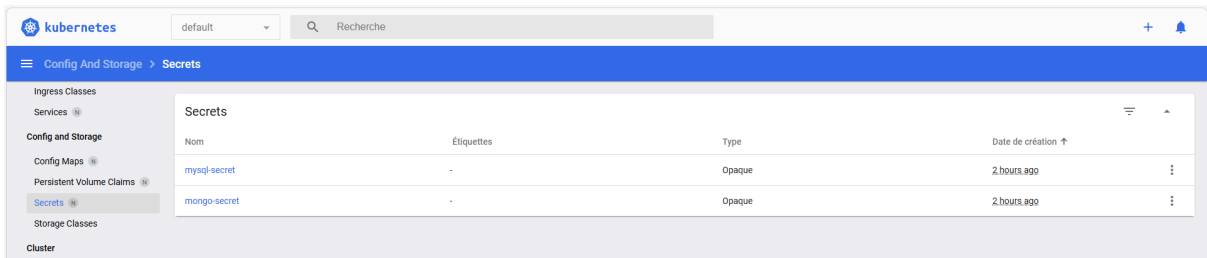
| Nom            | Étiquettes                                   | Type      | IP cluster     | Terminaisons internes  | Terminaisons externes | Date de création |
|----------------|--|-----------|----------------|--|-----------------------|------------------|
| mysql          | -  | ClusterIP | 10.107.152.182 | mysql-3306 TCP<br>mysql-0 TCP  | -                     | 2 hours ago      |
| wordpress      | -  | NodePort  | 10.108.202.173 | wordpress-80 TCP<br>wordpress-30080 TCP                                | -                     | 2 hours ago      |
| webapp-service | -  | NodePort  | 10.107.105.109 | webapp-service:3000 TCP<br>webapp-service:30100 TCP                    | -                     | 2 hours ago      |
| mongo-service  | -  | ClusterIP | 10.100.26.88   | mongo-service:27017 TCP<br>mongo-service:0 TCP                         | -                     | 2 hours ago      |
| mailpit        | app: mailpit                                 | ClusterIP | 10.96.177.29   | mailpit:1025 TCP<br>mailpit:0 TCP<br>mailpit:8025 TCP<br>mailpit:0 TCP | -                     | 3 hours ago      |
| kubernetes     | component: apiserver<br>provider: kubernetes | ClusterIP | 10.96.0.1      | kubernetes:443 TCP<br>kubernetes:0 TCP                                 | -                     | 4 hours ago      |

Affichage des ConfigMaps :

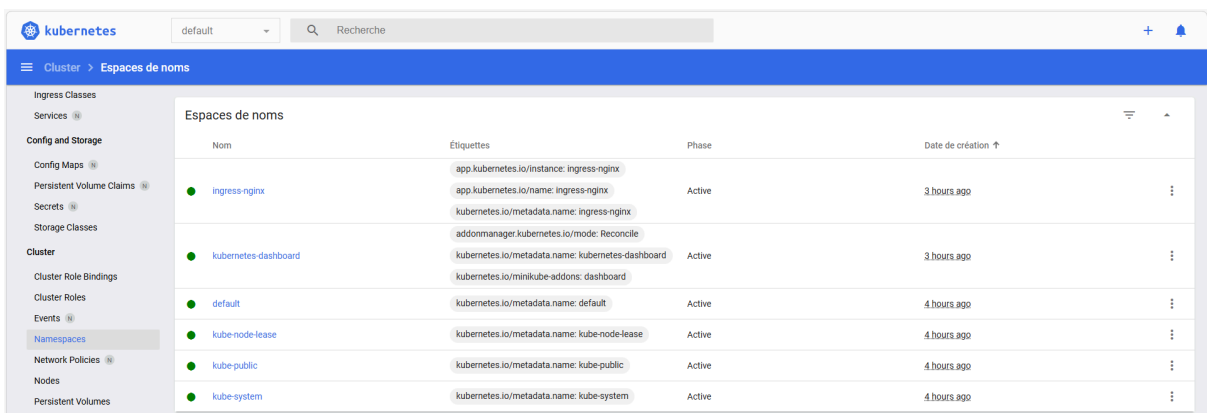
The screenshot displays the 'Config Maps' section in the Kubernetes dashboard. The top part shows a list of Config Maps with columns for Name, Labels, and Date de création. Below this, a detailed view of the 'mongo-config' Config Map is shown, including its configuration and pod status.

| Nom              | Étiquettes | Date de création |
|------------------|------------|------------------|
| mongo-config     | -          | 2 hours ago      |
| kube-root-ca.crt | -          | 4 hours ago      |

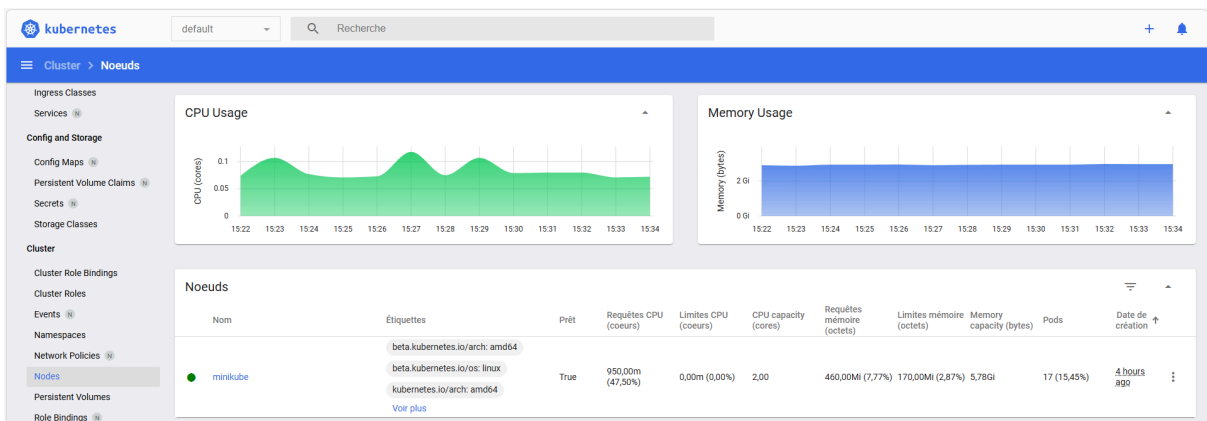
Affichage des Secrets :



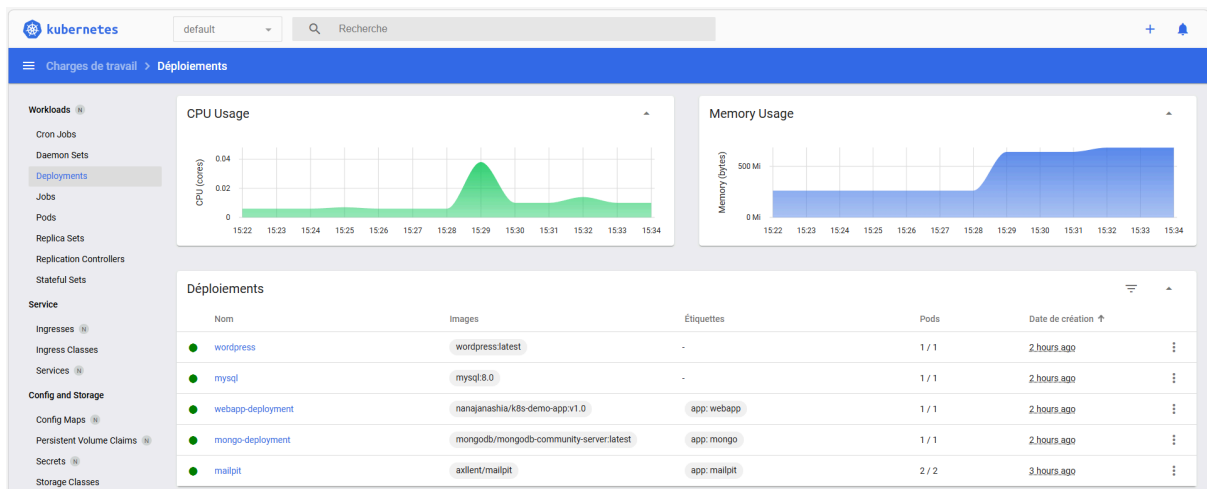
Affichage des Namespaces :



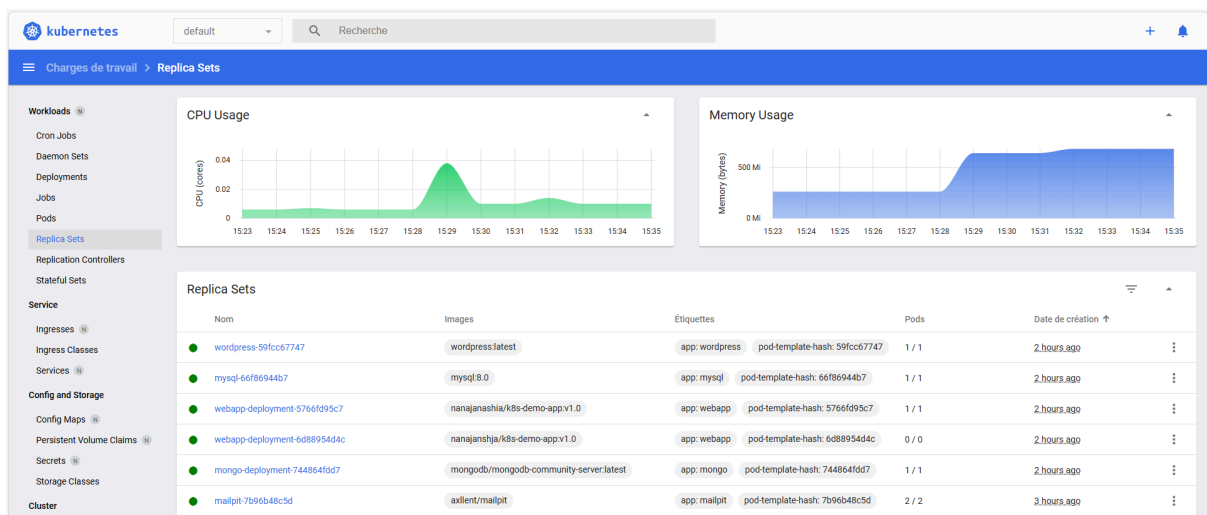
Affichage du Node (noeud) en cours d'exécution :



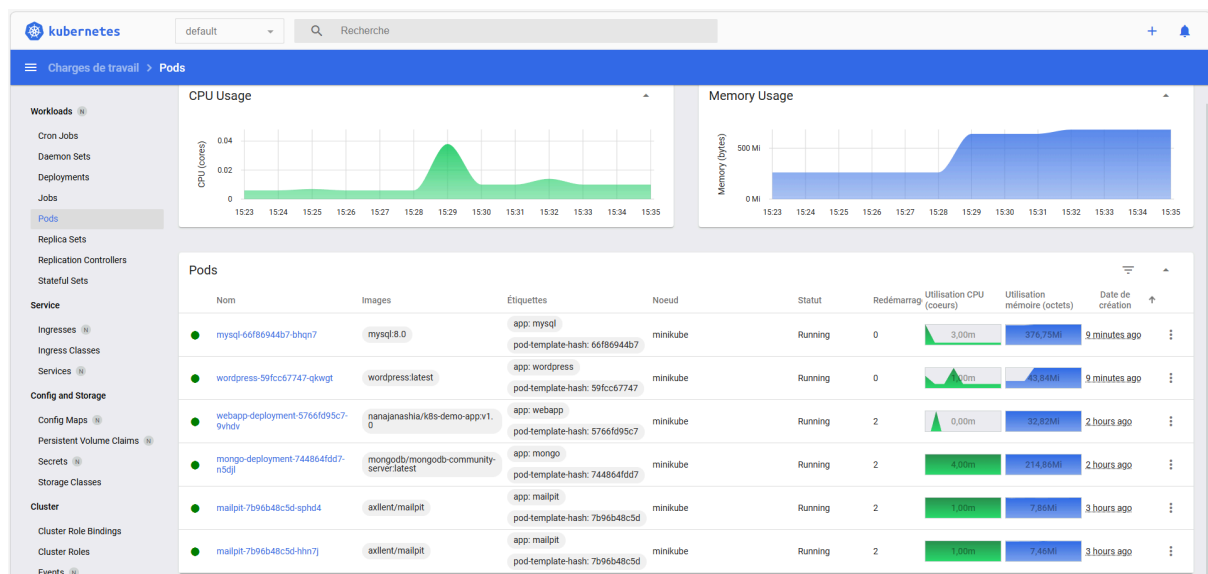
Affichage des Déploiements en cours d'exécution :



Affichage des ReplicaSets en cours d'exécution :



Affichage des Pods en cours d'exécution :

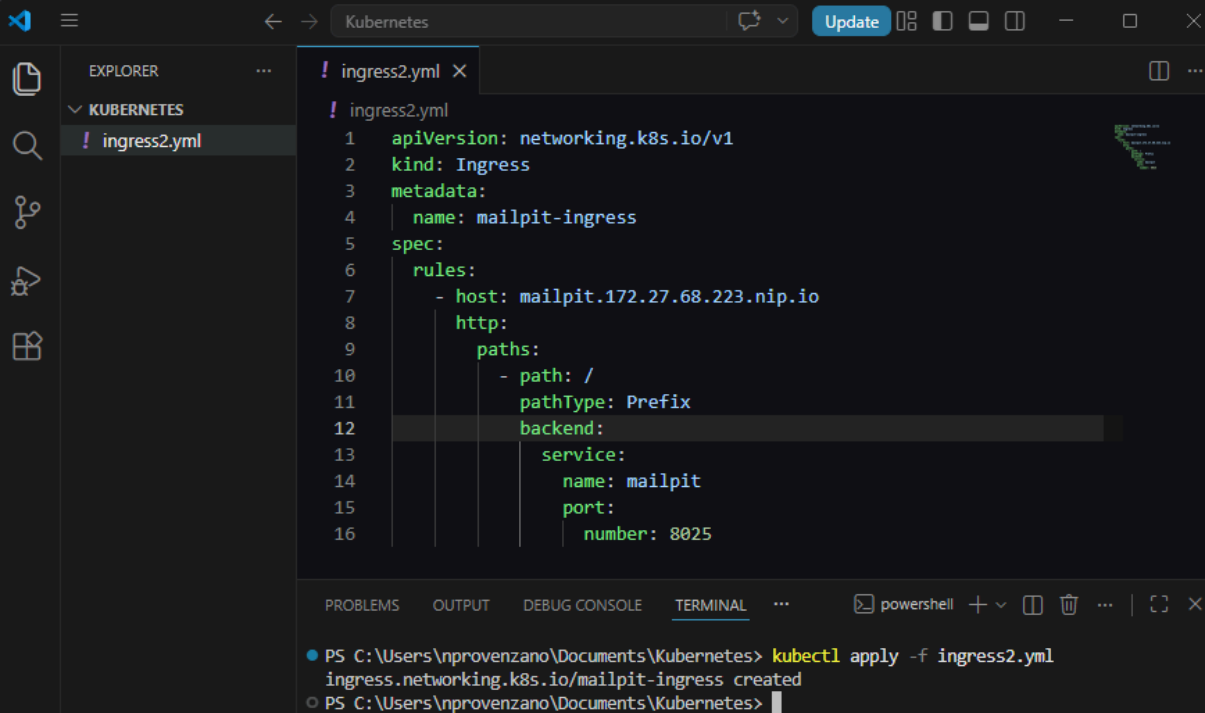


## 7. Suite Ingress (Mailpit) : hôtes virtuels et nom de domaine nip.io

La commande

```
PS C:\Users\nprovenzano> kubectl get ingress
NAME          CLASS  HOSTS  ADDRESS  PORTS  AGE
test-ingress  nginx  *      172.28.25.20  80     6d21h
PS C:\Users\nprovenzano> minikube ip
172.28.25.20
PS C:\Users\nprovenzano> |
```

Fichier ingress2.yml

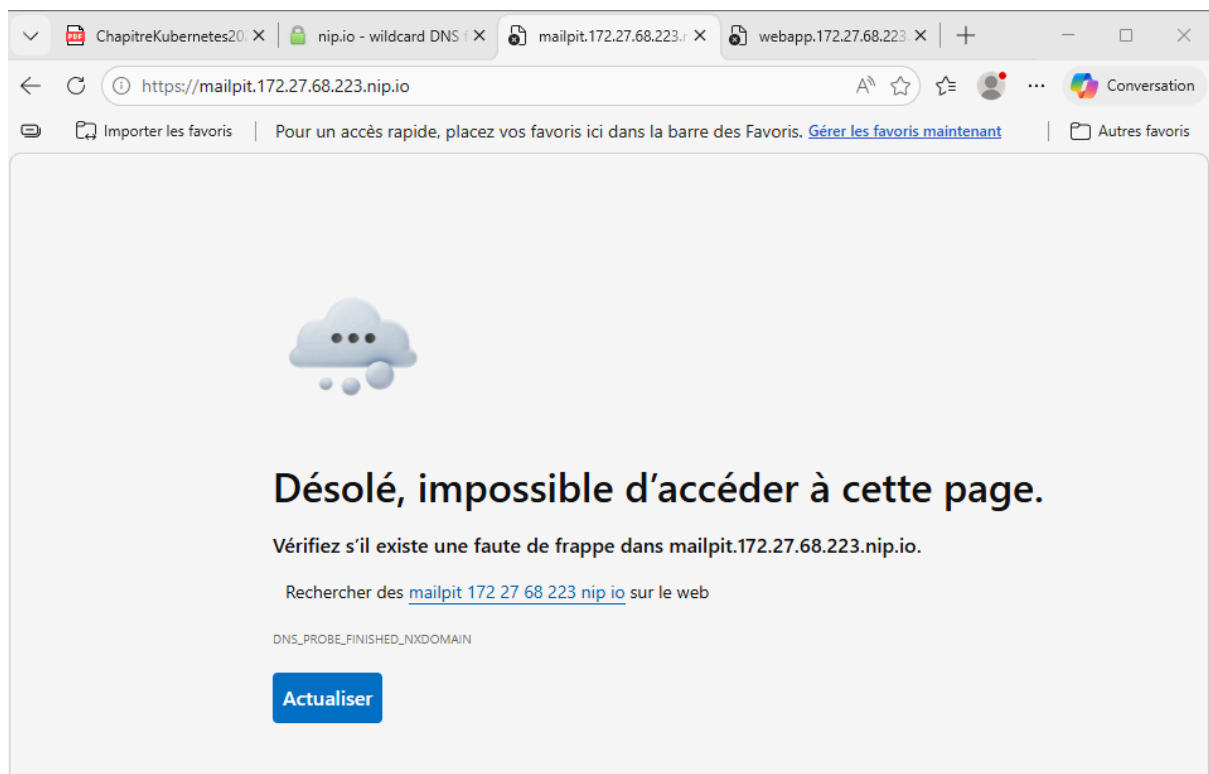


```
1  apiVersion: networking.k8s.io/v1
2  kind: Ingress
3  metadata:
4    name: mailpit-ingress
5  spec:
6    rules:
7      - host: mailpit.172.27.68.223.nip.io
8        http:
9          paths:
10           - path: /
11             pathType: Prefix
12             backend:
13               service:
14                 name: mailpit
15                 port:
16                   number: 8025
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... powershell + -

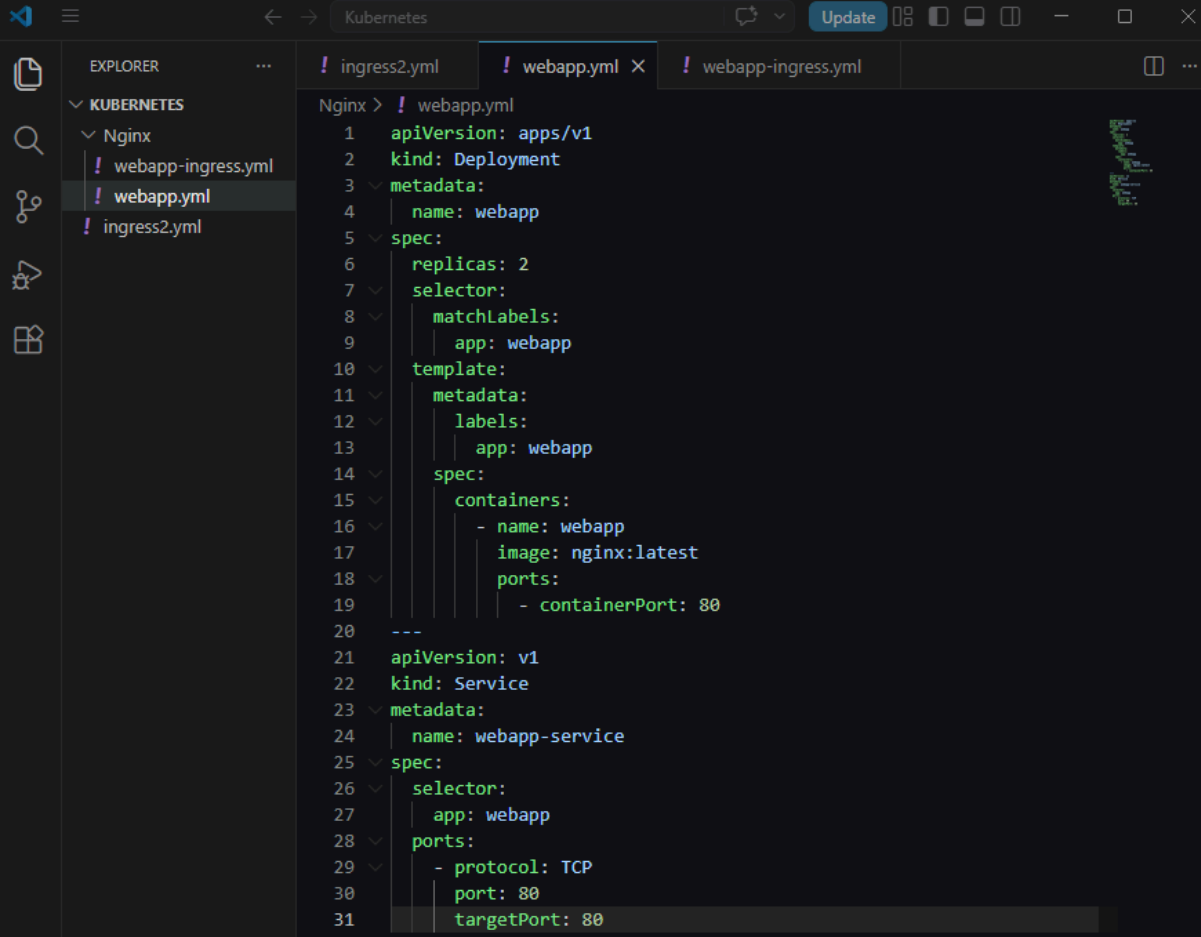
```
PS C:\Users\nprovenzano\Documents\Kubernetes> kubectl apply -f ingress2.yml
ingress.networking.k8s.io/mailpit-ingress created
PS C:\Users\nprovenzano\Documents\Kubernetes>
```

Le nom de domaine [.nip.io](https://nip.io) étant bloqué par le serveur du lycée, la redirection ne s'est pas pu se faire.



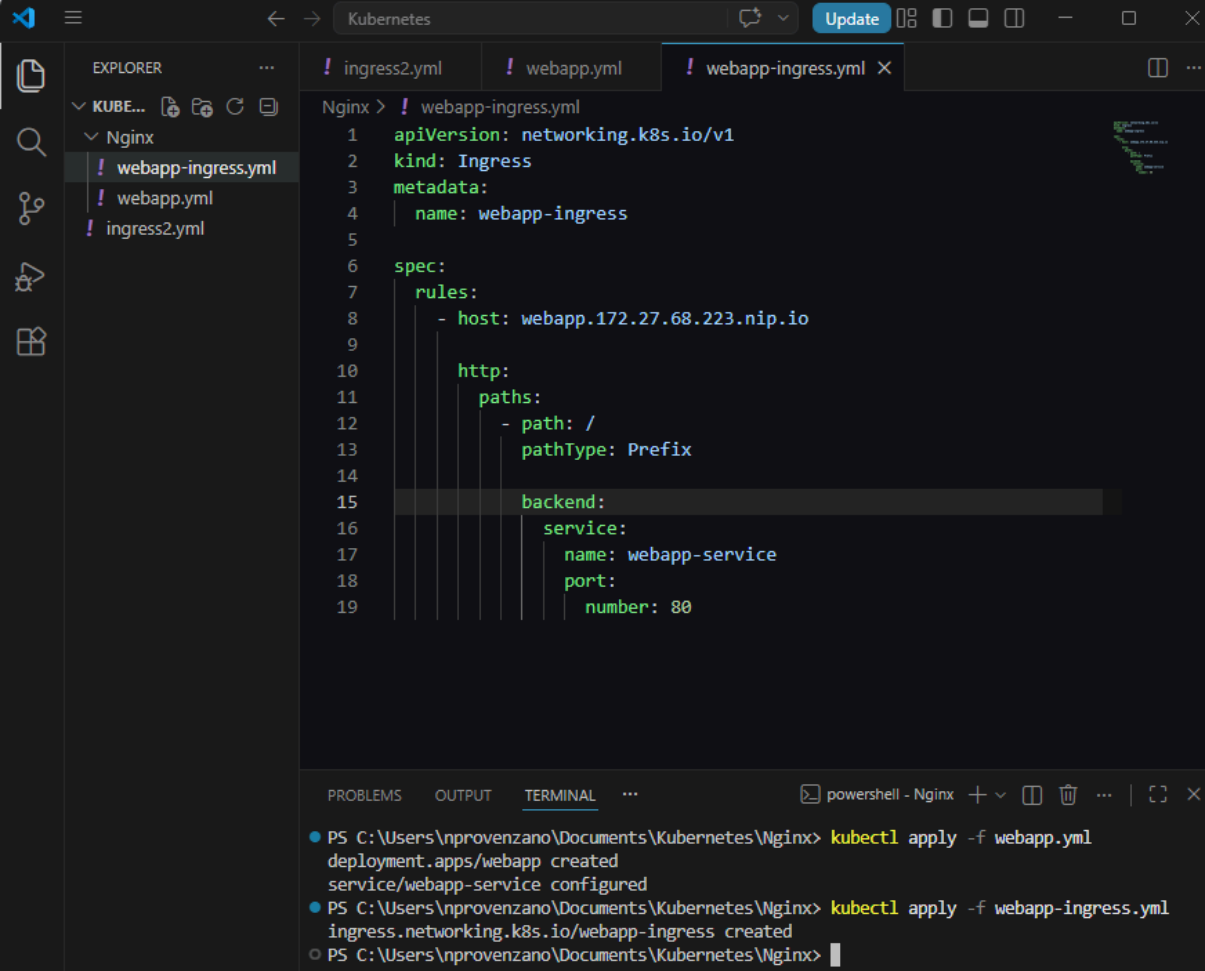
## 8. Hôtes virtuels et nom de domaine nip.io : autre exemple

Fichier webapp.yml



```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: webapp
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: webapp
10   template:
11     metadata:
12       labels:
13         app: webapp
14     spec:
15       containers:
16         - name: webapp
17           image: nginx:latest
18           ports:
19             - containerPort: 80
20   ---
21  apiVersion: v1
22  kind: Service
23  metadata:
24    name: webapp-service
25  spec:
26    selector:
27      app: webapp
28    ports:
29      - protocol: TCP
30        port: 80
31        targetPort: 80
```

Fichier webapp-ingress.yml



The screenshot shows the Visual Studio Code editor with the file explorer on the left and the code editor in the center. The file explorer shows the following structure:

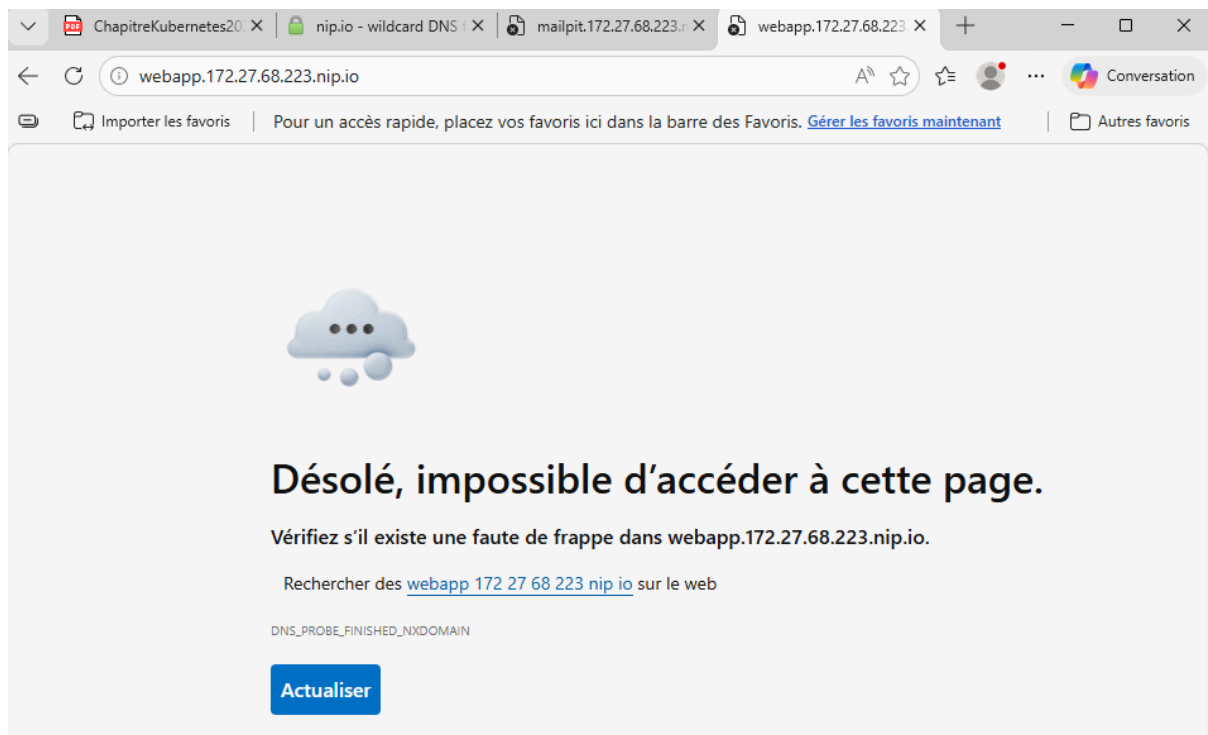
- KUBE...
  - Nginx
    - webapp-ingress.yml
    - webapp.yml
    - ingress2.yml

The code editor displays the content of `webapp-ingress.yml`:

```
1  apiVersion: networking.k8s.io/v1
2  kind: Ingress
3  metadata:
4    name: webapp-ingress
5
6  spec:
7    rules:
8      - host: webapp.172.27.68.223.nip.io
9
10     http:
11       paths:
12         - path: /
13           pathType: Prefix
14
15       backend:
16         service:
17           name: webapp-service
18           port:
19             number: 80
```

The terminal at the bottom shows the following commands and output:

```
PS C:\Users\provenzano\Documents\Kubernetes\Nginx> kubectl apply -f webapp.yml
deployment.apps/webapp created
service/webapp-service configured
PS C:\Users\provenzano\Documents\Kubernetes\Nginx> kubectl apply -f webapp-ingress.yml
ingress.networking.k8s.io/webapp-ingress created
PS C:\Users\provenzano\Documents\Kubernetes\Nginx>
```



## 9. Cycle de vie d'un conteneur dans Kubernetes

La commande `kubectl get pods -l app=mailpit` permet d'afficher les pods de l'application Mailpit.

```
Administrateur : Windows Po... x + v
PS C:\Users\nprovenzano> kubectl get pods -l app=mailpit
NAME                                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-hhn7j            1/1     Running   3 (43m ago) 6d22h
mailpit-7b96b48c5d-sphd4            1/1     Running   3 (43m ago) 6d22h
PS C:\Users\nprovenzano> |
```

La commande `kubectl exec -it deployment/mailpit -- sh` permet d'accéder au terminal du conteneur Mailpit.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl exec -it deployment/mailpit -- sh
/ # ps -ef
PID    USER      TIME  COMMAND
   1   root         0:00  /mailpit
  30   root         0:00  sh
  36   root         0:00  ps -ef
/ # mkdir /tmp/test
/ # ls -ld /tmp/test
drwxr-xr-x  2 root      root      4096 May  7 09:00 /tmp/test
/ # kill 1
/ # command terminated with exit code 137
PS C:\Users\nprovenzano>
```

La commande `kubectl get pods -l app=mailpit` permet d'afficher les pods de l'application Mailpit et leur état.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl get pods -l app=mailpit
NAME                                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-hhn7j           1/1     Running   3 (49m ago) 6d23h
mailpit-7b96b48c5d-sphd4           1/1     Running   4 (27s ago) 6d22h
PS C:\Users\nprovenzano> |
```

La commande `kubectl exec -it deployment/mailpit -- sh` permet d'accéder au terminal du conteneur Mailpit.

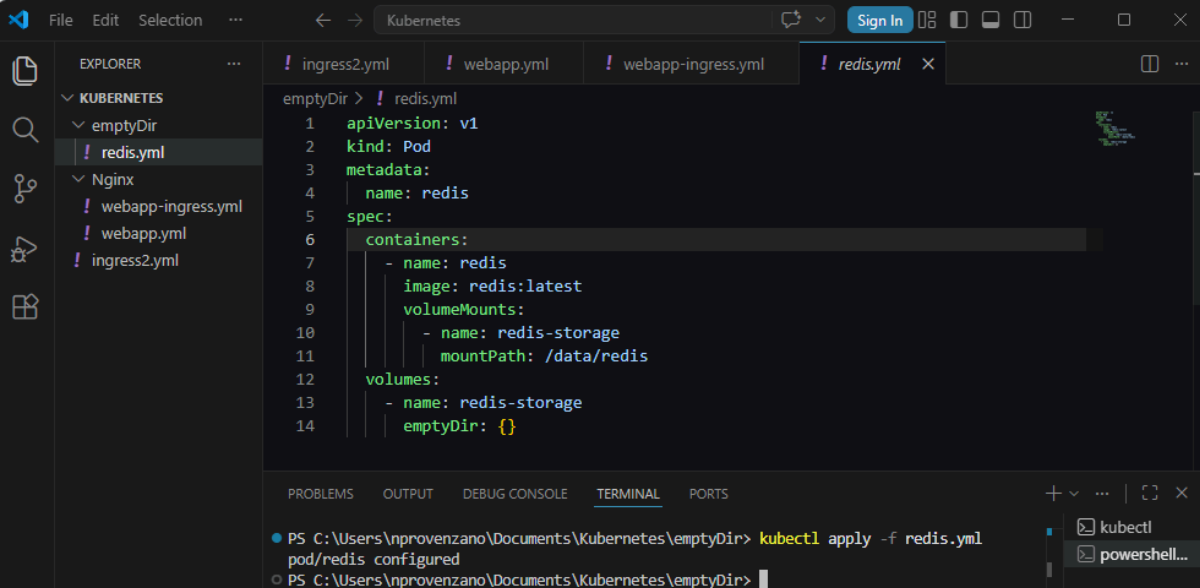
L'absence du répertoire `/tmp/test` après le redémarrage du conteneur montre que les données non persistantes ont été supprimées.

```
Administrateur : Windows Po1 x + v
PS C:\Users\nprovenzano> kubectl exec -it deployment/mailpit -- sh
/ # ls -ld /tmp/test
ls: /tmp/test: No such file or directory
/ # exit
command terminated with exit code 1
PS C:\Users\nprovenzano> |
```

## 10. Persistence des données

### 1) Volume de type emptyDir

Le fichier **redis.yml** permet de créer un pod Redis utilisant un volume de type emptyDir. La commande **kubectl apply -f redis.yml** permet de créer le pod Redis.

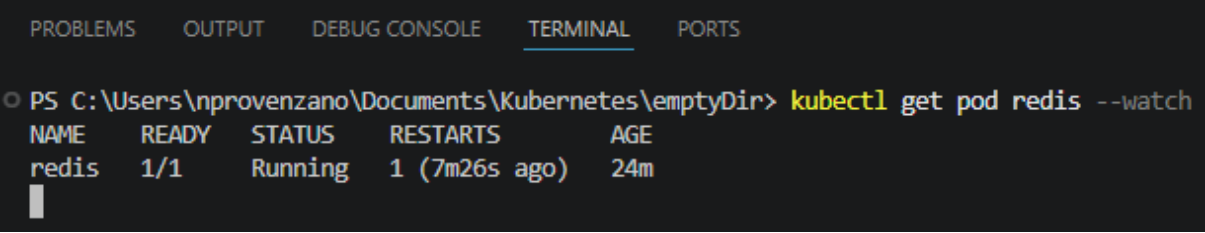


The screenshot shows the Visual Studio Code editor with the **redis.yml** file open. The file content is as follows:

```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: redis
5 spec:
6   containers:
7     - name: redis
8       image: redis:latest
9       volumeMounts:
10        - name: redis-storage
11          mountPath: /data/redis
12   volumes:
13     - name: redis-storage
14       emptyDir: {}
```

The terminal window at the bottom shows the command **kubectl apply -f redis.yml** being executed, resulting in the output: **pod/redis configured**.

La commande **kubectl get pod redis --watch** permet de surveiller en temps réel l'état du pod Redis.



The terminal window shows the command **kubectl get pod redis --watch** being executed, resulting in the following output:

| NAME  | READY | STATUS  | RESTARTS      | AGE |
|-------|-------|---------|---------------|-----|
| redis | 1/1   | Running | 1 (7m26s ago) | 24m |

La commande **kubectl exec -it redis -- /bin/bash** permet d'accéder au terminal du conteneur Redis.

La création du fichier **test-file** dans **/data/redis** permet de tester la persistance du volume `emptyDir`.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\nprovenzano\Documents\Kubernetes\emptyDir> kubectl exec -it redis -- /bin/bash
root@redis:/data# cd /data/redis/
root@redis:/data/redis# echo Hello > test-file
root@redis:/data/redis#
```

La commande **apt-get update** permet de mettre à jour la liste des paquets du conteneur Redis.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
root@redis:/data/redis# 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 (7056 kB/s)
Reading package lists... Done
root@redis:/data/redis#
```

La commande **apt-get install procs** permet d'installer les outils nécessaires pour afficher les processus du conteneur.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

root@redis:/data/redis# apt-get install procs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 libncursesw6 libproc2-0 linux-sysctl-defaults psmisc
Suggested packages:
  gpm
The following NEW packages will be installed:
  libgpm2 libncursesw6 libproc2-0 linux-sysctl-defaults procs psmisc
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 1370 kB of archives.
After this operation, 4163 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

La commande **ps aux** permet d'afficher les processus en cours d'exécution dans le conteneur Redis.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

root@redis:/data/redis# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
redis     1  0.0  0.3 160376 23296 ?        Ss1   09:22   0:00 redis-server *:6379
root      29  0.0  0.0   4324  3456 pts/0    Ss   09:31   0:00 /bin/bash
root      70  0.0  0.0     0     0 pts/0    Z    09:33   0:00 [dpkg-preconfigu] <defunct>
root     160  0.0  0.0   6388  3584 pts/0    R+   09:33   0:00 ps aux
root@redis:/data/redis#
```

La commande **kill 1** permet d'arrêter le processus principal du conteneur Redis, provoquant son redémarrage automatique.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

root@redis:/data/redis# kill 1
root@redis:/data/redis# command terminated with exit code 137
○ PS C:\Users\nprovenzano\Documents\Kubernetes\emptyDir>
```

Le pod Redis redémarre automatiquement après l'arrêt du processus principal du conteneur.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
○ PS C:\Users\nprovenzano\Documents\Kubernetes> kubectl get pod redis --watch
NAME     READY   STATUS    RESTARTS   AGE
redis    1/1     Running   1 (22s ago) 17m
redis    0/1     Completed 1 (11m ago) 27m
redis    1/1     Running   2 (2s ago)  27m
█
```

Le pod Redis redémarre automatiquement après l'arrêt du processus principal du conteneur.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\emptyDir> kubectl exec -it redis -- /bin/bash
root@redis:/data# cd /data/redis
root@redis:/data/redis# ls
test-file
root@redis:/data/redis# exit
exit
○ PS C:\Users\nprovenzano\Documents\Kubernetes\emptyDir> █
```

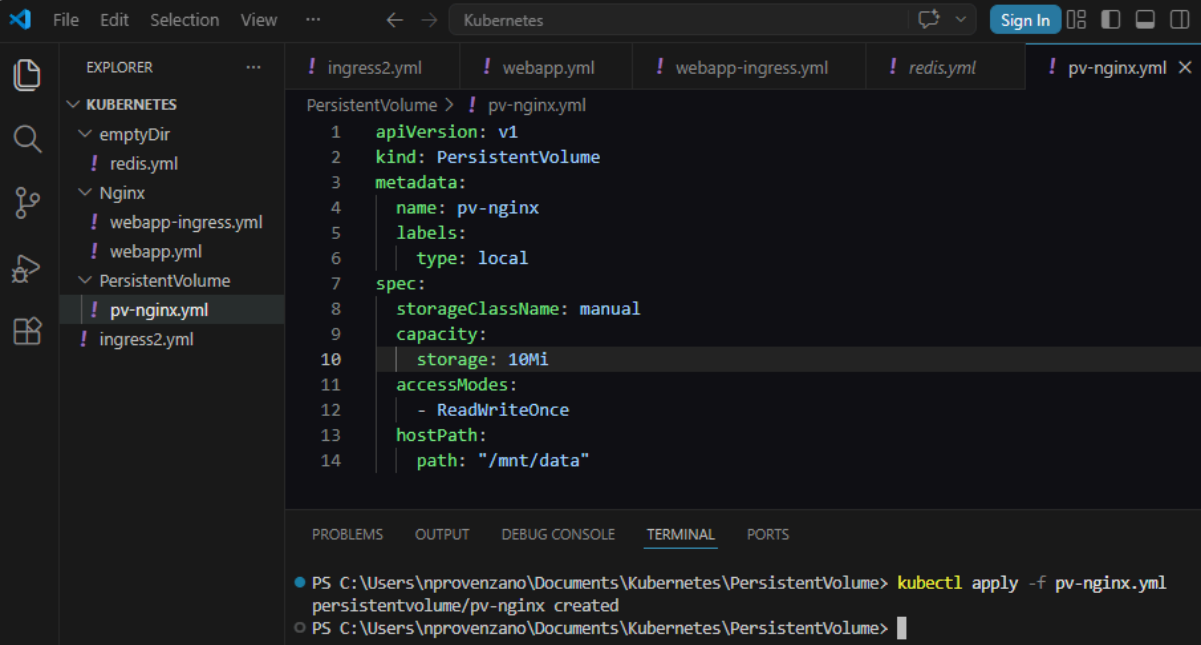
La commande **kubectl delete pod redis** permet de supprimer le pod Redis du cluster Kubernetes.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\emptyDir> kubectl delete pod redis
pod "redis" deleted from default namespace
○ PS C:\Users\nprovenzano\Documents\Kubernetes\emptyDir> █
```



Le fichier **pv-nginx.yml** permet de créer un PersistentVolume de type hostPath pour le stockage persistant.

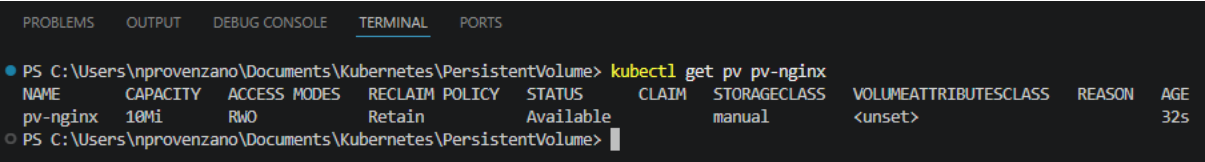
La commande **kubectl apply -f pv-nginx.yml** permet de créer le PersistentVolume pv-nginx.



```
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    name: pv-nginx
5    labels:
6      type: local
7  spec:
8    storageClassName: manual
9    capacity:
10   storage: 10Mi
11   accessModes:
12     - ReadWriteOnce
13   hostPath:
14     path: "/mnt/data"
```

```
PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> kubectl apply -f pv-nginx.yml
persistentvolume/pv-nginx created
PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume>
```

La commande **kubectl get pv pv-nginx** permet de vérifier que le PersistentVolume est créé et disponible.



```
PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> kubectl get pv pv-nginx
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM  STORAGECLASS  VOLUMEATTRIBUTESCLASS  REASON  AGE
pv-nginx      10Mi     RWO           Retain          Available  manual  <unset>      <unset>                 <unset> 32s
PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume>
```

Le fichier `pvc-nginx.yml` permet de créer un PersistentVolumeClaim associé au volume persistant `pv-nginx`.

La commande `kubectl apply -f pvc-nginx.yml` permet de créer le PersistentVolumeClaim `task-pv-claim`.

```

1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: task-pv-claim
5  spec:
6    storageClassName: manual
7    accessModes:
8      - ReadWriteOnce
9    resources:
10     requests:
11       storage: 10Mi
12     volumeName: pv-nginx

```

```

PS C:\Users\provenzano\Documents\Kubernetes\PersistentVolume> kubectl apply -f pvc-nginx.yml
persistentvolumeclaim/task-pv-claim created
PS C:\Users\provenzano\Documents\Kubernetes\PersistentVolume>

```

Le PersistentVolume `pv-nginx` est maintenant associé au PersistentVolumeClaim `task-pv-claim` avec le statut `Bound`.

```

PS C:\Users\provenzano\Documents\Kubernetes\PersistentVolume> kubectl get pv pv-nginx
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM                STORAGECLASS  VOLUMEATTRIBUTESCLASS  REASON  AGE
pv-nginx      10Mi      RWO           Retain          Bound   default/task-pv-claim  manual         <unset>                 <empty> 3m54s
PS C:\Users\provenzano\Documents\Kubernetes\PersistentVolume>

```

La commande `kubectl get pvc` permet d'afficher les PersistentVolumeClaims et leurs états dans le cluster Kubernetes.

```

PS C:\Users\provenzano\Documents\Kubernetes\PersistentVolume> kubectl get pvc
NAME          STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
mysql-pvc     Bound   pvc-5f49d2b5-9cad-49b2-bd60-3da2e09ac5c8  2Gi       RWO           standard      <unset>                 6d22h
task-pv-claim Bound   pv-nginx                                   10Mi      RWO           manual        <unset>                 108s
wordpress-pvc Bound   pvc-15b23d24-7525-4ae4-895a-b57dd439c6e9  2Gi       RWO           standard      <unset>                 6d22h
PS C:\Users\provenzano\Documents\Kubernetes\PersistentVolume>

```

Le fichier **pvc-pod.yml** permet de créer un pod Nginx utilisant le PersistentVolumeClaim **task-pv-claim**.

La commande **kubectl apply -f pvc-pod.yml** permet de créer le pod **task-pv-pod**.

The screenshot shows the Visual Studio Code editor with the file **pvc-pod.yml** open. The file content is as follows:

```

1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: task-pv-pod
5  spec:
6    volumes:
7      - name: task-pv-storage
8        persistentVolumeClaim:
9          claimName: task-pv-claim
10   containers:
11     - name: task-pv-container
12       image: nginx:latest
13       ports:
14         - containerPort: 80
15           name: "http-server"
16       volumeMounts:
17         - mountPath: "/usr/share/nginx/html"
18           name: task-pv-storage

```

The terminal at the bottom shows the execution of the **kubectl apply -f pvc-pod.yml** command, resulting in the pod **task-pv-pod** being created.

```

PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> kubectl apply -f pvc-pod.yml
pod/task-pv-pod created
PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume>

```

La commande **kubectl get pod task-pv-pod** permet de vérifier que le pod est en état **Running**.

La commande **kubectl describe pod task-pv-pod** permet d'afficher les informations détaillées du pod.

The screenshot shows the terminal output of the **kubectl get** and **kubectl describe** commands for the pod **task-pv-pod**.

```

PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> kubectl get pod task-pv-pod
NAME          READY   STATUS    RESTARTS   AGE
task-pv-pod   1/1     Running   0           112s
PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> kubectl describe pod task-pv-pod
Name:          task-pv-pod
Namespace:     default
Priority:       0
Service Account: default
Node:          minikube/172.28.25.20
Start Time:    Thu, 07 May 2026 11:49:48 +0200
Labels:        <none>
Annotations:   <none>
Status:        Running
IP:            10.244.0.69
IPs:
IP: 10.244.0.69

```

La commande **kubectl get pod task-pv-pod** permet de vérifier que le pod Nginx est en cours d'exécution.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> kubectl get pod task-pv-pod
NAME          READY  STATUS   RESTARTS  AGE
task-pv-pod   1/1    Running  0          3m7s
○ PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> |
```

La commande **kubectl describe pod task-pv-pod** permet d'afficher les détails du pod ainsi que le volume persistant utilisé.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> kubectl describe pod task-pv-pod
Name:          task-pv-pod
Namespace:     default
Priority:       0
Service Account: default
Node:          minikube/172.28.25.20
Start Time:    Thu, 07 May 2026 11:49:48 +0200
Labels:        <none>
Annotations:   <none>
Status:        Running
IP:            10.244.0.69
IPs:
IP: 10.244.0.69
Containers:
  task-pv-container:
    Container ID:  docker://795ecd8773d1f2d08160e1d6135fb00a7442d43c8043c9d50b137aa30c873d50
    Image:         nginx:latest
    Image ID:      docker-pullable://nginx@sha256:6e23479198b998e5e25921dff8455837c7636a67111a04a635cf1bb363d19
9dc
    Port:          80/TCP (http-server)
    Host Port:     0/TCP (http-server)
    State:         Running
      Started:     Thu, 07 May 2026 11:49:50 +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-c9pbw (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 namespace)
    ClaimName:     task-pv-claim
```

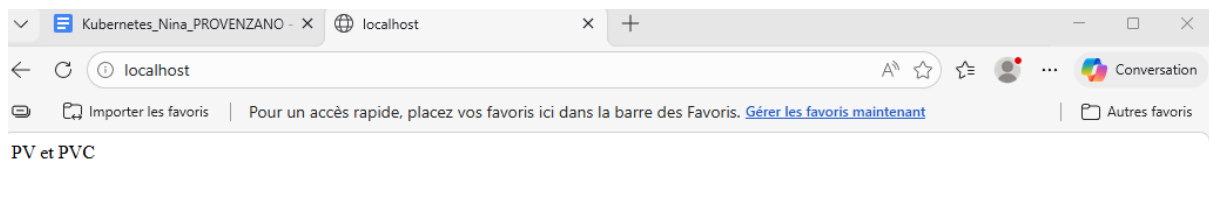
La commande `kubectl exec -it task-pv-pod -- /bin/bash` permet d'accéder au terminal du conteneur Nginx.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> kubectl exec -it task-pv-pod -- /bin/bash
root@task-pv-pod:/#
```

La commande `kubectl port-forward task-pv-pod 80` permet d'accéder au serveur Nginx du pod depuis localhost.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> 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
█
```

L'accès à localhost permet de vérifier que le contenu stocké dans le volume persistant est bien affiché par le serveur Nginx.



La commande `curl http://localhost` permet de vérifier que le serveur Nginx affiche le contenu du volume persistant.

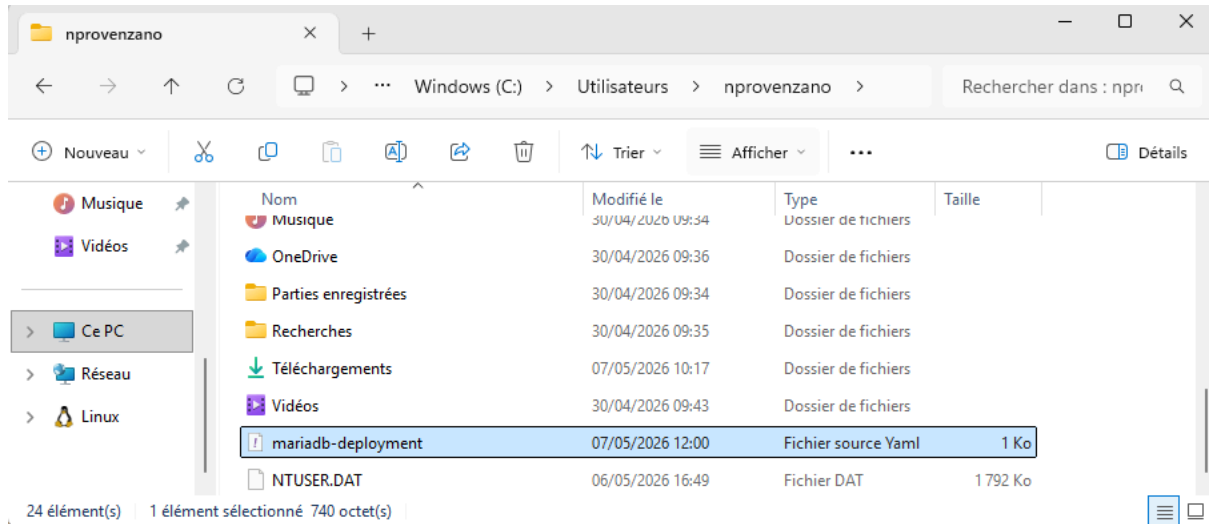
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\nprovenzano\Documents\Kubernetes\PersistentVolume> kubectl exec -it task-pv-pod -- /bin/bash
root@task-pv-pod:/# apt-get update
Get:1 http://deb.debian.org/debian trixie InRelease [140 kB]
Get:2 http://deb.debian.org/debian trixie-updates 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 (7006 kB/s)
Reading package lists... Done
root@task-pv-pod:/# apt-get install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.14.1-2+deb13u2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@task-pv-pod:/# curl http://localhost
PV et PVC
root@task-pv-pod:/#
```

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

La commande `kubectl create deployment mariadb --image=mariadb:latest --dry-run=client --output yml > mariadb-deployment.yml` permet de générer automatiquement le fichier YAML du déploiement MariaDB.

```
Administrateur : Windows Pow  x  Administrateur : Windows Poi  x  +  -  □  ×
PS C:\Users\nprovenzano> kubectl create deployment mariadb --image=mariadb:latest --dry-run=client --output yml > maria
db-deployment.yml
PS C:\Users\nprovenzano> |
```

Le fichier **mariadb-deployment.yaml** contient la configuration du déploiement MariaDB générée automatiquement par Kubernetes.



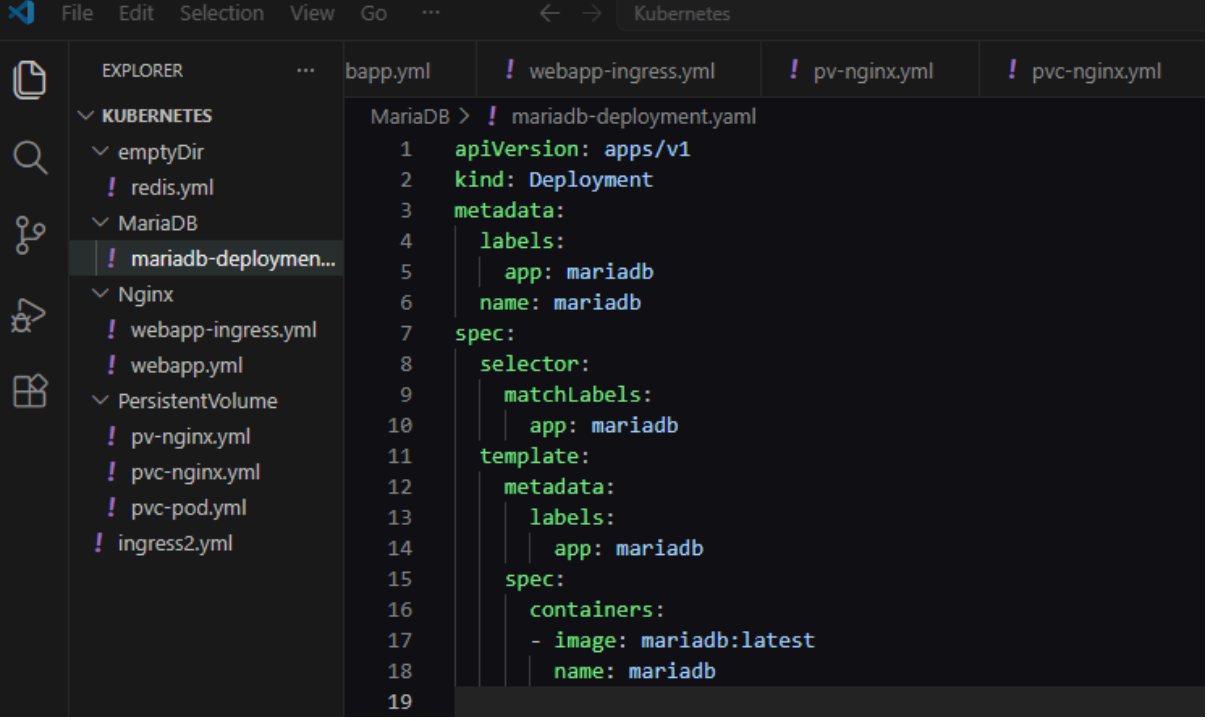
Le fichier **mariadb-deployment.yaml** contient la configuration du déploiement MariaDB avec l'image Docker mariadb:latest.

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    labels:
5      app: mariadb
6    name: mariadb
7  spec:
8    replicas: 1
9    selector:
10   matchLabels:
11     app: mariadb
12   strategy: {}
13   template:
14     metadata:
15       labels:
16         app: mariadb
17     spec:
18       containers:
19         - image: mariadb:latest
20           name: mariadb
21           resources: {}
22   status: {}
23

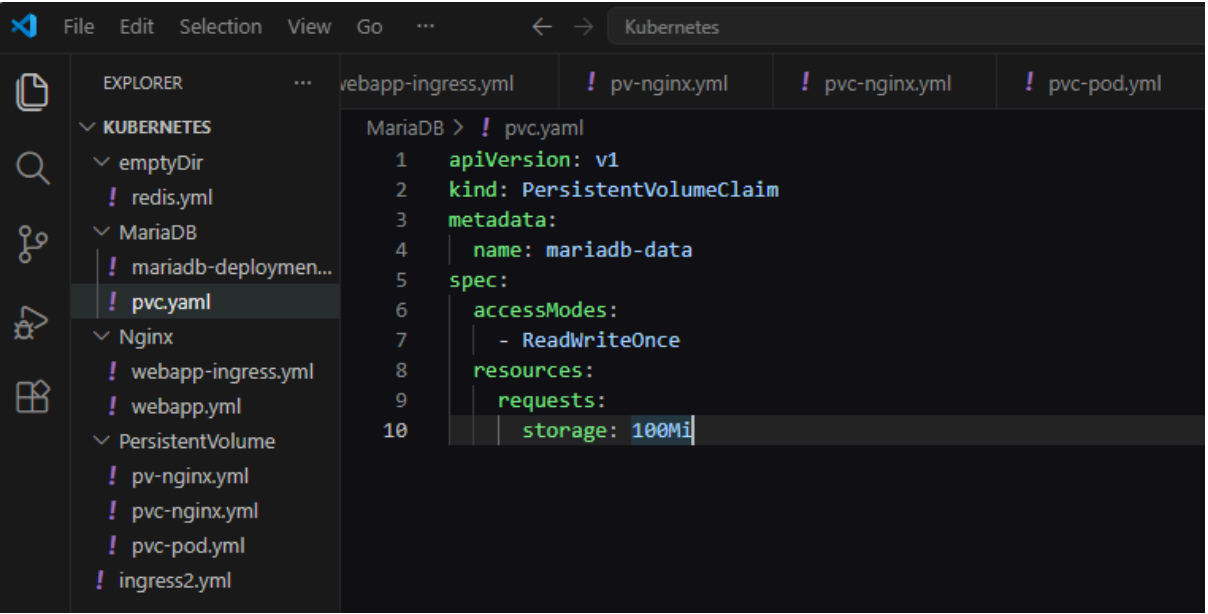
```

La modification du fichier **mariadb-deployment.yaml** permet de simplifier la configuration du déploiement MariaDB.



```
File Edit Selection View Go ... Kubernetes
EXPLORER
KUBERNETES
  emptyDir
  ! redis.yaml
  MariaDB
    ! mariadb-deploymen...
  Nginx
    ! webapp-ingress.yaml
    ! webapp.yaml
  PersistentVolume
    ! pv-nginx.yaml
    ! pvc-nginx.yaml
    ! pvc-pod.yaml
    ! ingress2.yaml
bapp.yaml ! webapp-ingress.yaml ! pv-nginx.yaml ! pvc-nginx.yaml
MariaDB > ! mariadb-deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    labels:
5      app: mariadb
6    name: mariadb
7  spec:
8    selector:
9      matchLabels:
10     app: mariadb
11  template:
12    metadata:
13     labels:
14     app: mariadb
15  spec:
16    containers:
17     - image: mariadb:latest
18     name: mariadb
19
```

Le fichier **pvc.yaml** permet de créer un PersistentVolumeClaim pour stocker les données de MariaDB.



```
File Edit Selection View Go ... Kubernetes
EXPLORER
KUBERNETES
  emptyDir
  ! redis.yaml
  MariaDB
    ! mariadb-deploymen...
    ! pvc.yaml
  Nginx
    ! webapp-ingress.yaml
    ! webapp.yaml
  PersistentVolume
    ! pv-nginx.yaml
    ! pvc-nginx.yaml
    ! pvc-pod.yaml
    ! ingress2.yaml
webapp-ingress.yaml ! pv-nginx.yaml ! pvc-nginx.yaml ! pvc-pod.yaml
MariaDB > ! pvc.yaml
1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: mariadb-data
5  spec:
6    accessModes:
7     - ReadWriteOnce
8    resources:
9     requests:
10     storage: 100Mi
```

Le PersistentVolumeClaim mariadb-data est associé à un volume persistant avec le statut Bound.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pvc mariadb-data
NAME          STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
mariadb-data  Bound  pvc-8dbc6c9b-3f4f-43e2-8ea9-cb9854be3735  100Mi     RWO           standard      <unset>                13m
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>
```

La commande **kubectl apply -f pvc.yaml** permet de créer le PersistentVolumeClaim mariadb-data.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl apply -f pvc.yaml
persistentvolumeclaim/mariadb-data created
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>
```

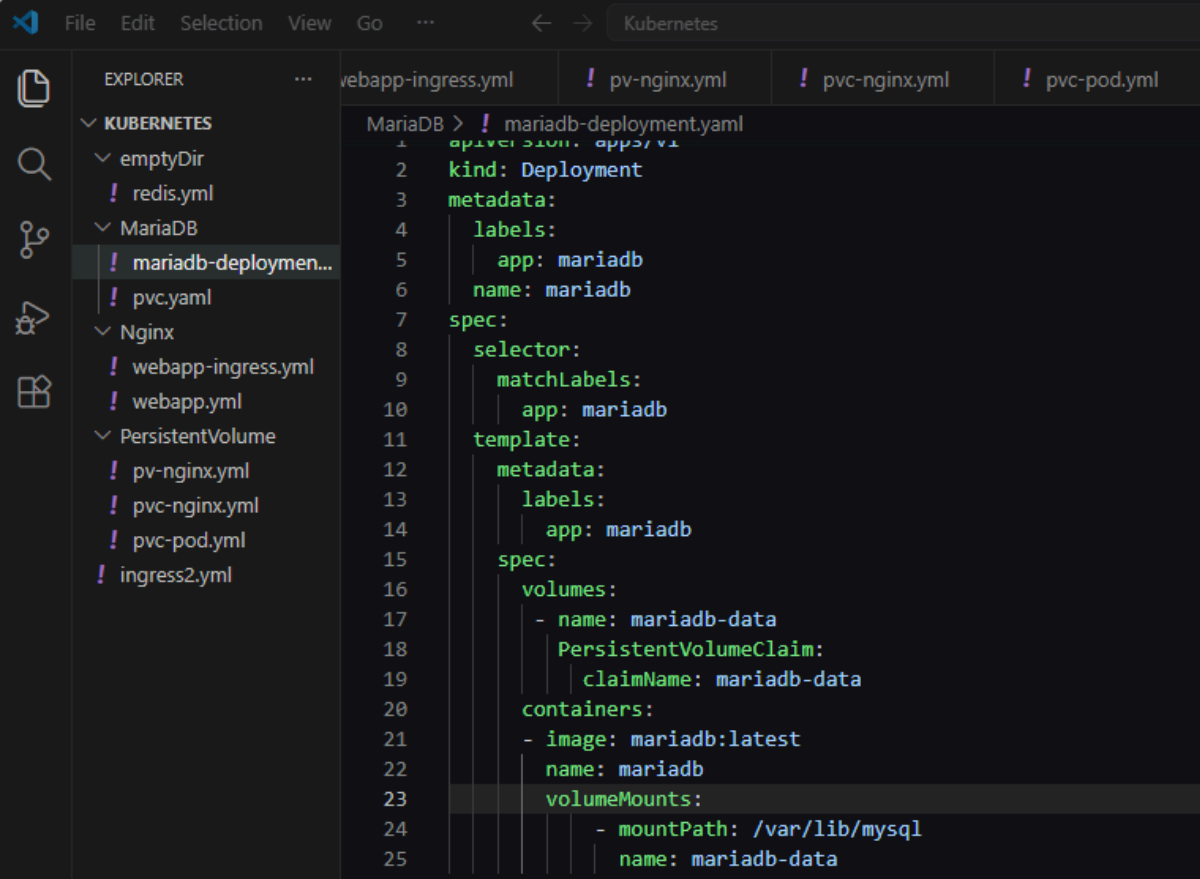
Le pod MariaDB est en état CrashLoopBackOff, indiquant un échec du démarrage du conteneur suivi de redémarrages automatiques.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pod -l app=mariadb
NAME          READY  STATUS             RESTARTS  AGE
mariadb-5c476b5c79-256d2  0/1    CrashLoopBackOff   1 (8s ago)  11s
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>
```

Le pod MariaDB est en état Error, indiquant un problème empêchant le démarrage correct du conteneur.

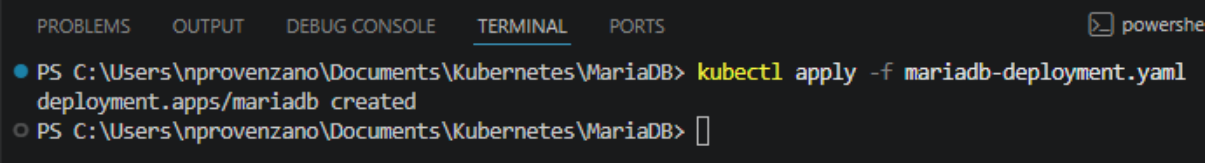
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pod -l app=mariadb
NAME          READY  STATUS   RESTARTS  AGE
mariadb-5c476b5c79-256d2  0/1    Error    3 (38s ago)  55s
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>
```

La modification du fichier **mariadb-deployment.yaml** permet d'ajouter un volume persistant au conteneur MariaDB.



```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   labels:
5     app: mariadb
6   name: mariadb
7 spec:
8   selector:
9     matchLabels:
10      app: mariadb
11   template:
12     metadata:
13       labels:
14         app: mariadb
15     spec:
16       volumes:
17         - name: mariadb-data
18           PersistentVolumeClaim:
19             claimName: mariadb-data
20       containers:
21         - image: mariadb:latest
22           name: mariadb
23           volumeMounts:
24             - mountPath: /var/lib/mysql
25               name: mariadb-data
```

La commande **kubectl apply -f mariadb-deployment.yaml** permet de créer le déploiement MariaDB avec le volume persistant.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell
PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl apply -f mariadb-deployment.yaml
deployment.apps/mariadb created
PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> 
```



La commande **kubectl apply -f mariadb-deployment.yaml** permet de mettre à jour le déploiement MariaDB.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl apply -f mariadb-deployment.yaml
deployment.apps/mariadb configured
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> █
```

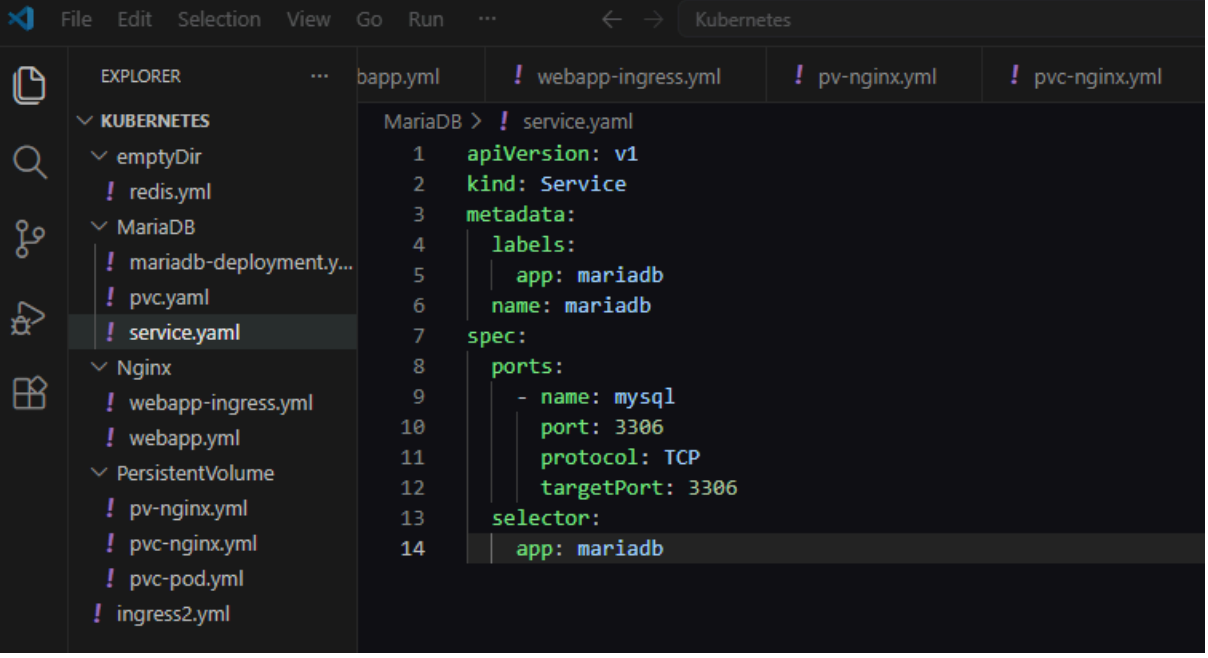
Le pod MariaDB est maintenant en état Running, indiquant que le conteneur fonctionne correctement.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pod -l app=mariadb
NAME                                READY  STATUS   RESTARTS  AGE
mariadb-5bbb9c9b6d-1lktt            1/1    Running  0          28s
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> █
```

La commande **mariadb-admin status** permet de vérifier que le serveur MariaDB fonctionne correctement dans le conteneur.

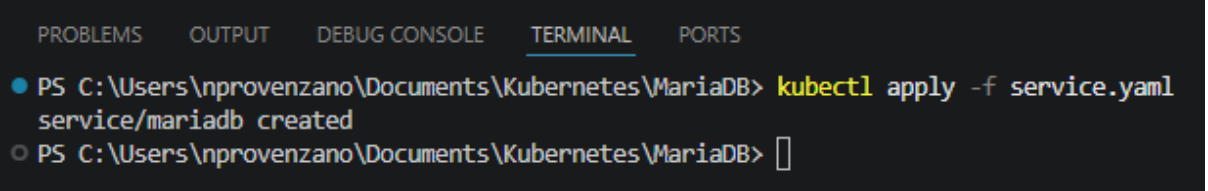
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl exec -it deployment/mariadb -- bash
root@mariadb-5bbb9c9b6d-1lktt:/# mariadb-admin status -p$MARIADB_ROOT_PASSWORD
Uptime: 127 Threads: 1 Questions: 1 Slow queries: 0 Opens: 17 Open tables: 10 Queries per second avg: 0.007
root@mariadb-5bbb9c9b6d-1lktt:/# exit
exit
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> █
```

Le fichier **service.yaml** permet de créer un service Kubernetes pour exposer MariaDB sur le port 3306.



```
File Edit Selection View Go Run ... Kubernetes
EXPLORER
KUBERNETES
  emptyDir
  ! redis.yaml
  MariaDB
    ! mariadb-deployment.y...
    ! pvc.yaml
    ! service.yaml
  Nginx
    ! webapp-ingress.yaml
    ! webapp.yaml
  PersistentVolume
    ! pv-nginx.yaml
    ! pvc-nginx.yaml
    ! pvc-pod.yaml
    ! ingress2.yaml
MariaDB > ! service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    labels:
5      app: mariadb
6  name: mariadb
7  spec:
8    ports:
9      - name: mysql
10        port: 3306
11          protocol: TCP
12            targetPort: 3306
13  selector:
14    app: mariadb
```

La commande **kubectl apply -f service.yaml** permet de créer le service MariaDB.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\provenzano\Documents\Kubernetes\MariaDB> kubectl apply -f service.yaml
service/mariadb created
○ PS C:\Users\provenzano\Documents\Kubernetes\MariaDB> 
```

La configuration des probes permet à Kubernetes de surveiller l'état et le bon fonctionnement du conteneur MariaDB.

```

7 spec:
11   template:
12     metadata:
13     spec:
14       volumes:
15         - name: mariadb-data
16           persistentVolumeClaim:
17             claimName: mariadb-data
18       containers:
19         - image: mariadb:latest
20           name: mariadb
21           volumeMounts:
22             - mountPath: /var/lib/mysql
23               name: mariadb-data
24           env:
25             - name: MARIADB_ROOT_PASSWORD
26               value: mot-de-passe-root
27           startupProbe: &probe
28           exec:
29             commande:
30               - "sh"
31               - "-c"
32               - "mariadb-admin status -p$MARIADB_ROOT_PASSWORD"
33           livenessProbe: *probe
34           readinessProbe: *probe
35
36
37

```

La commande `kubectl apply -f mariadb-deployment.yaml` permet d'appliquer les modifications du déploiement MariaDB.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl apply -f mariadb-deployment.yaml
deployment.apps/mariadb configured
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

Les pods MariaDB sont en état Running, indiquant que les probes fonctionnent correctement.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pod -l app=mariadb
NAME                                READY STATUS RESTARTS AGE
mariadb-5bbb9c9b6d-1lkt             1/1   Running 0      11m
mariadb-6574478f7f-b886w           0/1   Running 3 (9s ago) 104s
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

La commande `kubectl describe pods mariadb-6574478f7f-b080w` permet d'afficher les détails du pod ainsi que les informations des probes Kubernetes.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pod -l app=mariadb
NAME                READY  STATUS   RESTARTS   AGE
mariadb-5bbb9c9b6d-1lkt  1/1    Running  0           12m
mariadb-6574478f7f-b886w  0/1    Running  4 (10s ago)  2m17s
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl describe pods mariadb-6574478f7f-b886w
Name:                mariadb-6574478f7f-b886w
Namespace:           default
Priority:             0
Service Account:     default
Node:                minikube/172.28.25.20
Start Time:          Thu, 07 May 2026 12:34:19 +0200
Labels:              app=mariadb
                    pod-template-hash=6574478f7f
Annotations:         <none>
Status:              Running
IP:                 10.244.0.73
IPs:
  IP:               10.244.0.73
Controlled By:      ReplicaSet/mariadb-6574478f7f
Containers:
  mariadb:
    Container ID:   docker://b01ed11f2b0b3731f5f6255031a25a8303a263b7cb3f1caa046c6f241cc1bd66
    Image:          mariadb:latest
    Image ID:       docker-pullable://mariadb@sha256:e0236fc6386e7eacd9359e59d0a078bd7aa0d18280d36d13061121bedeae903
    Port:          <none>
    Host Port:     <none>
    State:         Waiting
      Reason:      CrashLoopBackOff
    Last State:    Terminated
      Reason:      Error
      Exit Code:   1
      Started:    Thu, 07 May 2026 12:36:27 +0200
      Finished:   Thu, 07 May 2026 12:36:58 +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-passe-root
    Mounts:
      /var/lib/mysql from mariadb-data (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-9xcrp (ro)

```

Les événements du pod indiquent que la startupProbe a échoué car le serveur MariaDB n'était pas encore prêt au démarrage.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl describe pods mariadb-6574478f7f-b886w
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-passe-root
Mounts:
  /var/lib/mysql from mariadb-data (rw)
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-9xcrp (ro)
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-9xcrp:
    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   3m55s  default-scheduler  Successfully assigned default/mariadb-6574478f7f-b886w to minikube
  Normal  Pulled     3m4s   kubelet        Successfully pulled image "mariadb:latest" in 1.023s (1.023s including waiting). Image size: 335568219 bytes.
  Normal  Pulled     2m32s  kubelet        Successfully pulled image "mariadb:latest" in 878ms (878ms including waiting). Image size: 335568219 bytes.
  Normal  Pulled     2m      kubelet        Successfully pulled image "mariadb:latest" in 914ms (914ms including waiting). Image size: 335568219 bytes.
  Normal  Pulled     89s    kubelet        Successfully pulled image "mariadb:latest" in 823ms (823ms including waiting). Image size: 335568219 bytes.
  Normal  Pulling    58s (x5 over 3m5s)  kubelet        Pulling image "mariadb:latest"
  Normal  Created   57s (x5 over 3m3s)  kubelet        Container created
  Normal  Started   57s (x5 over 3m3s)  kubelet        Container started
  Normal  Pulled    57s          kubelet        Successfully pulled image "mariadb:latest" in 886ms (886ms including waiting). Image size: 335568219 bytes.
  Warning Unhealthy 35s (x15 over 2m55s) 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 mariadb is running and that the socket: '/run/mysqld/mysqld.sock' exists!
  Normal Killing 35s (x5 over 2m35s)  kubelet        Container mariadb failed startup probe, will be restarted
  Warning BackOff 25s (x3 over 26s)   kubelet        Back-off restarting failed container mariadb in pod mariadb-6574478f7f-b886w_default(2aac4fdd-edf8-4d5e-ad98-3429141b40f7)
PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

Le nouveau pod MariaDB fonctionne correctement tandis que l'ancien pod reste en état CrashLoopBackOff.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pod -l app=mariadb
NAME                READY  STATUS             RESTARTS  AGE
mariadb-5bbb9c9b6d-1lkt  1/1    Running            0          14m
mariadb-6574478f7f-b886w  0/1    CrashLoopBackOff   6 (15s ago)  4m56s
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

Les logs indiquent un conflit d'accès aux fichiers de la base de données, empêchant le démarrage correct du second pod MariaDB.

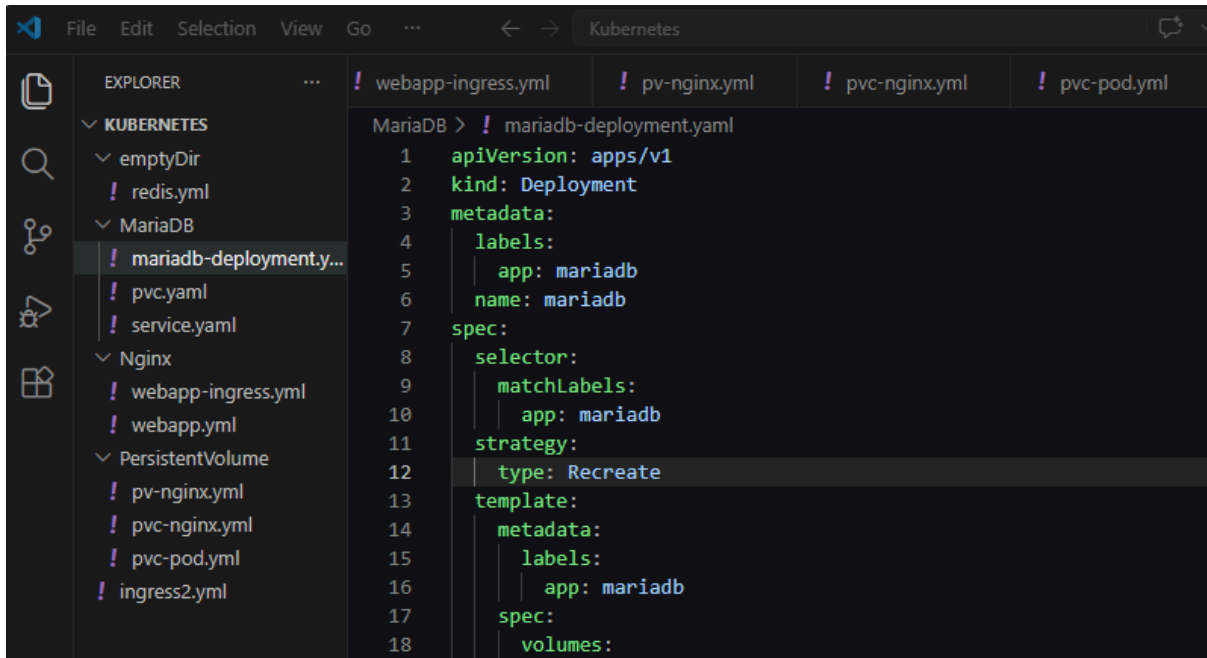
```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl logs mariadb-6574478f7f-b886w
2026-05-07 10:38:29+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2+maria-ubu2404 started.
2026-05-07 10:38:29+00:00 [Warn] [Entrypoint]: /sys/fs/cgroup///memory.pressure not writable, functionality unavailable to MariaDB
2026-05-07 10:38:29+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2026-05-07 10:38:29+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2+maria-ubu2404 started.
2026-05-07 10:38:30+00:00 [Note] [Entrypoint]: MariaDB upgrade not required
2026-05-07 10:38:30 0 [Note] Starting MariaDB 12.2.2-MariaDB-ubu2404 source revision d26a6f44c1f2119377e79a9540886c6d8c01472f server_uid ldktFOeG0gKgWuSt9MPgsZ26d3Q= as process 1
2026-05-07 10:38:30 0 [ERROR] mariadbd: Can't lock aria control file '/var/lib/mysql/aria_log_control' for exclusive use, error: 11. Will retry for 30 seconds
2026-05-07 10:39:00 0 [ERROR] mariadbd: Got error 'Could not get an exclusive lock; file is probably in use by another process' when trying to use aria control file '/var/lib/mysql/aria_log_control'
2026-05-07 10:39:00 0 [ERROR] Plugin 'Aria' registration as a STORAGE ENGINE failed.
2026-05-07 10:39:00 0 [Note] InnoDB: Compressed tables use zlib 1.3
2026-05-07 10:39:00 0 [Note] InnoDB: Number of transaction pools: 1
2026-05-07 10:39:00 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
2026-05-07 10:39:00 0 [Note] mariadbd: O_TMPFILE is not supported on /tmp (disabling future attempts)
2026-05-07 10:39:00 0 [Note] InnoDB: Using io_uring
2026-05-07 10:39:00 0 [Note] InnoDB: innodb_buffer_pool_size_max=128m, innodb_buffer_pool_size=128m
2026-05-07 10:39:00 0 [Note] InnoDB: Completed initialization of buffer pool
2026-05-07 10:39:00 0 [Note] InnoDB: File system buffers for log disabled (block size=512 bytes)
2026-05-07 10:39:00 0 [ERROR] InnoDB: Unable to lock ./ibdata1 error: 11
2026-05-07 10:39:00 0 [Note] InnoDB: Check that you do not already have another mariadbd process using the same InnoDB data or log files.
2026-05-07 10:39:00 0 [ERROR] InnoDB: Plugin initialization aborted with error Generic error
2026-05-07 10:39:00 0 [Note] InnoDB: Starting shutdown...
2026-05-07 10:39:00 0 [ERROR] Plugin 'InnoDB' registration as a STORAGE ENGINE failed.
2026-05-07 10:39:00 0 [Note] Plugin 'FEEDBACK' is disabled.
2026-05-07 10:39:00 0 [Note] Plugin 'wsrep-provider' is disabled.
2026-05-07 10:39:00 0 [ERROR] Could not open mysql.plugin table: "Unknown storage engine 'Aria'". Some plugins may be not loaded
2026-05-07 10:39:00 0 [ERROR] Failed to initialize plugins.
2026-05-07 10:39:00 0 [ERROR] Aborting
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

La stratégie Recreate permet d'éviter le démarrage simultané de plusieurs pods MariaDB utilisant le même volume persistant.

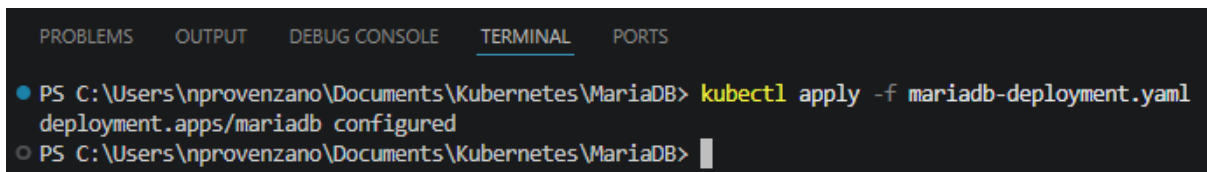


```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    labels:
5      app: mariadb
6    name: mariadb
7  spec:
8    selector:
9      matchLabels:
10       app: mariadb
11    strategy:
12     type: Recreate
13  template:
14    metadata:
15     labels:
16       app: mariadb
17    spec:
18     volumes:

```

La commande `kubectl apply -f mariadb-deployment.yaml` permet d'appliquer la stratégie Recreate au déploiement MariaDB.

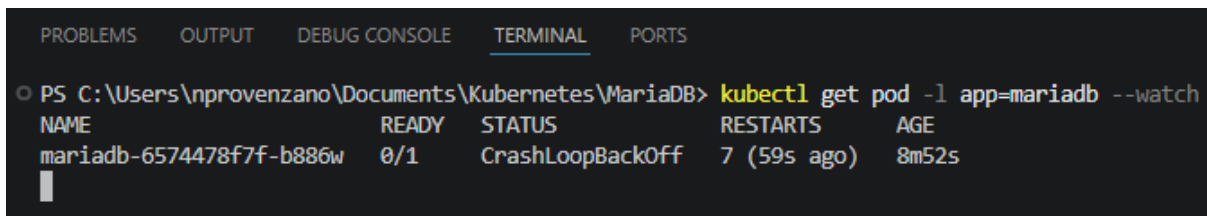


```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl apply -f mariadb-deployment.yaml
deployment.apps/mariadb configured
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

Le pod MariaDB redémarre automatiquement après un échec détecté par Kubernetes.



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pod -l app=mariadb --watch
NAME                READY  STATUS             RESTARTS  AGE
mariadb-6574478f7f-b886w  0/1    CrashLoopBackOff   7 (59s ago)  8m52s

```

Le pod MariaDB est toujours en état CrashLoopBackOff malgré l'application de la stratégie Recreate.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pod -l app=mariadb
NAME                                READY  STATUS              RESTARTS  AGE
mariadb-6574478f7f-b886w            0/1    CrashLoopBackOff    7 (2m36s ago)  10m
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

La commande `kubectl delete pod -l app=mariadb` permet de supprimer les pods MariaDB du déploiement.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl delete pod -l app=mariadb
pod "mariadb-6574478f7f-b886w" deleted from default namespace
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

Le nouveau pod MariaDB est maintenant en état Running après la recréation du déploiement.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pod -l app=mariadb
NAME                                READY  STATUS   RESTARTS  AGE
mariadb-6574478f7f-65tkj            1/1    Running  0          22s
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

La commande `kubectl scale deployment mariadb --replicas=2` permet d'augmenter le nombre de pods du déploiement MariaDB.

Le second pod MariaDB démarre mais rencontre des problèmes liés au partage du même volume persistant.

```

Administrateur : Windows Po1  x  +  v
PS C:\Users\nprovenzano> kubectl scale deployment mariadb --replicas=2
deployment.apps/mariadb scaled
PS C:\Users\nprovenzano> kubectl get pods -l app=mariadb --watch
NAME                                READY  STATUS   RESTARTS  AGE
mariadb-6574478f7f-65tkj            1/1    Running  0          2m33s
mariadb-6574478f7f-qqdvt            0/1    Running  0          20s
mariadb-6574478f7f-qqdvt            0/1    Running  1 (2s ago)  34s

```

La commande `kubectl describe pods mariadb-6574478f7f-qgdvt` permet d'afficher les informations détaillées et l'état des probes du pod MariaDB.

```

Administrateur : Windows Poi x + v
PS C:\Users\nprovenzano> kubectl describe pods mariadb-6574478f7f-qgdvt
Name:          mariadb-6574478f7f-qgdvt
Namespace:    default
Priority:      0
Service Account: default
Node:         minikube/172.28.25.20
Start Time:   Thu, 07 May 2026 12:47:32 +0200
Labels:       app=mariadb
              pod-template-hash=6574478f7f
Annotations:  <none>
Status:       Running
IP:           10.244.0.75
IPs:
IP:           10.244.0.75
Controlled By: ReplicaSet/mariadb-6574478f7f
Containers:
  mariadb:
    Container ID:  docker://3047193eda541c added9dc966859a9db028cbffa279831fffd13cad2c66b1cac3e9
    Image:         mariadb:latest
    Image ID:     docker-pullable://mariadb@sha256:e0236fc6386e7eacd9359e59d0a078bd7aa0d18280d36d13061121bedeae903
    Port:         <none>
    Host Port:    <none>
    State:        Running
      Started:    Thu, 07 May 2026 12:48:37 +0200
    Last State:   Terminated
      Reason:     Error
      Exit Code:  1
      Started:    Thu, 07 May 2026 12:48:05 +0200
      Finished:   Thu, 07 May 2026 12:48:36 +0200
    Ready:        False
    Restart Count: 2
    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-passe-root
    Mounts:
      /var/lib/mysql from mariadb-data (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-czxc (ro)
Conditions:
  Type                               Status
  PodReadyToStartContainers          True
  Initialized                         True
  Ready                               False
  ContainersReady                    False
  PodScheduled                        True
Volumes:
  mariadb-data:

```

Les événements du pod indiquent un échec de la startupProbe empêchant le démarrage correct du second pod MariaDB.

```

Mounts:
  /var/lib/mysql from mariadb-data (rw)
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-czxcp (ro)
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-czxcp:
    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   90s   default-scheduler  Successfully assigned default/mariadb-6574478f7f-qqdvt to mi
nikube
  Normal  Pulled     88s   kubelet       Successfully pulled image "mariadb:latest" in 886ms (886ms i
ncluding waiting). Image size: 335568219 bytes.
  Normal  Pulled     57s   kubelet       Successfully pulled image "mariadb:latest" in 893ms (893ms i
ncluding waiting). Image size: 335568219 bytes.
  Normal  Killing    29s (x2 over 59s)  kubelet       Container mariadb failed startup probe, will be restarted
  Normal  Pulling    26s (x3 over 89s)  kubelet       Pulling image "mariadb:latest"
  Normal  Created    25s (x3 over 88s)  kubelet       Container created
  Normal  Started    25s (x3 over 88s)  kubelet       Container started
  Normal  Pulled     25s   kubelet       Successfully pulled image "mariadb:latest" in 850ms (850ms i
ncluding waiting). Image size: 335568219 bytes.
  Warning Unhealthy  9s (x8 over 79s)  kubelet       Startup probe failed: mariadb-admin: connect to server at 'l
ocalhost' 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!
PS C:\Users\nprovenzano>

```

Les logs du second pod MariaDB indiquent un conflit d'accès au volume persistant déjà utilisé par un autre pod.

```

Administrateur : Windows Po x + v
PS C:\Users\nprovenzano> kubectl get pods -l app=mariadb
NAME                READY   STATUS    RESTARTS   AGE
mariadb-6574478f7f-65tkj   1/1     Running   0           6m39s
mariadb-6574478f7f-qqdvt   0/1     Running   6 (18s ago) 4m26s
PS C:\Users\nprovenzano> kubectl logs mariadb-6574478f7f-qqdvt
2026-05-07 10:51:41+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2+maria-ubu2404 started.
2026-05-07 10:51:41+00:00 [Warn] [Entrypoint]: /sys/fs/cgroup///memory.pressure not writable, functionality unavailable to MariaDB
2026-05-07 10:51:41+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2026-05-07 10:51:41+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2+maria-ubu2404 started.
2026-05-07 10:51:42+00:00 [Note] [Entrypoint]: MariaDB upgrade not required
2026-05-07 10:51:42 0 [Note] Starting MariaDB 12.2.2-MariaDB-ubu2404 source revision d26a6f44c1f2119377e79a9540886c6d8c01472f server_uid vDdrKvKreeTlMvKkXRttIO8vAUU= as process 1
2026-05-07 10:51:42 0 [ERROR] mariadbd: Can't lock aria control file '/var/lib/mysql/aria_log_control' for exclusive use, error: 11. Will retry for 30 seconds
2026-05-07 10:52:12 0 [ERROR] mariadbd: Got error 'Could not get an exclusive lock; file is probably in use by another process' when trying to use aria control file '/var/lib/mysql/aria_log_control'
2026-05-07 10:52:12 0 [ERROR] Plugin 'Aria' registration as a STORAGE ENGINE failed.
2026-05-07 10:52:12 0 [Note] InnoDB: Compressed tables use zlib 1.3
2026-05-07 10:52:12 0 [Note] InnoDB: Number of transaction pools: 1
2026-05-07 10:52:12 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
2026-05-07 10:52:12 0 [Note] mariadbd: O_TMPFILE is not supported on /tmp (disabling future attempts)
2026-05-07 10:52:12 0 [Note] InnoDB: Using io_uring
2026-05-07 10:52:12 0 [Note] InnoDB: innodb_buffer_pool_size_max=128m, innodb_buffer_pool_size=128m
2026-05-07 10:52:12 0 [Note] InnoDB: Completed initialization of buffer pool
2026-05-07 10:52:12 0 [Note] InnoDB: File system buffers for log disabled (block size=512 bytes)
2026-05-07 10:52:12 0 [ERROR] InnoDB: Unable to lock ./ibdata1 error: 11
2026-05-07 10:52:12 0 [Note] InnoDB: Check that you do not already have another mariadbd process using the same InnoDB data or log files.
2026-05-07 10:52:12 0 [ERROR] InnoDB: Plugin initialization aborted with error Generic error
2026-05-07 10:52:12 0 [Note] InnoDB: Starting shutdown...
2026-05-07 10:52:12 0 [ERROR] Plugin 'InnoDB' registration as a STORAGE ENGINE failed.
2026-05-07 10:52:12 0 [Note] Plugin 'FEEDBACK' is disabled.
2026-05-07 10:52:12 0 [Note] Plugin 'wsrep-provider' is disabled.
2026-05-07 10:52:12 0 [ERROR] Could not open mysql.plugin table: "Unknown storage engine 'Aria'". Some plugins may be not loaded
2026-05-07 10:52:12 0 [ERROR] Failed to initialize plugins.
2026-05-07 10:52:12 0 [ERROR] Aborting
PS C:\Users\nprovenzano>

```

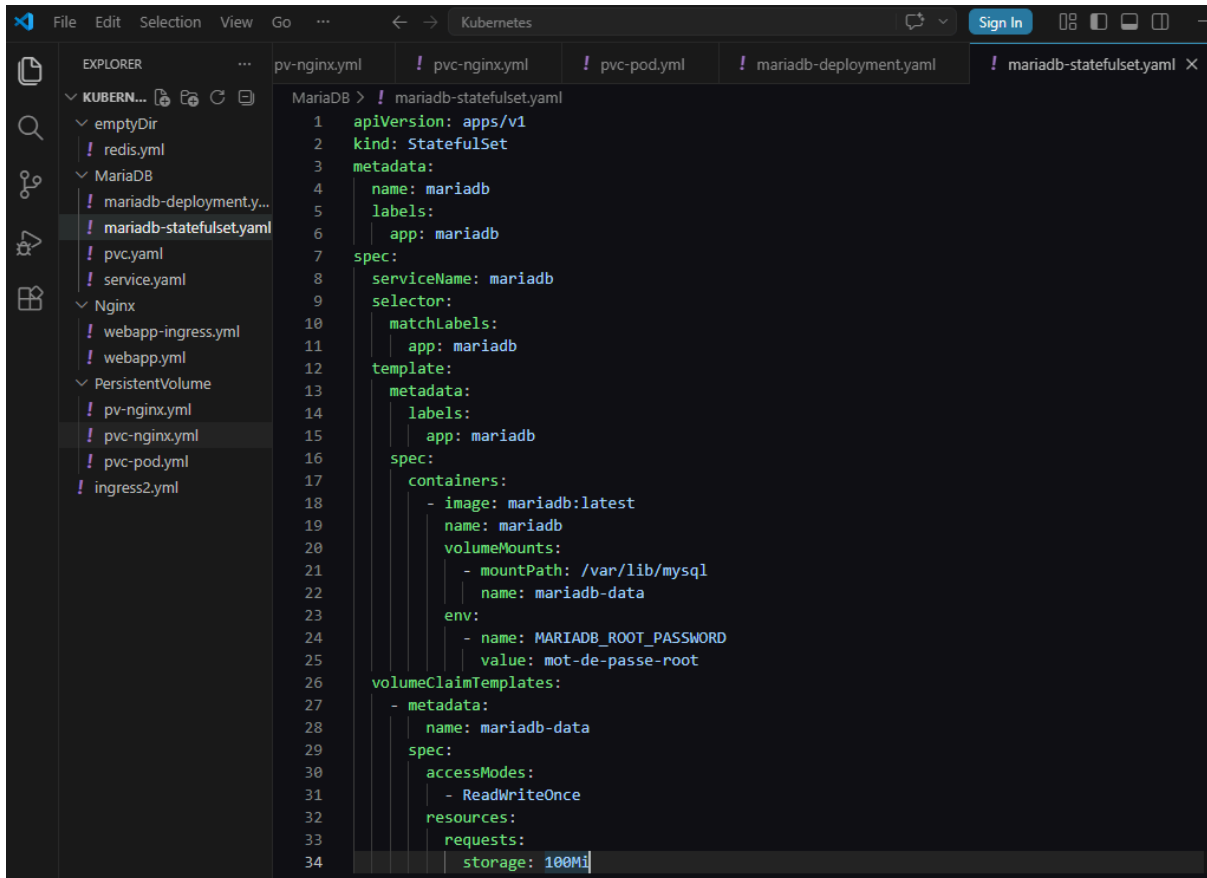
La suppression du déploiement MariaDB entraîne également la suppression des pods associés.

```

Administrateur : Windows Po x + v
PS C:\Users\nprovenzano> kubectl delete deployment mariadb
deployment.apps "mariadb" deleted from default namespace
PS C:\Users\nprovenzano> kubectl get pods -l app=mariadb
No resources found in default namespace.
PS C:\Users\nprovenzano>

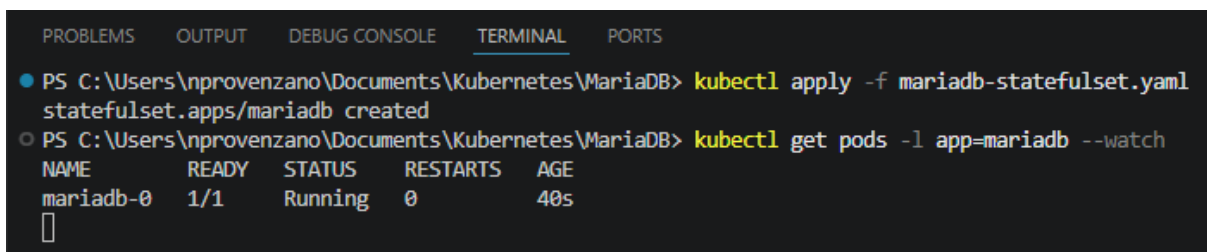
```

Le fichier **mariadb-statefulset.yaml** permet de créer un StatefulSet pour gérer les pods MariaDB avec un stockage persistant dédié.



```
1 apiVersion: apps/v1
2 kind: StatefulSet
3 metadata:
4   name: mariadb
5   labels:
6     app: mariadb
7 spec:
8   serviceName: mariadb
9   selector:
10    matchLabels:
11      app: mariadb
12   template:
13     metadata:
14       labels:
15         app: mariadb
16     spec:
17       containers:
18         - image: mariadb:latest
19           name: mariadb
20           volumeMounts:
21             - mountPath: /var/lib/mysql
22               name: mariadb-data
23           env:
24             - name: MARIADB_ROOT_PASSWORD
25               value: mot-de-passe-root
26   volumeClaimTemplates:
27     - metadata:
28       name: mariadb-data
29       spec:
30         accessModes:
31           - ReadWriteOnce
32         resources:
33           requests:
34             storage: 100Mi
```

La commande **kubectl apply -f mariadb-statefulset.yaml** permet de créer le StatefulSet MariaDB. Le pod mariadb-0 démarre correctement avec le StatefulSet.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl apply -f mariadb-statefulset.yaml
statefulset.apps/mariadb created
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pods -l app=mariadb --watch
NAME        READY  STATUS   RESTARTS  AGE
mariadb-0  1/1    Running  0          40s
□
```

La commande **kubectl get pvc** permet d'afficher les PersistentVolumeClaims créés automatiquement par le StatefulSet MariaDB.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pvc
NAME                STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
mariadb-data        Bound  pvc-8dbc6c9b-3f4f-43e2-8ea9-cb9854be3735  100Mi     RWO           standard      <unset>                 67m
mariadb-data-mariadb-0  Bound  pvc-8291442b-b727-48b3-8e02-2447a14730ef  100Mi     RWO           standard      <unset>                 72s
mysql-pvc           Bound  pvc-5f49d2b5-9cad-49b2-bd60-3da2e09ac5c8  2Gi       RWO           standard      <unset>                 6d23h
task-pv-claim       Bound  pv-nginx                                    10Mi      RWO           manual        <unset>                 88m
wordpress-pvc       Bound  pvc-15b23d24-7525-4ae4-895a-b57dd439c6e9  2Gi       RWO           standard      <unset>                 6d23h
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

Le StatefulSet crée automatiquement un volume persistant dédié pour chaque pod MariaDB.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pv
NAME                CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM                                     STORAGECLASS  VOLUMEATTRIBUTESCLASS  REASON  AGE
mysql-pv           2Gi       RWO           Retain           Available
pv-nginx           10Mi      RWO           Retain           Bound   default/task-pv-claim                   manual        <unset>                 92m
pvc-15b23d24-7525-4ae4-895a-b57dd439c6e9  2Gi       RWO           Delete          Bound   default/wordpress-pvc                  standard     <unset>                 6d23h
pvc-5f49d2b5-9cad-49b2-bd60-3da2e09ac5c8  2Gi       RWO           Delete          Bound   default/mysql-pvc                      standard     <unset>                 6d23h
pvc-8291442b-b727-48b3-8e02-2447a14730ef  100Mi     RWO           Delete          Bound   default/mariadb-data-mariadb-0         standard     <unset>                 112s
pvc-8dbc6c9b-3f4f-43e2-8ea9-cb9854be3735  100Mi     RWO           Delete          Bound   default/mariadb-data                   standard     <unset>                 68m
wordpress-pv       2Gi       RWO           Retain           Available
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

La commande **kubectl delete pvc/mariadb-data** permet de supprimer l'ancien PersistentVolumeClaim du déploiement MariaDB.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl delete pvc/mariadb-data
persistentvolumeclaim "mariadb-data" deleted from default namespace
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

La commande **kubectl scale sts mariadb --replicas=2** permet d'augmenter le nombre de pods du StatefulSet MariaDB.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl scale sts mariadb --replicas=2
statefulset.apps/mariadb scaled
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

Les pods mariadb-0 et mariadb-1 sont en état Running avec leurs propres volumes persistants créés automatiquement par le StatefulSet.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pvc -l app=mariadb
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS VOLUMEATTRIBUTESCLASS AGE
mariadb-data-mariadb-0 Bound pvc-8291442b-b727-48b3-8e02-2447a14730ef 100Mi RWO standard <unset> 4m20s
mariadb-data-mariadb-1 Bound pvc-eb057c98-0c3f-4435-8ecc-5a0b4c14c22c 100Mi RWO standard <unset> 35s
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pods -l app=mariadb
NAME READY STATUS RESTARTS AGE
mariadb-0 1/1 Running 0 4m31s
mariadb-1 1/1 Running 0 46s
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl get pods,pvc -l app=mariadb
NAME READY STATUS RESTARTS AGE
pod/mariadb-0 1/1 Running 0 4m54s
pod/mariadb-1 1/1 Running 0 69s
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS VOLUMEATTRIBUTESCLASS AGE
persistentvolumeclaim/mariadb-data-mariadb-0 Bound pvc-8291442b-b727-48b3-8e02-2447a14730ef 100Mi RWO standard <unset> 4m54s
persistentvolumeclaim/mariadb-data-mariadb-1 Bound pvc-eb057c98-0c3f-4435-8ecc-5a0b4c14c22c 100Mi RWO standard <unset> 69s
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

Le fichier **configmap.yaml** permet de stocker les variables de configuration utilisées par MariaDB.

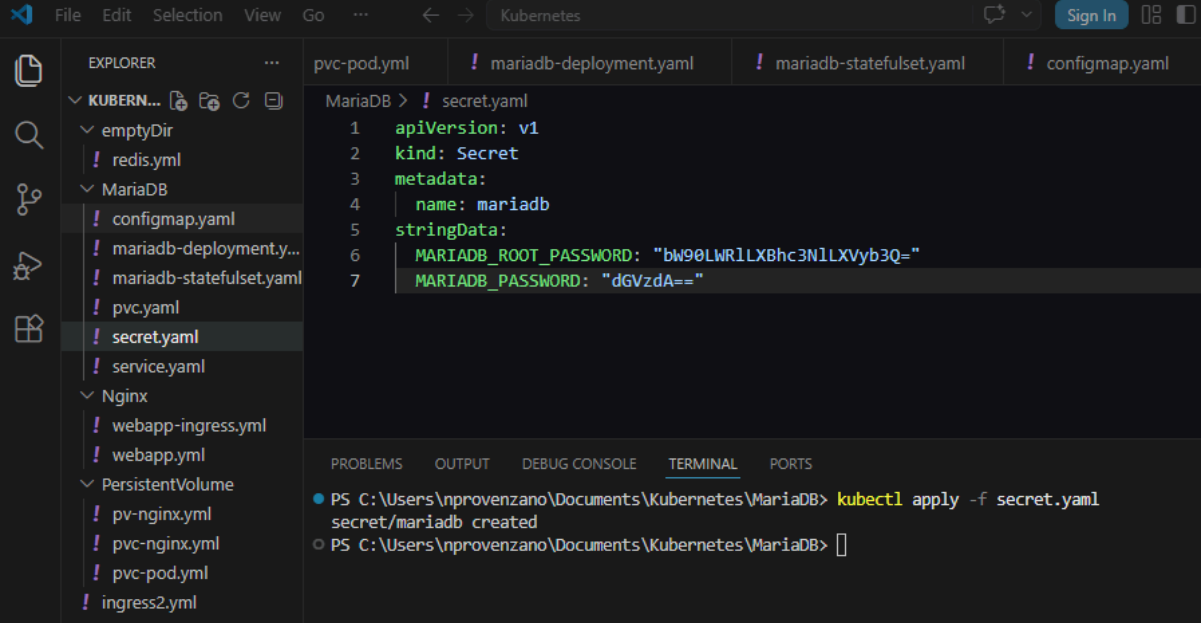
La commande **kubectl apply -f configmap.yaml** permet de créer le ConfigMap mariadb.

```

Kubernetes
File Edit Selection View Go ...
Sign In
EXPLORER
KUBERN...
  emptyDir
    ! redis.yml
  MariaDB
    ! configmap.yaml
    ! mariadb-deployment.y...
    ! mariadb-statefulset.yaml
    ! pvc.yaml
    ! secret.yaml
    ! service.yaml
  Nginx
    ! webapp-ingress.yaml
    ! webapp.yaml
  PersistentVolume
    ! pv-nginx.yaml
    ! pvc-nginx.yaml
    ! pvc-pod.yaml
    ! ingress2.yaml
pvc-pod.yaml mariadb-deployment.yaml mariadb-statefulset.yaml configmap.yaml
MariaDB > ! configmap.yaml
1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4   name: mariadb
5 data:
6   MARIADB_DATABASE: test
7   MARIADB_USER: test
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl apply -f configmap.yaml
configmap/mariadb created
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```

Le fichier **secret.yaml** permet de stocker les mots de passe MariaDB de manière sécurisée. La commande **kubectl apply -f secret.yaml** permet de créer le Secret mariadb.



The screenshot shows the Visual Studio Code interface with the Explorer view on the left and the Editor view on the right. The Explorer view shows a project structure with folders for 'KUBERN...', 'emptyDir', 'MariaDB', 'Nginx', and 'PersistentVolume'. The 'secret.yaml' file is selected in the Explorer view. The Editor view shows the content of 'secret.yaml' with the following YAML code:

```
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: mariadb
5 stringData:
6   MARIADB_ROOT_PASSWORD: "bW90LWR1LXBhc3N1LXVyY3Q="
7   MARIADB_PASSWORD: "dgVzdA=="
```

The Terminal view at the bottom shows the command `kubectl apply -f secret.yaml` being executed, resulting in the output `secret/mariadb created`.

La modification du fichier **mariadb-statefulset.yaml** permet d'utiliser les variables du ConfigMap et du Secret dans le conteneur MariaDB.

```

1  ---
2  apiVersion: apps/v1
3  kind: StatefulSet
4  metadata:
5    name: mariadb
6    labels:
7      app: mariadb
8  spec:
9    serviceName: mariadb
10   selector:
11     matchLabels:
12       app: mariadb
13   template:
14     metadata:
15       labels:
16         app: mariadb
17     spec:
18       containers:
19         - image: mariadb:latest
20           name: mariadb
21           volumeMounts:
22             - mountPath: /var/lib/mysql
23               name: mariadb-data
24           envFrom:
25             - configMapRef:
26                 name: mariadb
27             - secretRef:
28                 name: mariadb
29           startupProbe: &probe
30           exec:
31             command:
32               - "sh"
33               - "-c"
34               - "mariadb-admin status -p$MARIADB_ROOT_PASSWORD"
35           livenessProbe: *probe
36           readinessProbe: *probe
37   volumeClaimTemplates:
38     - metadata:
39         name: mariadb-data
40     spec:
41       accessModes:
42         - "ReadWriteOnce"
43       resources:
44         requests:
45           storage: 100Mi

```

La commande **kubectl apply -f mariadb-statefulset2.yaml** permet de mettre à jour le StatefulSet MariaDB avec le ConfigMap et le Secret.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB> kubectl apply -f mariadb-statefulset2.yaml
statefulset.apps/mariadb configured
○ PS C:\Users\nprovenzano\Documents\Kubernetes\MariaDB>

```