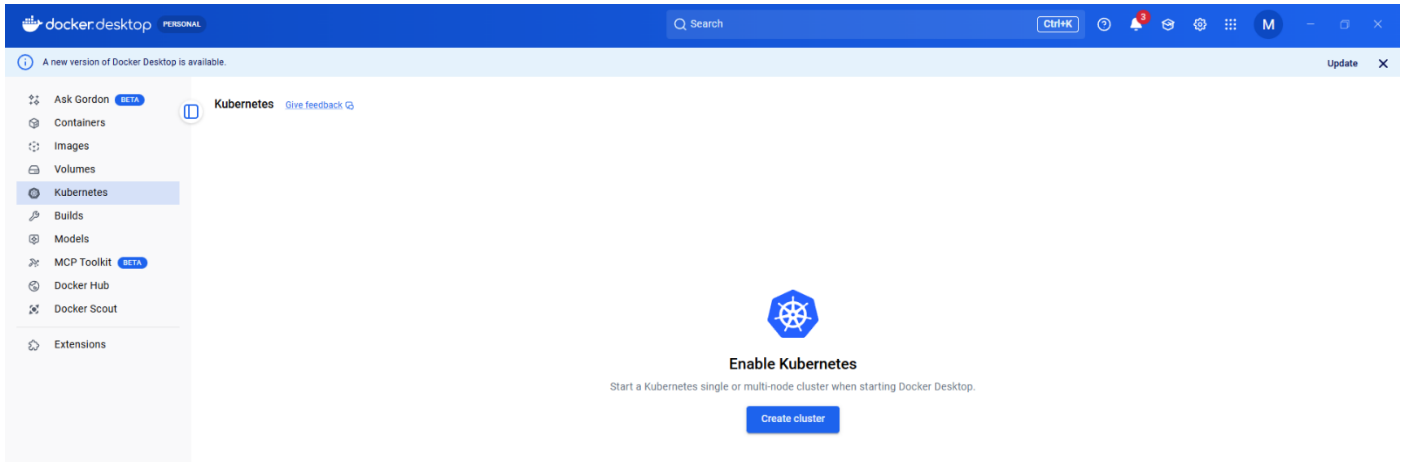


Sommaire

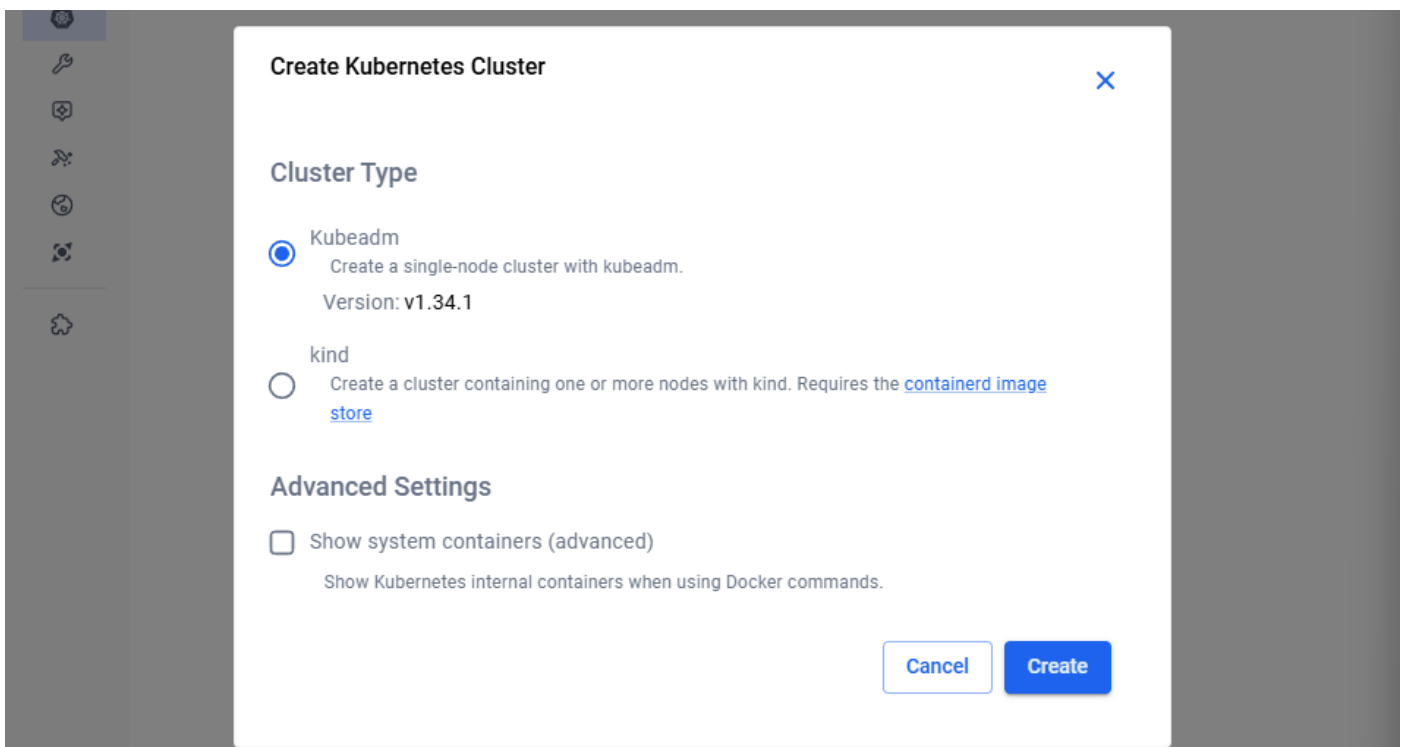
1. Kubernetes avec Docker Desktop.....	2
2. Installation de Minikube.	6
3. Le tableau de bord de Kubernetes.	14
3. Créer un déploiement à l'aide de l'outil kubectl.....	21
4. Automatisation de déploiement par fichier YAML.	29
5. Application.....	39
6. Suite Ingress (Mailpit) : hôtes virtuels et nom de domaine nip.io	44
7. Hôtes virtuels et nom de domaine nip.io : autre exemple	46
8. Cycle de vie d'un conteneur dans Kubernetes	46
9. Persistance des données	47
10. Hébergement d'un groupe de pods fonctionnant en cluster	55

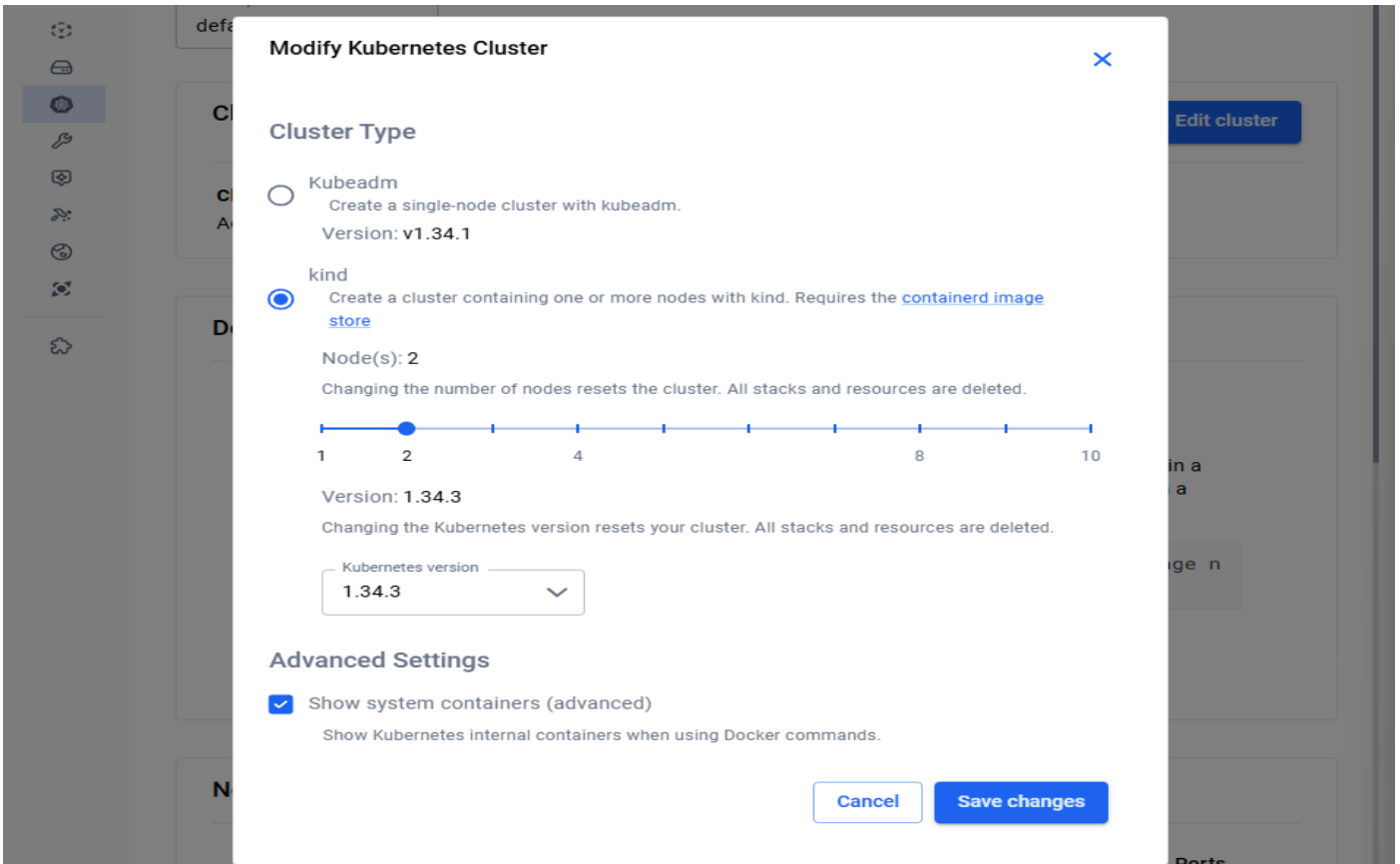
1. Kubernetes avec Docker Desktop.

- Je me rends dans l'onglet Kubernetes, puis je clique sur Create cluster :

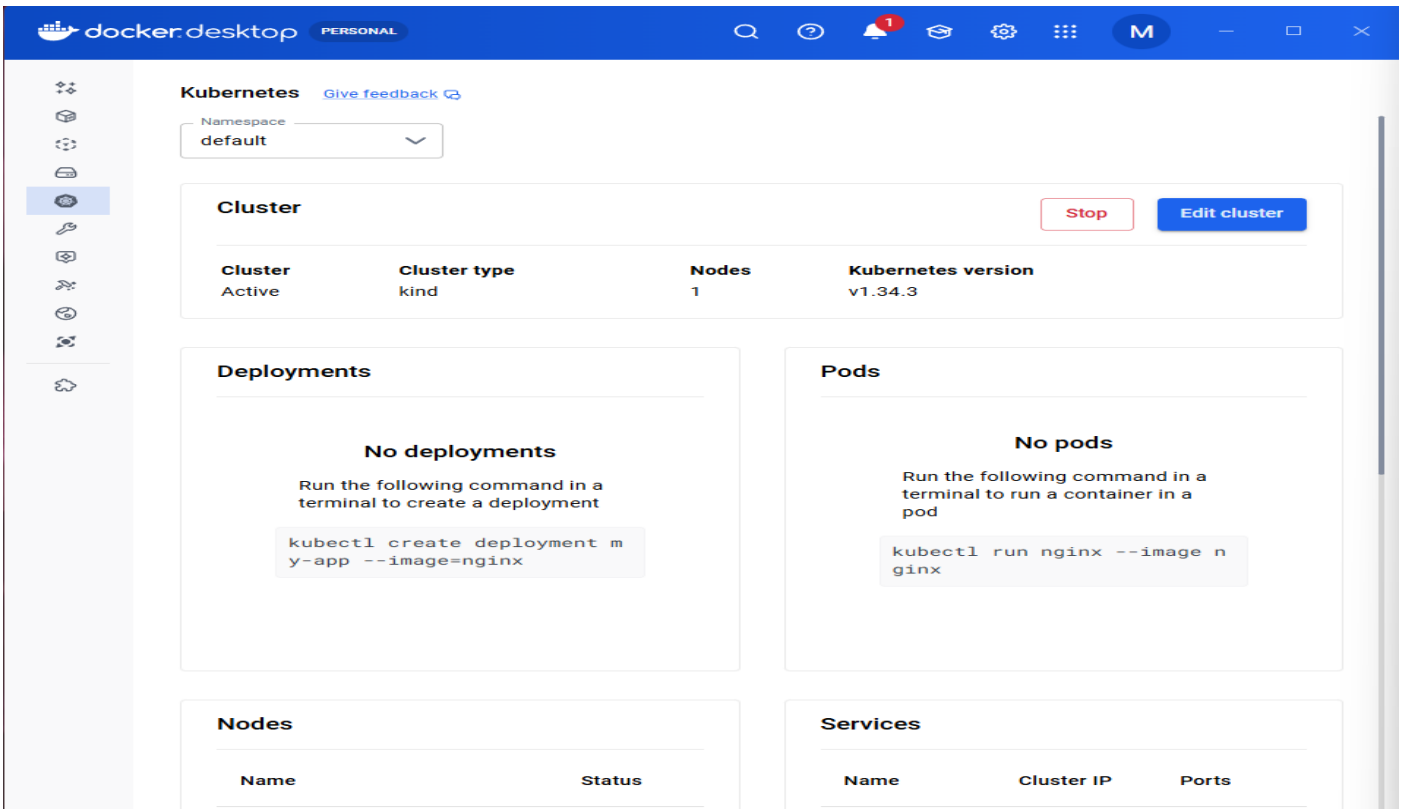


- L'option de base est celle de Kubeadm je sélectionne kind, je choisis 2 Nodes et je coche la case Show system containers (advanced) :





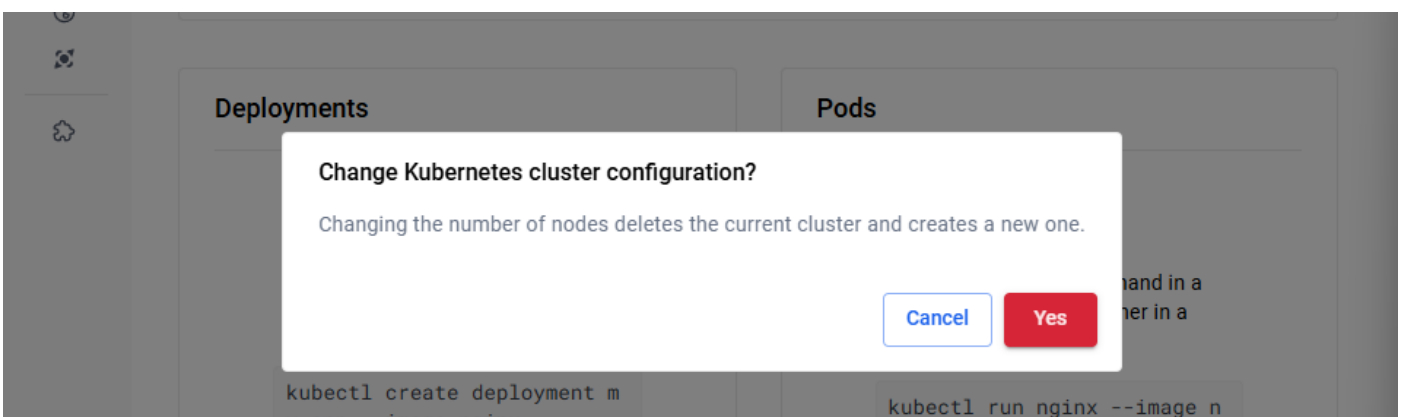
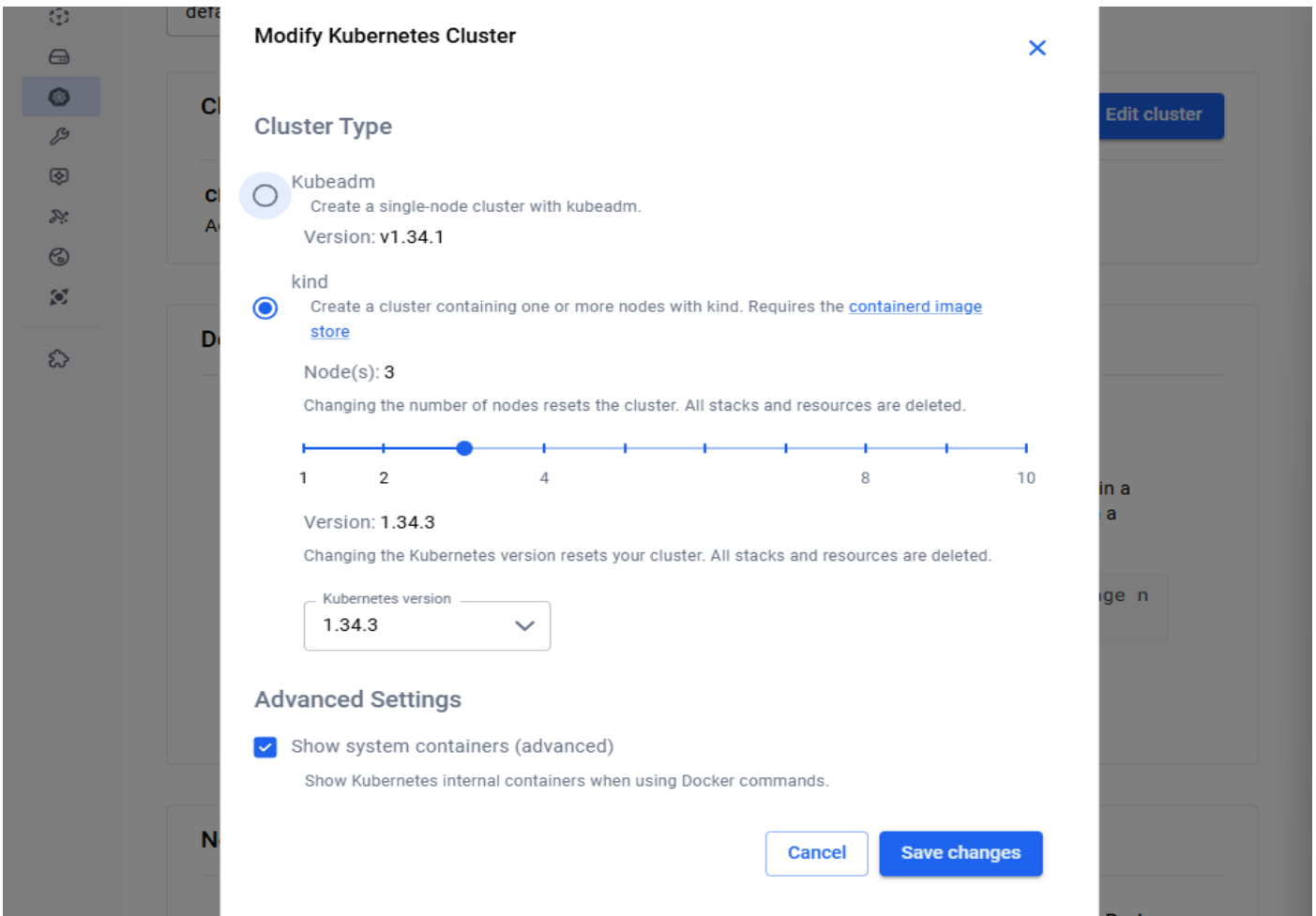
➤ Mon cluster est maintenant créé :



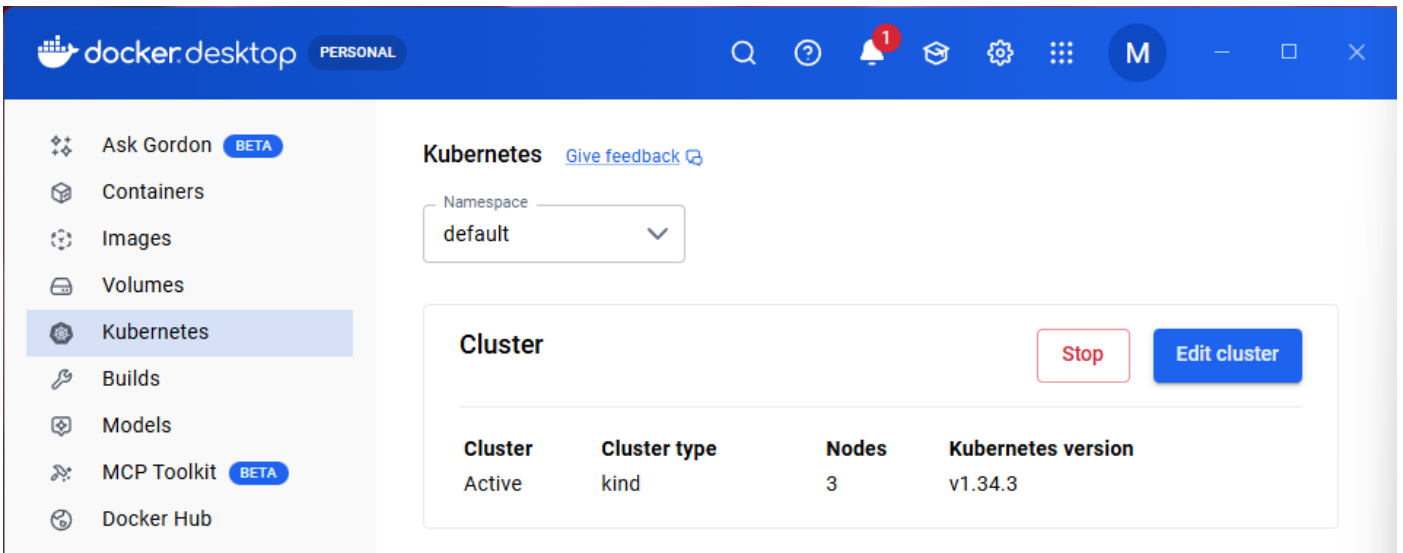
➤ Je peux constater la présence des deux nœuds un maître et un worker :

```
Windows PowerShell
PS C:\Users\Administrateur> kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
desktop-control-plane Ready    control-plane 29s   v1.34.3
desktop-worker      Ready    <none>    13s   v1.34.3
PS C:\Users\Administrateur>
```

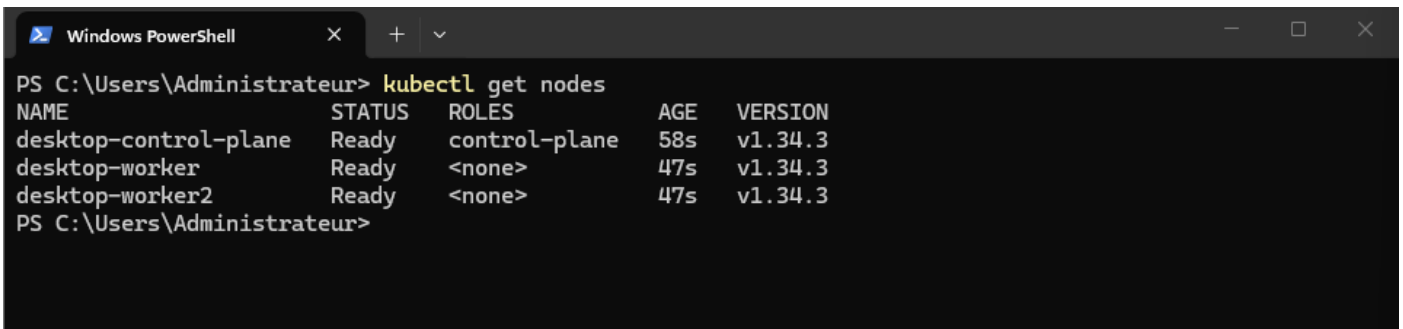
➤ Je retourne dans la configuration du cluster pour rajouter un 3^{ème} nœud :



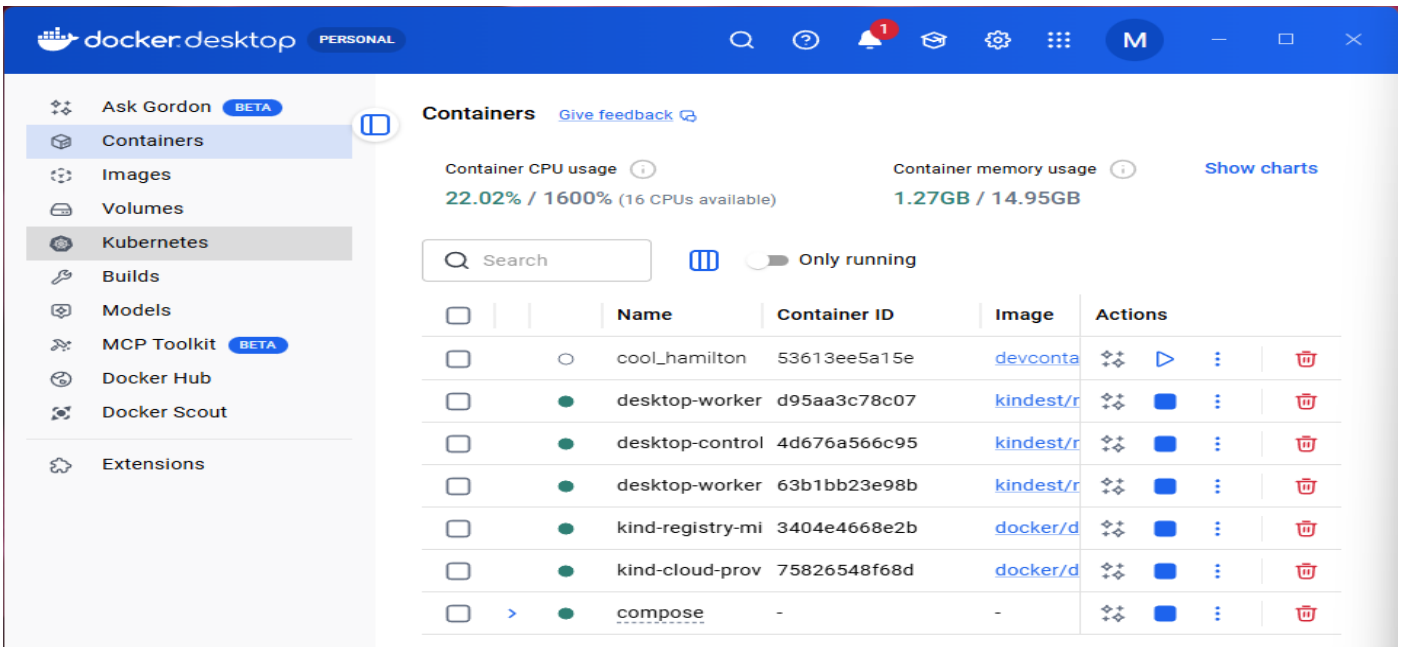
➤ Le 3^{ème} nœud a été créé correctement :



➤ Dans le PowerShell je peux voir le 3^{ème} nœud en tant que worker soit un master et 2 workers :

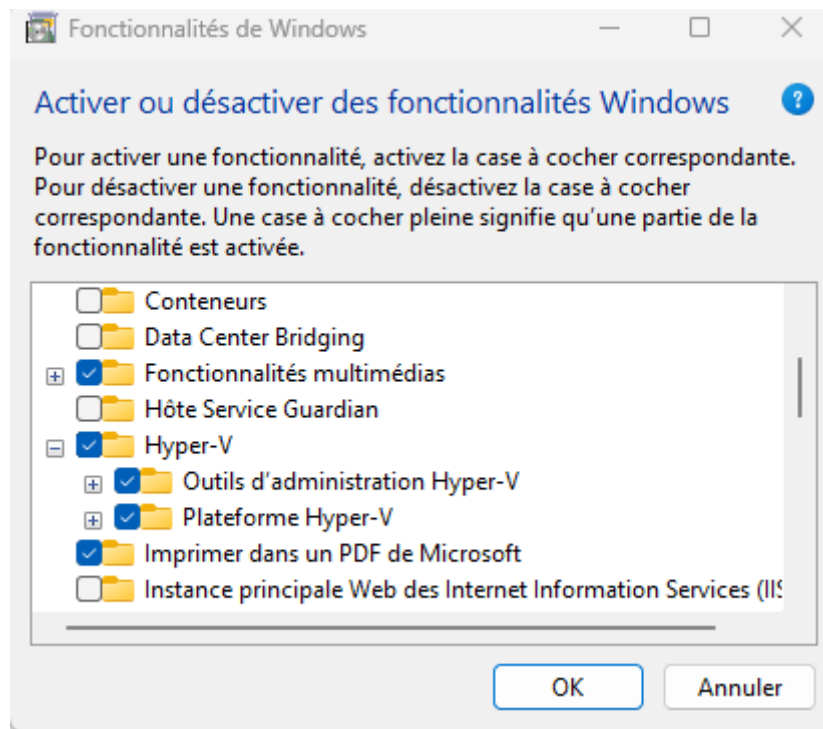


➤ Sous docker desktop les nœuds apparaissent sous forme de containers :



2. Installation de Minikube.

- Dans un premier temps j'active la fonctionnalité Hyper-V ainsi que les sous fonctionnalités :



- Deuxièmement je me rends sur le site de Minikube pour récupérer les commandes à taper dans le PowerShell pour l'installer :

minikube Community GitHub [Try minikube in browser](#)

Search this site...

- Documentation
 - Get Started!
 - Handbook
 - Basic controls
 - Deploying apps
 - Kubectl
 - Accessing apps
 - Addons
 - Configuration
 - Dashboard
 - Pushing images
 - Proxies and VPNs
 - Registries
 - Certificates
 - Offline usage
 - Host access
 - Network Policy
 - Persistent Volumes
 - Mounting filesystems
 - File Sync
 - Troubleshooting
 - FAQ
 - Presentations
 - Tutorials
 - Commands

Operating system

Linux macOS **Windows** Try in Browser

Architecture

x86-64

Release type

Stable

Installer type

.exe download Windows Package Manager Chocolatey

To install the latest minikube **stable** release on **x86-64 Windows** using **.exe download**:

1. Download and run the installer for the [latest release](#).
Or if using PowerShell, use this command:

```

curl -s -L https://github.com/kubernetes/minikube/releases/latest/download/minikube-windows-amd64.exe -UseBasicParsing
```

2. Add the minikube.exe binary to your PATH.
Make sure to run PowerShell as Administrator.

```

$oldPath = [Environment]::GetEnvironmentVariable('Path', [EnvironmentVariableTarget]::User)
if ($oldPath.Split(';') -notcontains 'C:\minikube'){
  [Environment]::SetEnvironmentVariable('Path', $('{0};C:\minikube' -f $oldPath), [EnvironmentVariableTarget]::User)
}
```

If you used a terminal (like powershell) for the installation, please close the terminal and reopen it before running minikube.

➤ Je les colle dans le PowerShell :

```
PS C:\WINDOWS\system32> New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d-----          29/04/2026   15:30           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]::Machine)
>> }
PS C:\WINDOWS\system32>
```

➤ Ensuite je démarre Minikube :

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

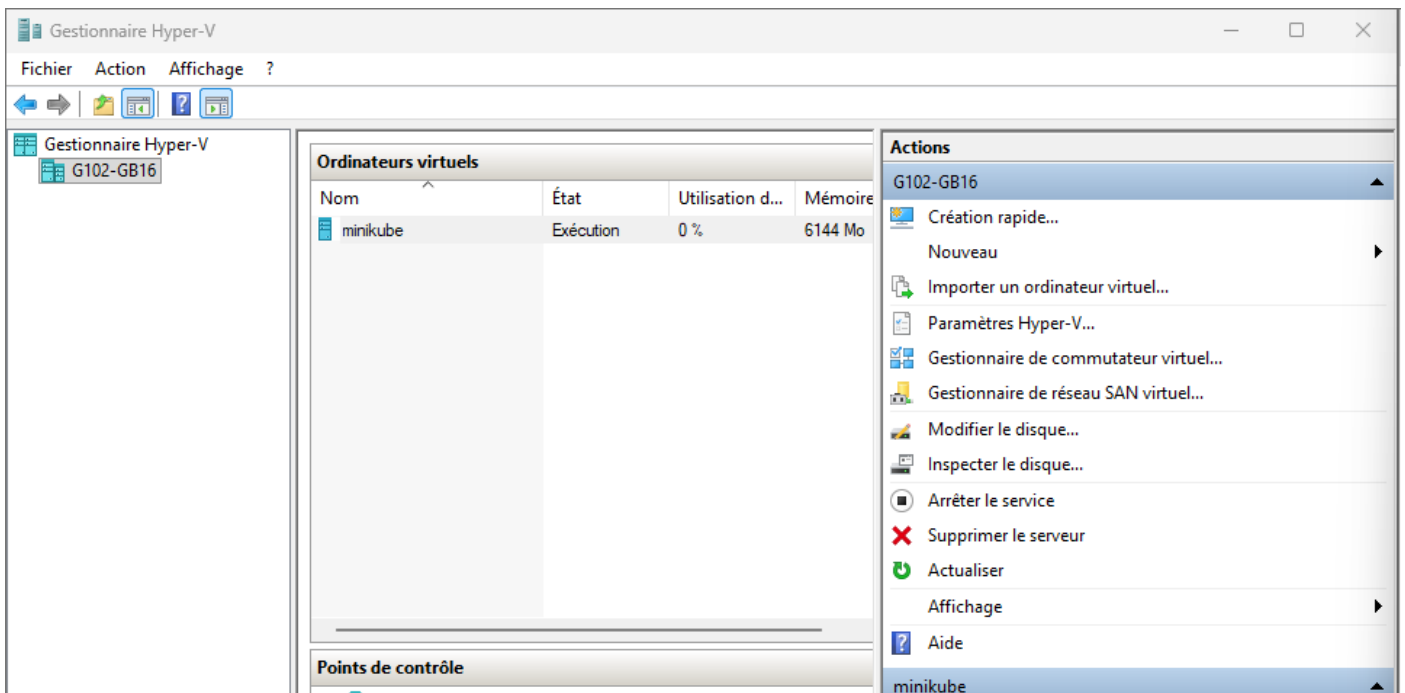
Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\Administrateur> minikube start
🐹 minikube v1.38.1 sur Microsoft Windows 11 Pro Education 25H2
🌟 Choix automatique du pilote docker. Autres choix: virtualbox, ssh
❗ Starting v1.39.0, minikube will default to "containerd" container runtime. See #21973 for more info.
🚀 Utilisation du pilote Docker Desktop avec le privilège root
👉 Démarrage du nœud "minikube" primary control-plane dans le cluster "minikube"
📦 Extraction de l'image de base v0.0.50...
📦 Téléchargement du préchargement de Kubernetes v1.35.1...
> preloaded-images-k8s-v18-v1...: 272.45 MiB / 272.45 MiB 100.00% 41.11 M
> gcr.io/k8s-minikube/kicbase...: 519.58 MiB / 519.58 MiB 100.00% 34.25 M
🔥 Création de docker container (CPU=2, Memory=8000Mo) ...
🌐 Préparation de Kubernetes v1.35.1 sur Docker 29.2.1...
🔗 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\Administrateur> |
```

➤ Je consulte les nœuds :

```
Windows PowerShell
PS C:\Users\Administrateur> kubectl get nodes
NAME        STATUS    ROLES    AGE   VERSION
minikube    Ready    control-plane   57s   v1.35.1
PS C:\Users\Administrateur>
```

- Je me rends dans le gestionnaire Hyper-V et je peux constater la présence de la VM minikube :



- Je liste les objets :

```
PS C:\Users\Administrateur> kubectl get all
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes  ClusterIP     10.96.0.1     <none>         443/TCP    3m5s
PS C:\Users\Administrateur>
```

- Je créer un pod Nginx via kubectl :

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

- Je vérifie la présence du pod :

```
PS C:\Users\Administrateur> kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
monpod  1/1     Running   0          20s
PS C:\Users\Administrateur>
```

```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\Administrateur> kubectl get all
NAME          READY   STATUS    RESTARTS   AGE
pod/monpod    1/1     Running   0           47s

NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes ClusterIP    10.96.0.1    <none>        443/TCP    5m37s
PS C:\Users\Administrateur>
```

➤ Je consulte le log d'un pod :

```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\Administrateur> 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
2026/04/29 13:39:20 [notice] 1#1: using the "epoll" event method
2026/04/29 13:39:20 [notice] 1#1: nginx/1.29.8
2026/04/29 13:39:20 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2026/04/29 13:39:20 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2
2026/04/29 13:39:20 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2026/04/29 13:39:20 [notice] 1#1: start worker processes
2026/04/29 13:39:20 [notice] 1#1: start worker process 29
2026/04/29 13:39:20 [notice] 1#1: start worker process 30
2026/04/29 13:39:20 [notice] 1#1: start worker process 31
2026/04/29 13:39:20 [notice] 1#1: start worker process 32
2026/04/29 13:39:20 [notice] 1#1: start worker process 33
2026/04/29 13:39:20 [notice] 1#1: start worker process 34
2026/04/29 13:39:20 [notice] 1#1: start worker process 35
2026/04/29 13:39:20 [notice] 1#1: start worker process 36
2026/04/29 13:39:20 [notice] 1#1: start worker process 37
2026/04/29 13:39:20 [notice] 1#1: start worker process 38
2026/04/29 13:39:20 [notice] 1#1: start worker process 39
2026/04/29 13:39:20 [notice] 1#1: start worker process 40
2026/04/29 13:39:20 [notice] 1#1: start worker process 41
2026/04/29 13:39:20 [notice] 1#1: start worker process 42
2026/04/29 13:39:20 [notice] 1#1: start worker process 43
2026/04/29 13:39:20 [notice] 1#1: start worker process 44
PS C:\Users\Administrateur>
```

➤ Je consulte les espaces de noms :

```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\Administrateur> kubectl get namespace
NAME          STATUS   AGE
default       Active  6m18s
kube-node-lease Active  6m18s
kube-public   Active  6m18s
kube-system   Active  6m18s
PS C:\Users\Administrateur>
```

- La commande pour consulter les objets peut être écrite au singulier ou au pluriel :

```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\Administrateur> kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
monpod    1/1     Running   0           106s
PS C:\Users\Administrateur> kubectl get pod
NAME      READY   STATUS    RESTARTS   AGE
monpod    1/1     Running   0           108s
PS C:\Users\Administrateur>
```

```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\Administrateur> kubectl get nodes
NAME      STATUS   ROLES    AGE   VERSION
minikube  Ready   control-plane  7m   v1.35.1
PS C:\Users\Administrateur> kubectl get node
NAME      STATUS   ROLES    AGE   VERSION
minikube  Ready   control-plane  7m2s  v1.35.1
PS C:\Users\Administrateur>
```

- Je peux renseigner des champs supplémentaires :

```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\Administrateur> kubectl get node -o wide
NAME      STATUS   ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE
          KERNEL-VERSION   CONTAINER-RUNTIME
minikube  Ready   control-plane  7m22s  v1.35.1  192.168.49.2  <none>        Debian GNU/Linux 12
(bookworm) 6.6.87.2-microsoft-standard-WSL2  docker://29.2.1
PS C:\Users\Administrateur>
```

- J'accède à minikube :

```

Windows PowerShell
PS C:\Users\Administrateur> minikube ssh
Linux minikube 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun  5 18:30:46 UTC 2025 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
docker@minikube:~$ sudo ctr namespace ls
NAME LABELS
moby
docker@minikube:~$ sudo ctr -n moby container ls
CONTAINER                                IMAGE                                RUNTIME
0439a71f4e6ddb693bc5b1bcc0f2488cfa5a7575e06c0fc1bbdb15974f91584b  -                                io.containerd.runc.v2
17355564fdb1b844cdccf0fb16b338fccd6f749a043216301ec5595dc8b73324  -                                io.containerd.runc.v2
1a28ffc00e875bbfa542b05d3c98ef4445b52f442cbe730d69b88c08b4e320db  -                                io.containerd.runc.v2
4ca8d7571926f9a2d764fbbcc0acd734010fe9a66071c65f6888f7a5d72f71f86  -                                io.containerd.runc.v2
64d5efd13beabb5eacc8a7208a295d6a163d36fba4eb81ed7c800d16492629ea  -                                io.containerd.runc.v2
7bf72af05164d7bab8259dc3deb4cd403d53f8119c666f3fe449f4cd7b68a50  -                                io.containerd.runc.v2
7eb34f6c8e4c7be1f62c7c930947778b68b22b3df1c1531b28124edf7b2e1405  -                                io.containerd.runc.v2
834a1c1201b270ee7bfd1c9944716e78c1e567fff4baff0cb403fe9f40a08a15  -                                io.containerd.runc.v2
9dfb633219b53138b1aae343e5fe9f8ae6709b60b77ad4f79760d89a840cc231  -                                io.containerd.runc.v2
c096b66465cda003f466461708e9b3aeb9bade6204afec59757ebba0f02879aa  -                                io.containerd.runc.v2
c8098ab4fc34f578d8d282f7a8263a3db0c5be28396d1180f45270602a020b25  -                                io.containerd.runc.v2
cf3ee241588075d0ce7b7cfa53ae9bbf11d386e866c3d4d35b2d664a9ae8578  -                                io.containerd.runc.v2
d74252b8a38d00dd035de2a9c228bebec79e4ddc2d6fd43ed01680c1d0bd04d8  -                                io.containerd.runc.v2
de89aaed3e6eb97f10fd8454130af9724d47036aec634b5848b3646d49179222  -                                io.containerd.runc.v2
e584866d8374a877d41dba38a631450383dc3efddcba91790189b24d09ec98bd  -                                io.containerd.runc.v2
f59efd58c7d08eb5c2d6d8c494518bf9580c703294049d88dea6db4b9cfa0262  -                                io.containerd.runc.v2
docker@minikube:~$ exit
logout
PS C:\Users\Administrateur>

```

➤ Je consulte la description d'un objet :

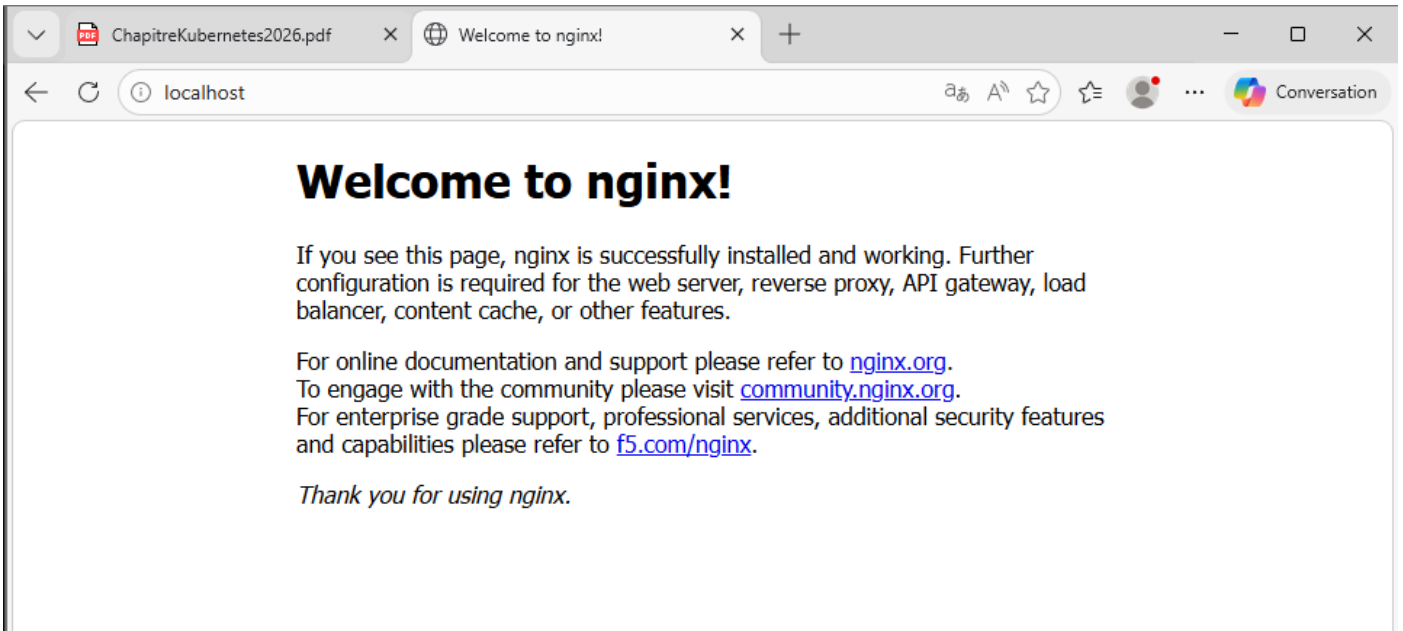
```

Windows PowerShell
PS C:\Users\Administrateur> kubectl describe pod/monpod
Name:                monpod
Namespace:           default
Priority:             0
Service Account:    default
Node:                minikube/192.168.49.2
Start Time:          Wed, 29 Apr 2026 15:39:12 +0200
Labels:              run=monpod
Annotations:         <none>
Status:              Running
IP:                  10.244.0.3
IPs:
  IP: 10.244.0.3
Containers:
  monpod:
    Container ID:    docker://cf3ee241588075d0ce7b7cfa53ae9bbf11d386e866c3d4d35b2d664a9ae8578
    Image:           nginx:latest
    Image ID:        docker-pullable://nginx@sha256:6e23479198b998e5e25921dff8455837c7636a67111a04a635c
f1bb363d199dc
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 29 Apr 2026 15:39:20 +0200
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-6s89w (ro)
Conditions:
  Type                Status
  PodReadyToStartContainers  True
  Initialized           True
  Ready                 True
  ContainersReady       True
  PodScheduled          True
Volumes:
  kube-api-access-6s89w:
    Type:              Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    Optional:           false
    DownwardAPI:        true

```

- J'accède à l'application Nginx depuis l'extérieur du cluster Kubernetes :

```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\Administrateur> 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
|
```



- Je détruis le pod nginx :

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

3. Le tableau de bord de Kubernetes.

- J'active le plugin dashboard :

```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\Administrateur> 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/OWNERS
  ▪ Utilisation de l'image docker.io/kubernetes/dashboard:v2.7.0
  ▪ Utilisation de l'image docker.io/kubernetes/metrics-scrapers: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

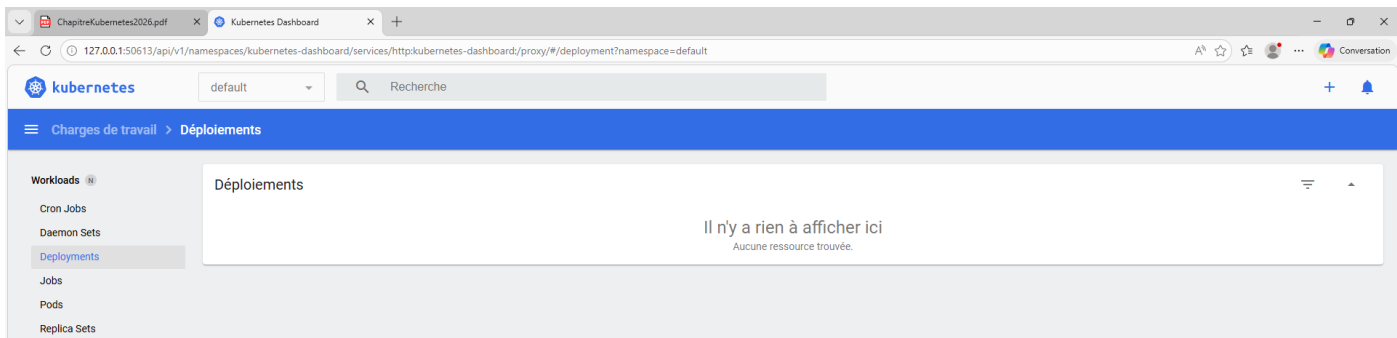
🌟 Le module 'dashboard' est activé
PS C:\Users\Administrateur>
```

- J'active le serveur de métriques :

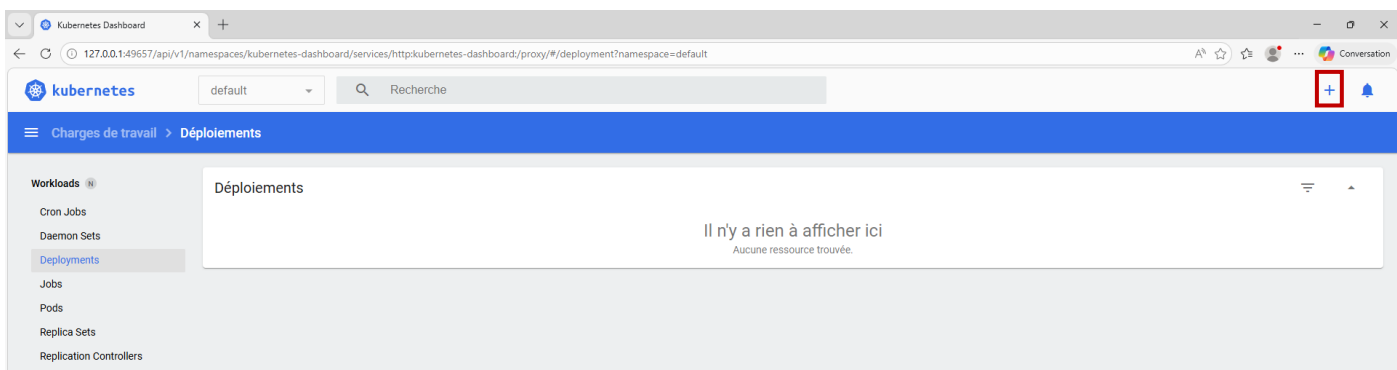
```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\Administrateur> minikube addons enable metrics-server
💡 metrics-server est un addon maintenu par Kubernetes. Pour toute question, contactez minikube sur GitHub.
Vous pouvez consulter la liste des mainteneurs de minikube sur : https://github.com/kubernetes/minikube/blob/master/OWNERS
  ▪ Utilisation de l'image registry.k8s.io/metrics-server/metrics-server:v0.8.1
🌟 Le module 'metrics-server' est activé
PS C:\Users\Administrateur>
```

- J'accède au tableau de bord :

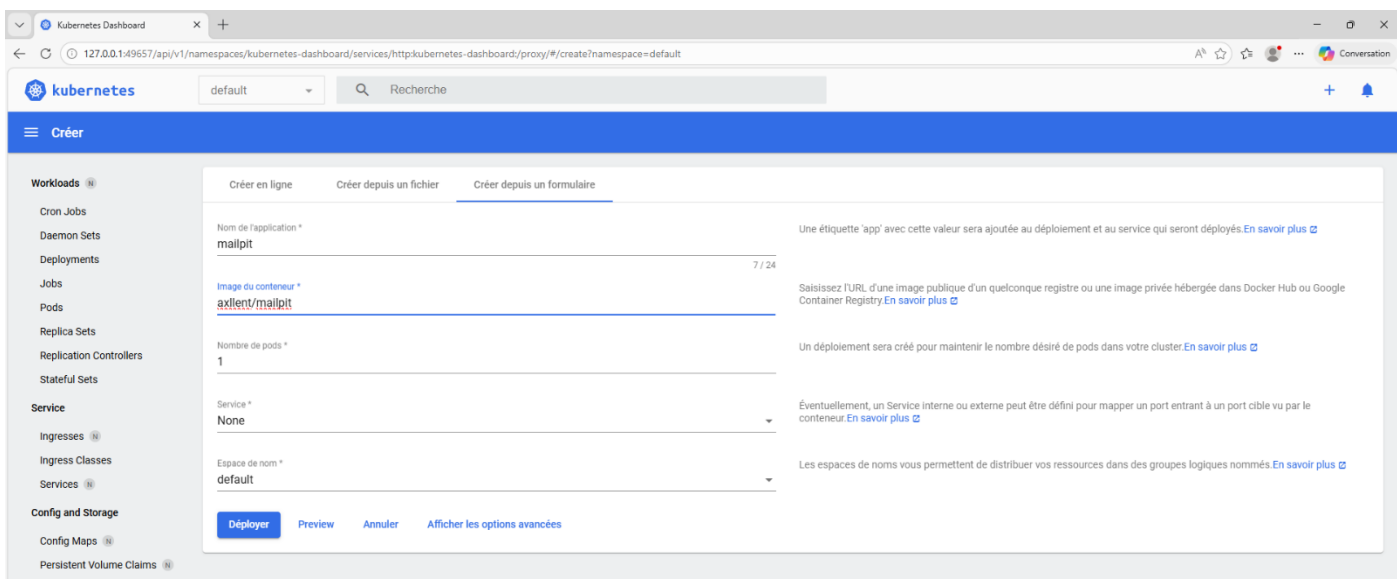
```
Windows PowerShell x Windows PowerShell x + v
PS C:\Users\Administrateur> minikube dashboard
🤖 Vérification de l'état du tableau de bord...
🚀 Lancement du proxy...
🤖 Vérification de l'état du proxy...
🌐 Ouverture de http://127.0.0.1:50613/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ dans votre navigateur par défaut...
```



➤ Je créer un déploiement :



➤ Je renseigne son nom, l'image du conteneur et le nombre de pod :



Overview

Déploiements

Nom	Images	Étiquettes	Pods	Date de création ↑
mailpit	axilent/mailpit	k8s-app: mailpit	1 / 1	28.seconds.ago

Pods

Nom	Images	Étiquettes	Noeud	Statut	Redémarrag	Utilisation CPU (coeurs)	Utilisation mémoire (octets)	Date de création ↑
mailpit-5d8c9cd4-px5r4	axilent/mailpit	k8s-app: mailpit pod-template-hash: 5d8c9cd4	minikube	Running	0	-	-	28.seconds.ago

Charges de travail

Running: 1 Déploiements Running: 1 Pods Running: 1 Replica Sets

Déploiements

Nom	Images	Étiquettes	Pods	Date de création ↑
mailpit	axilent/mailpit	k8s-app: mailpit	1 / 1	a.minute.ago

Pods

Nom	Images	Étiquettes	Noeud	Statut	Redémarrag	Utilisation CPU (coeurs)	Utilisation mémoire (octets)	Date de création ↑
mailpit-5d8c9cd4-px5r4	axilent/mailpit	k8s-app: mailpit pod-template-hash: 5d8c9cd4	minikube	Running	0	0.00%	7.77Mi	a.minute.ago

Replica Sets

Nom	Images	Étiquettes	Pods	Date de création ↑
mailpit-5d8c9cd4	axilent/mailpit	k8s-app: mailpit pod-template-hash: 5d8c9cd4	1 / 1	a.minute.ago

➤ Je me rends dans l'onglets Deployments :

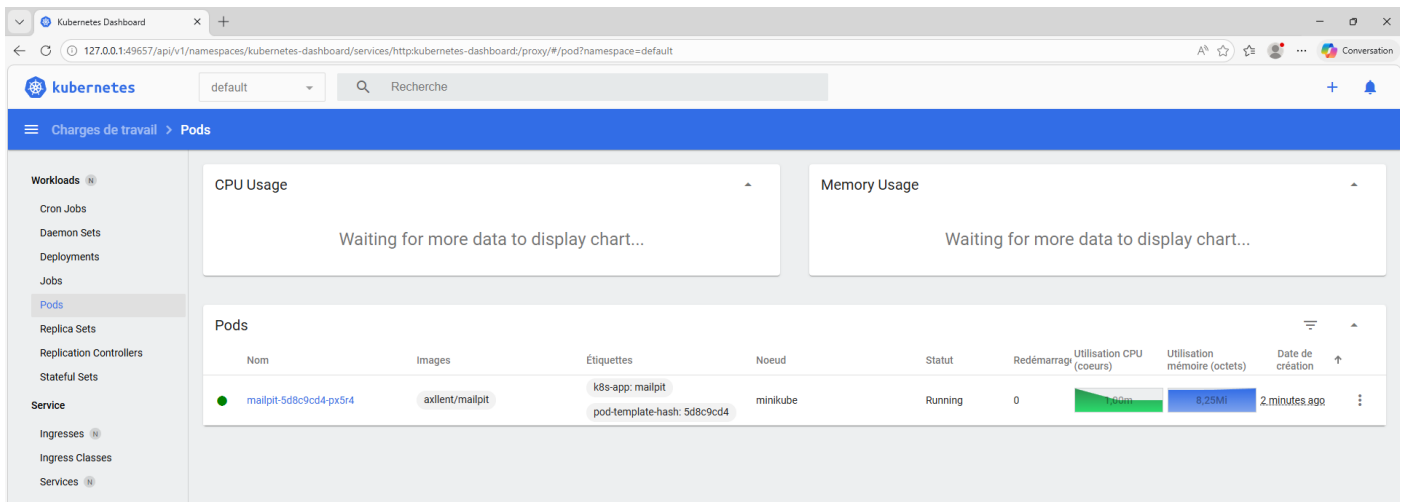
Charges de travail > Déploiements

CPU Usage Memory Usage

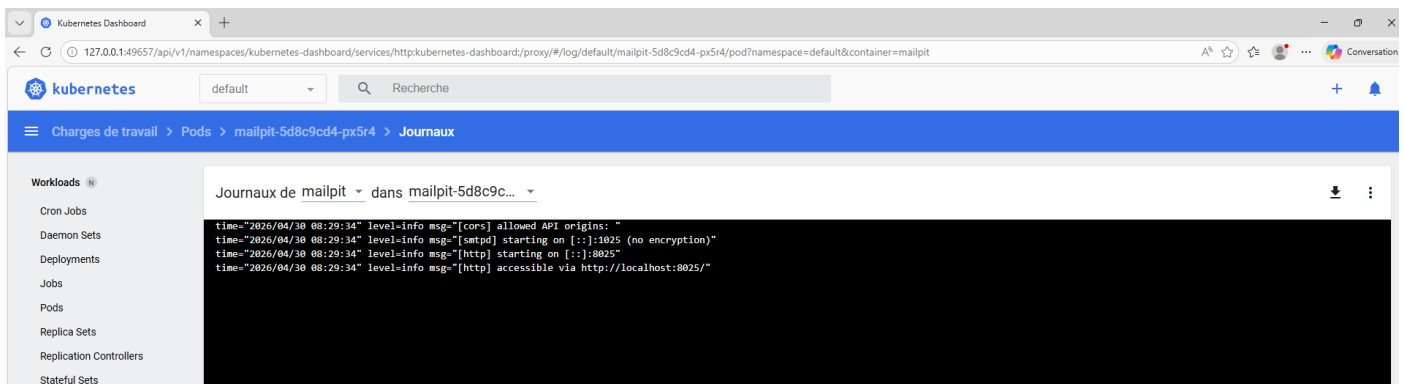
Waiting for more data to display chart... Waiting for more data to display chart...

Déploiements

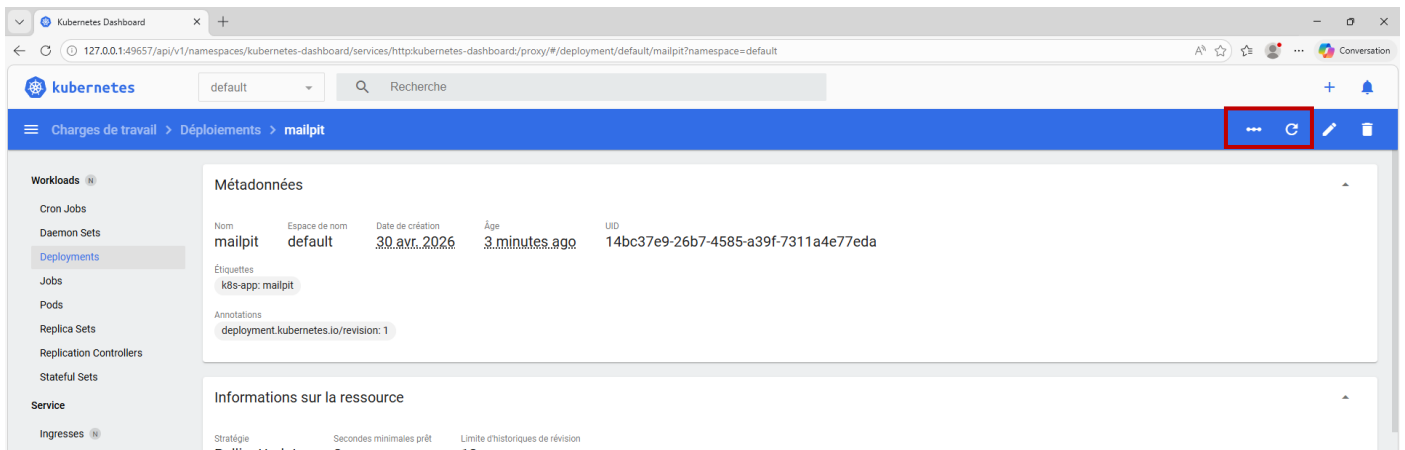
Nom	Images	Étiquettes	Pods	Date de création ↑
mailpit	axilent/mailpit	k8s-app: mailpit	1 / 1	2.minutes.ago

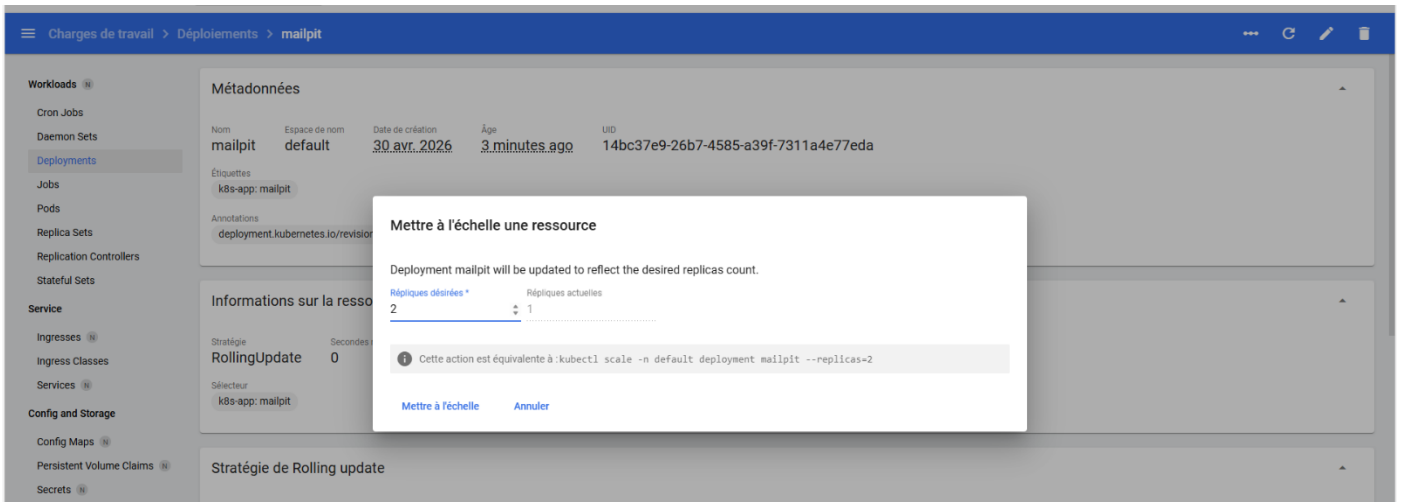


➤ Je me rends dans les journaux :

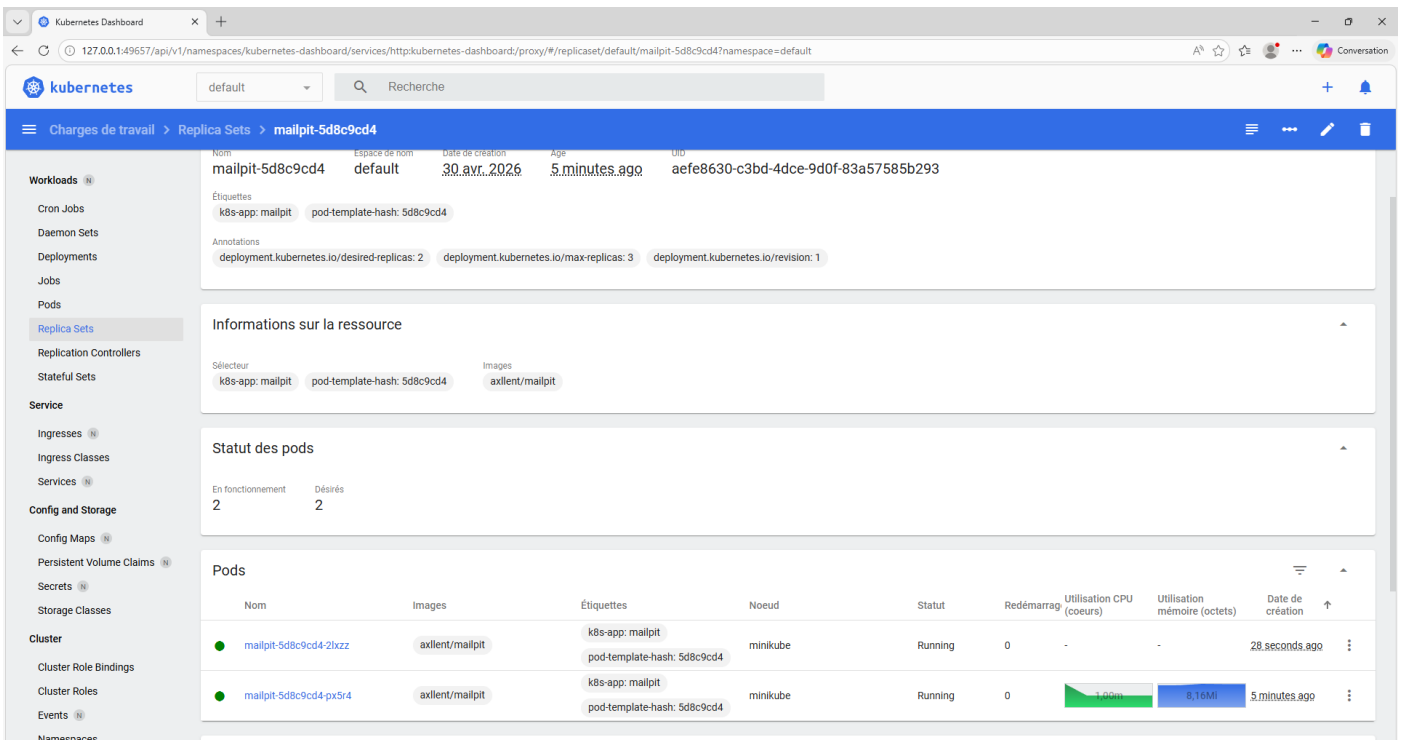


➤ Je peux mettre à l'échelle les ressources :

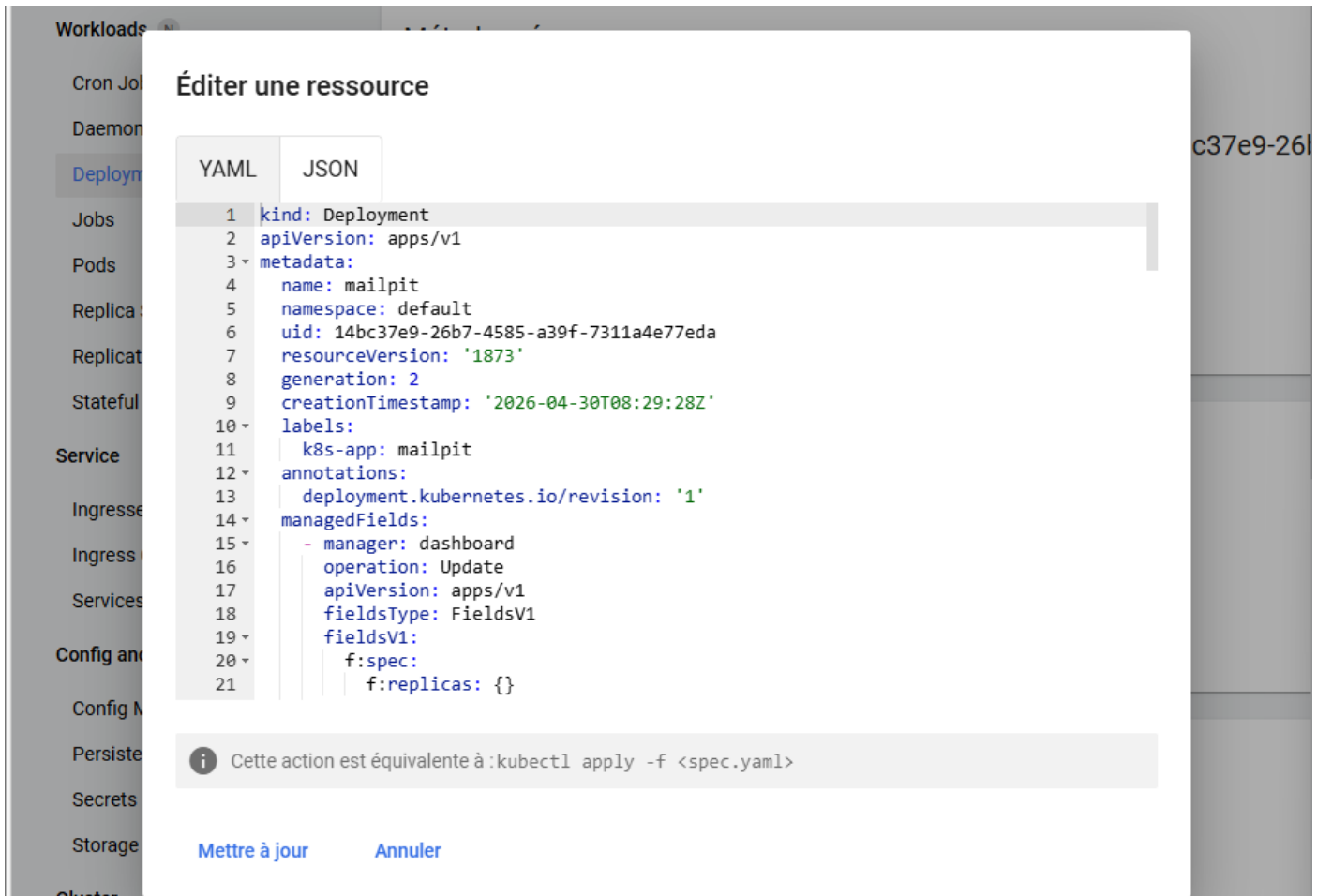
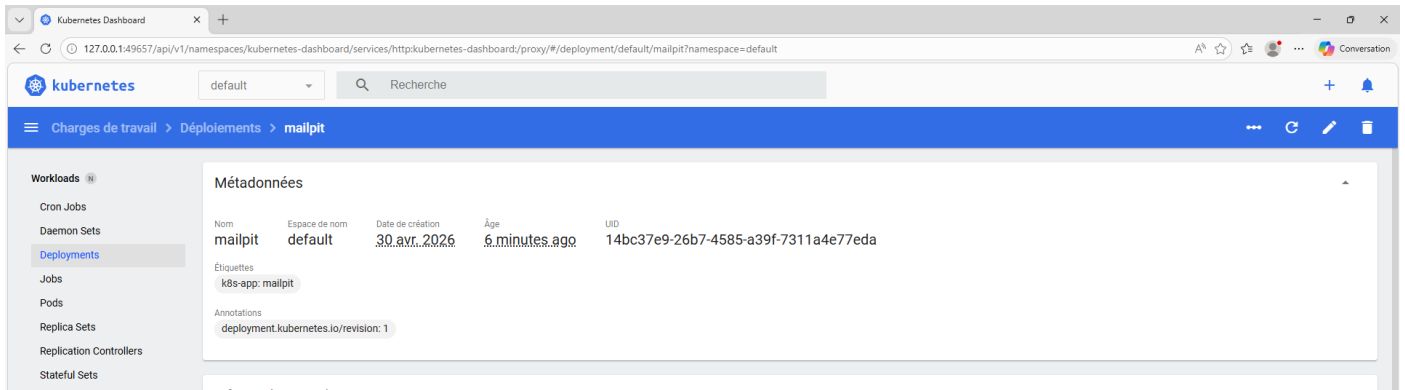




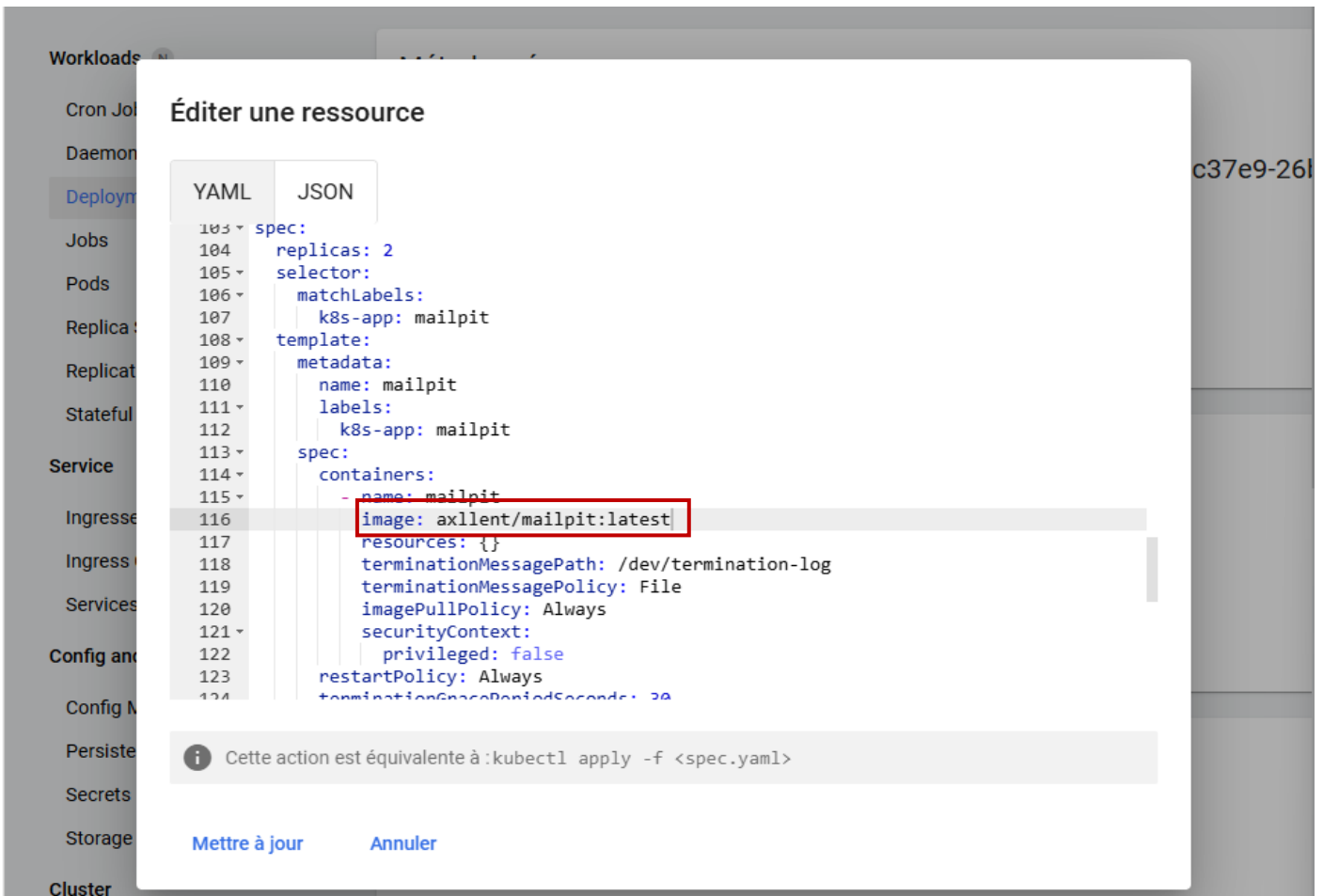
➤ Je constate qu'il y a maintenant 2 pods en fonctionnement :



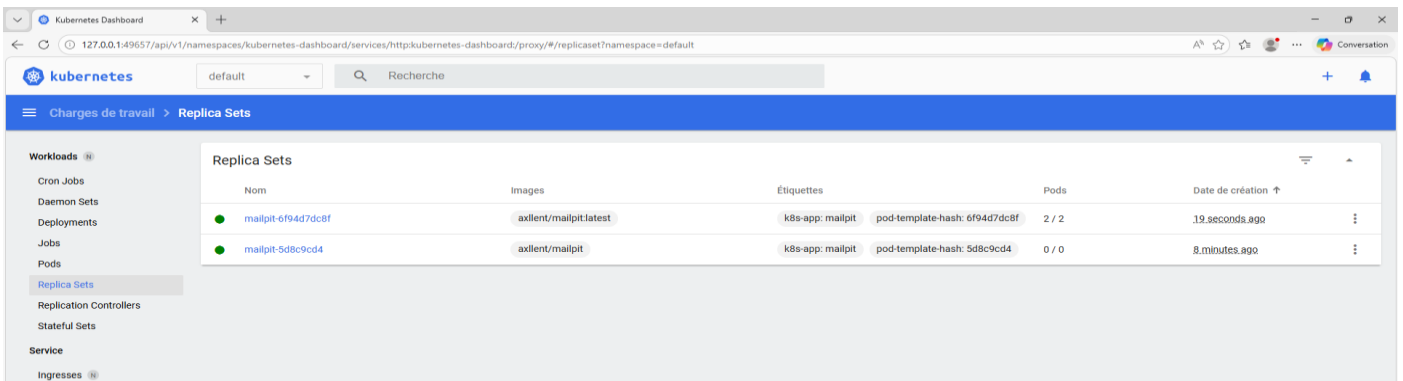
➤ Je constate que les métadonnées ont été mise à jour et que je peux les éditer directement en yaml ou en json :



➤ J'édite l'image du mailpit pour avoir la plus récente (latest) :



➤ Kubernetes va créer un nouveau ReplicaSet :



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

- Je supprime le déploiement mailpit :

```
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:\Users\Administrateur> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mailpit   2/2     2            2           9m56s
PS C:\Users\Administrateur>
```

```
PS C:\Users\Administrateur> kubectl delete deployment mailpit
deployment.apps "mailpit" deleted from default namespace
PS C:\Users\Administrateur>
```

- Le déploiement à été supprimé avec succès :

```
PS C:\Users\Administrateur> kubectl get deployment
No resources found in default namespace.
PS C:\Users\Administrateur>
```

- Je crée un autre déploiement :

```
PS C:\Users\Administrateur> kubectl create deployment mailpit --image=axllent/mailpit
deployment.apps/mailpit created
PS C:\Users\Administrateur>
```

- Je vérifie l'état du déploiement :

```
PS C:\Users\Administrateur> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mailpit   1/1     1            1           25s
PS C:\Users\Administrateur> kubectl get deployment -o wide
NAME      READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES           SELECTOR
mailpit   1/1     1            1           33s   mailpit      axllent/mailpit  app=mailpit
PS C:\Users\Administrateur>
```

```
PS C:\Users\Administrateur> kubectl describe deployment mailpit
Name:          mailpit
Namespace:    default
CreationTimestamp: Thu, 30 Apr 2026 10:49:23 +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  68s   deployment-controller  Scaled up replica set mailpit-7b96b48c5d from 0 to 1
PS C:\Users\Administrateur>
```

- Je consulte la liste des ReplicaSet et j'obtiens la description de mailpit :

```
PS C:\Users\Administrateur> kubectl get replicaset
NAME                DESIRED  CURRENT  READY  AGE
mailpit-7b96b48c5d  1        1        1      94s
PS C:\Users\Administrateur> 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  119s  replicaset-controller  Created pod: mailpit-7b96b48c5d-nx6ng
PS C:\Users\Administrateur>
```

- Je vérifie l'état du pod :

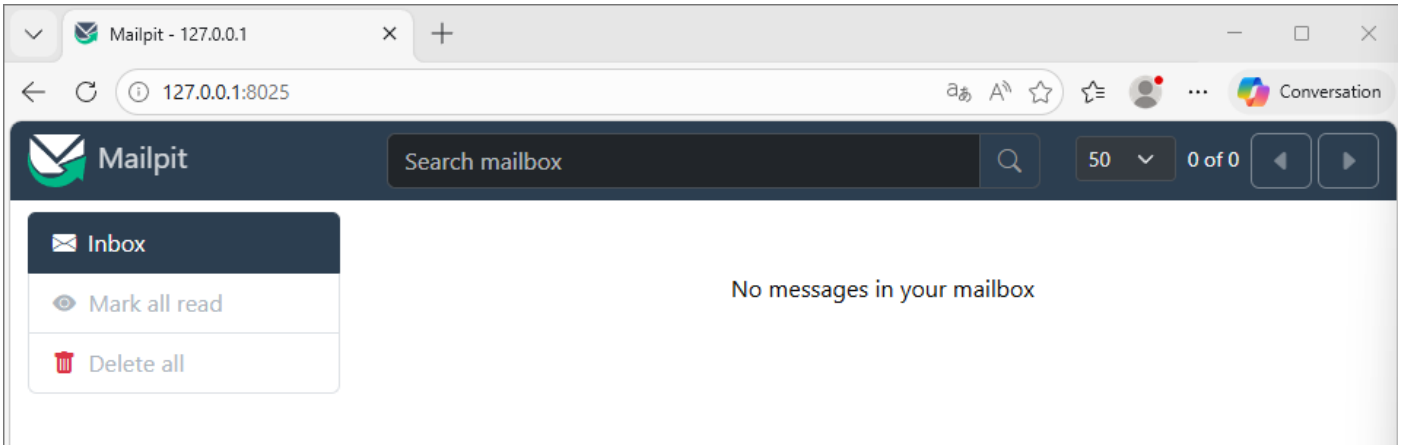
```
PS C:\Users\Administrateur> kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-nx6ng  1/1    Running   0           2m22s
PS C:\Users\Administrateur> kubectl describe pods mailpit-7b96b48c5d-nx6ng
Name:                mailpit-7b96b48c5d-nx6ng
Namespace:           default
Priority:             0
Service Account:     default
Node:                minikube/192.168.49.2
Start Time:          Thu, 30 Apr 2026 10:49:23 +0200
Labels:              app=mailpit
                    pod-template-hash=7b96b48c5d
Annotations:         <none>
Status:              Running
IP:                  10.244.0.17
IPs:
  IP:                10.244.0.17
Controlled By:       ReplicaSet/mailpit-7b96b48c5d
Containers:
  mailpit:
    Container ID:    docker://f536716e07d72e49f3585705c7d744ad2d47d6cb91d7b382d96bed861002d656
    Image:           axllent/mailpit
    Image ID:        docker-pullable://axllent/mailpit@sha256:757f22b56c1da03570afdb3d259effe5091018008a81bbedc8158cee7e16fdbc
    Port:            <none>
    Host Port:       <none>
    State:           Running
      Started:       Thu, 30 Apr 2026 10:49:25 +0200
    Ready:           True
    Restart Count:   0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-2b4h9 (ro)
Conditions:
  Type                    Status
  PodReadyToStartContainers  True
  Initialized              True
  Ready                    True
  ContainersReady          True
  PodScheduled              True
```

➤ J'accède aux logs du pod et des conteneurs :

```
PS C:\Users\Administrateur> kubectl logs mailpit-7b96b48c5d-nx6ng
time="2026/04/30 08:49:25" level=info msg="[smtpd] starting on [::]:1025 (no encryption)"
time="2026/04/30 08:49:25" level=info msg="[cors] allowed API origins: "
time="2026/04/30 08:49:25" level=info msg="[http] starting on [::]:8025"
time="2026/04/30 08:49:25" level=info msg="[http] accessible via http://localhost:8025/"
PS C:\Users\Administrateur> kubectl logs mailpit-7b96b48c5d-nx6ng -c mailpit
time="2026/04/30 08:49:25" level=info msg="[smtpd] starting on [::]:1025 (no encryption)"
time="2026/04/30 08:49:25" level=info msg="[cors] allowed API origins: "
time="2026/04/30 08:49:25" level=info msg="[http] starting on [::]:8025"
time="2026/04/30 08:49:25" level=info msg="[http] accessible via http://localhost:8025/"
PS C:\Users\Administrateur>
```

➤ J'accède à l'application Mailpit :

```
Invite de commandes x Windows PowerShell x Windows PowerShell x + - □ x
PS C:\Users\Administrateur> kubectl port-forward mailpit-7b96b48c5d-nx6ng 8025
Forwarding from 127.0.0.1:8025 -> 8025
Forwarding from [::1]:8025 -> 8025
```



➤ Je crée une entrée DNS stable dans Kubernetes :

```
Invite de commande x Windows PowerShell x Windows PowerShell x Windows PowerShell x + - □ x
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:\Users\Administrateur> kubectl expose deployment/mailpit --port 1025,8025
service/mailpit exposed
PS C:\Users\Administrateur>
```

➤ Je lance une requête DNS depuis le pod de Mailpit en mode interactif :

```
Invite de commande x Windows PowerShell x Windows PowerShell x Windows PowerShell x + - □ x
PS C:\Users\Administrateur> kubectl exec -it mailpit-7b96b48c5d-nx6ng -- sh
/ # getent hosts mailpit
10.104.23.76 mailpit.default.svc.cluster.local mailpit.default.svc.cluster.local mailpit
/ #
```

➤ Je lance un pod de test :

```
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:\Users\Administrateur> 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

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

Name:   mailpit.default.svc.cluster.local
Address: 10.104.23.76

** server can't find mailpit.svc.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\Administrateur>
```

➤ J'augmente la résilience et la scalabilité du déploiement :

```
PS C:\Users\Administrateur> kubectl scale deployment mailpit --replicas=2
deployment.apps/mailpit scaled
PS C:\Users\Administrateur>
```

```
PS C:\Users\Administrateur> kubectl get deployment mailpit
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mailpit   2/2     2            2           7m48s
PS C:\Users\Administrateur> kubectl get pods -l app=mailpit
NAME                                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-gp9g7            1/1    Running   0           40s
mailpit-7b96b48c5d-nx6ng            1/1    Running   0           8m
PS C:\Users\Administrateur>
```

- J'active le contrôleur Ingress dans Minikube :

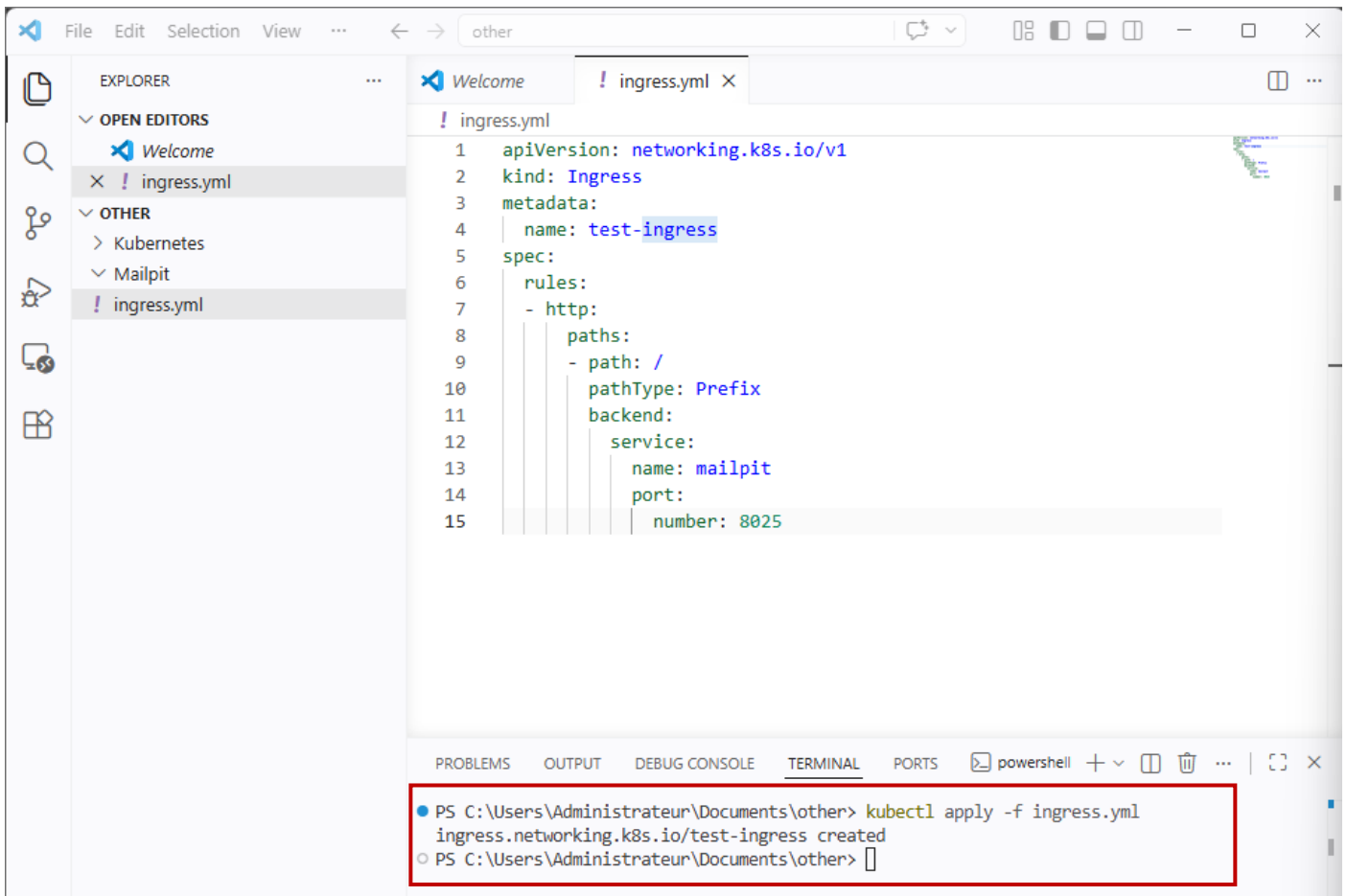
```
PS C:\Users\Administrateur> minikube addons enable ingress
⚠ ingress est un add-on maintenu par Kubernetes. Pour toute question, contactez minikube sur GitHub.
Vous pouvez consulter la liste des mainteneurs de minikube sur : https://github.com/kubernetes/minikube/blob/master/OWNERS
⚠ Après que le module est activé, veuillez exécuter "minikube tunnel" et vos ressources ingress seront disponibles à "127.0.0.1"
  ▪ Utilisation de l'image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.6.7
  ▪ 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
🔍 Vérification du module ingress...
🌟 Le module 'ingress' est activé
PS C:\Users\Administrateur>
```

- Je consulte les espaces de noms je constate la présence de ingress-nginx :

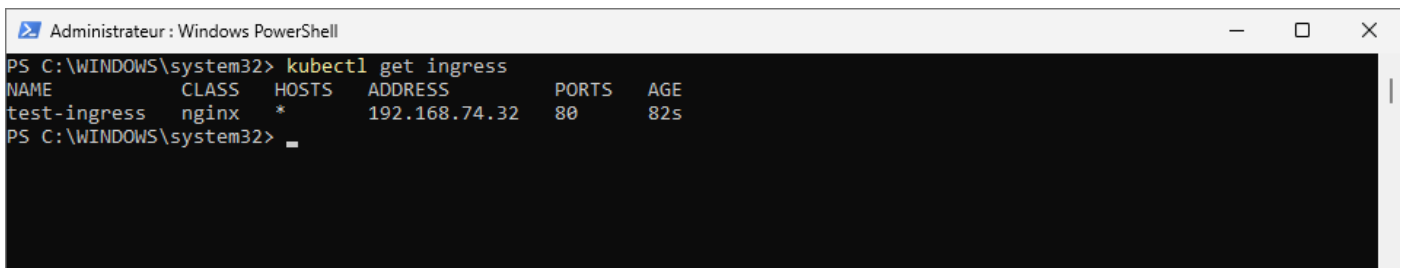
```
PS C:\Users\Administrateur> kubectl get namespace
NAME                STATUS  AGE
default             Active  19h
ingress-nginx       Active  38s
kube-node-lease     Active  19h
kube-public         Active  19h
kube-system         Active  19h
kubernetes-dashboard Active  19h
PS C:\Users\Administrateur>
```

```
PS C:\Users\Administrateur> kubectl -n ingress-nginx get pods -l app.kubernetes.io/name
NAME                                                    READY  STATUS   RESTARTS  AGE
ingress-nginx-admission-create-nlw9t                 0/1    Completed  0          70s
ingress-nginx-admission-patch-qc9zt                 0/1    Completed  1          70s
ingress-nginx-controller-596f8778bc-5vbjb           1/1    Running   0          70s
PS C:\Users\Administrateur>
```

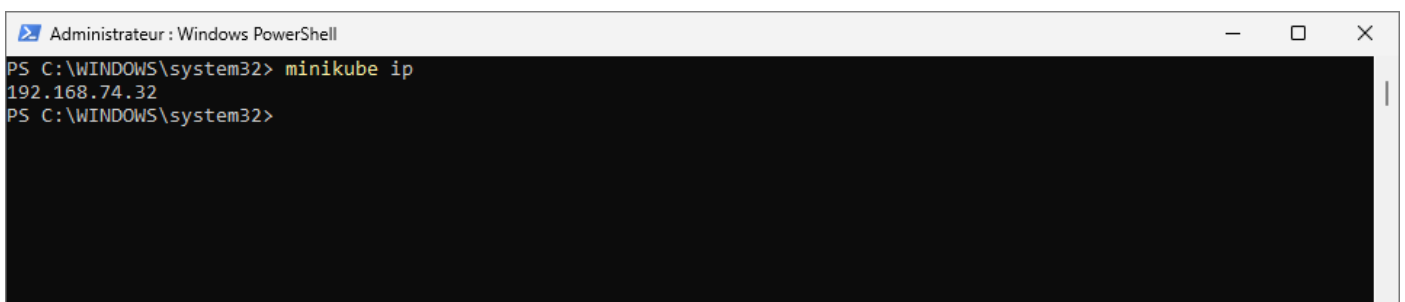
- Je fais la déclaration d'une règle Ingress et prise en compte du fichier YAML :



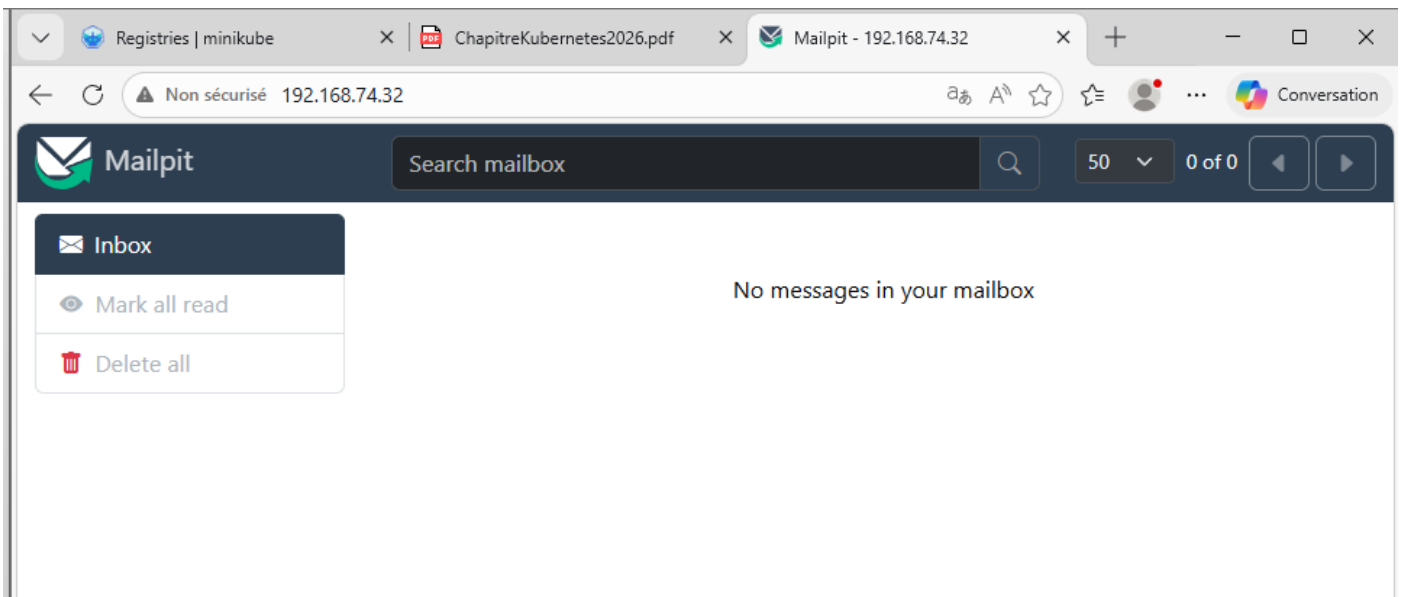
➤ Je regarde l'adresse ip du ingress :



➤ Je récupère l'ip de la minikube :

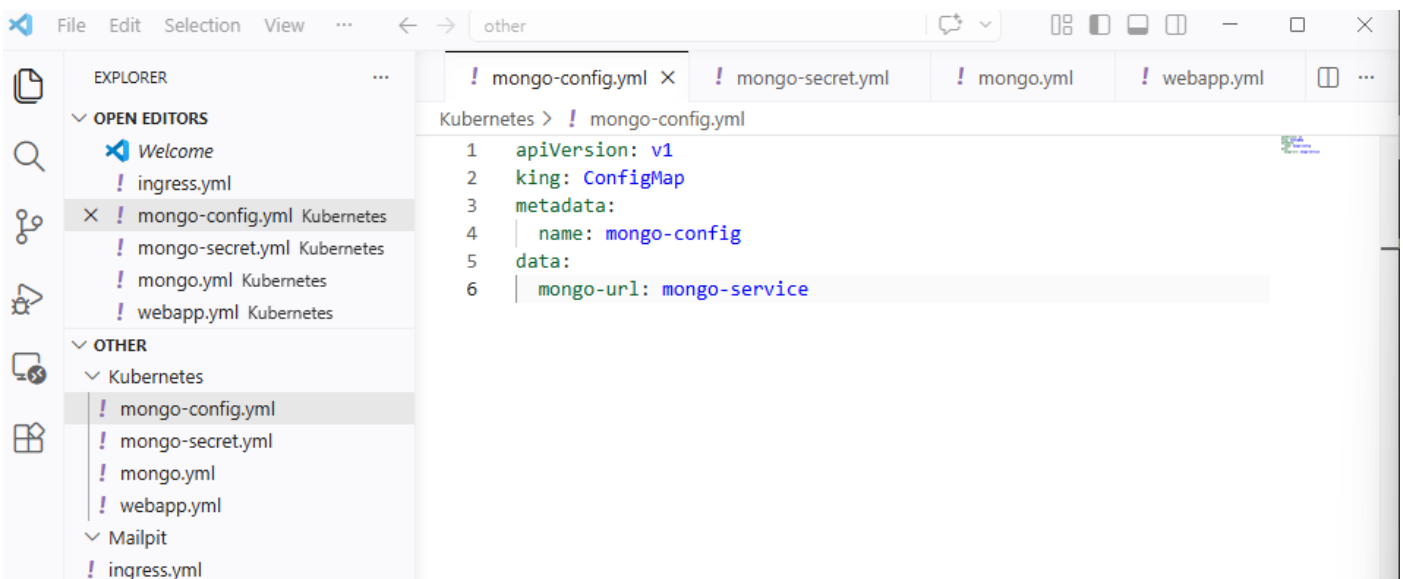


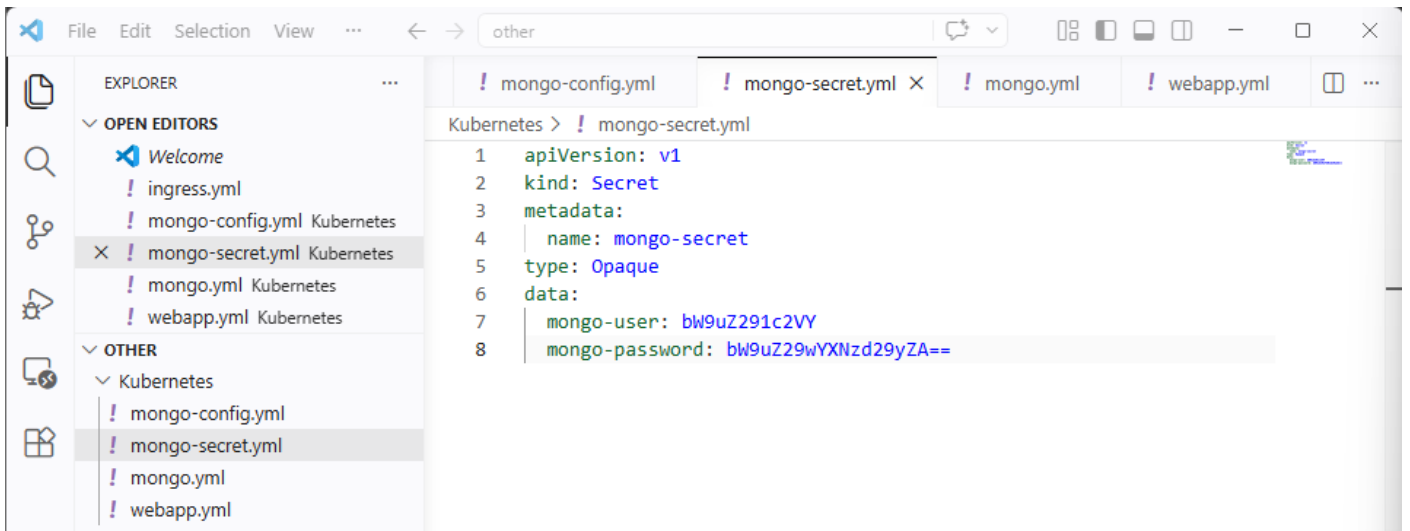
➤ Je me connecte au maipit avec l'ip de la minikube :



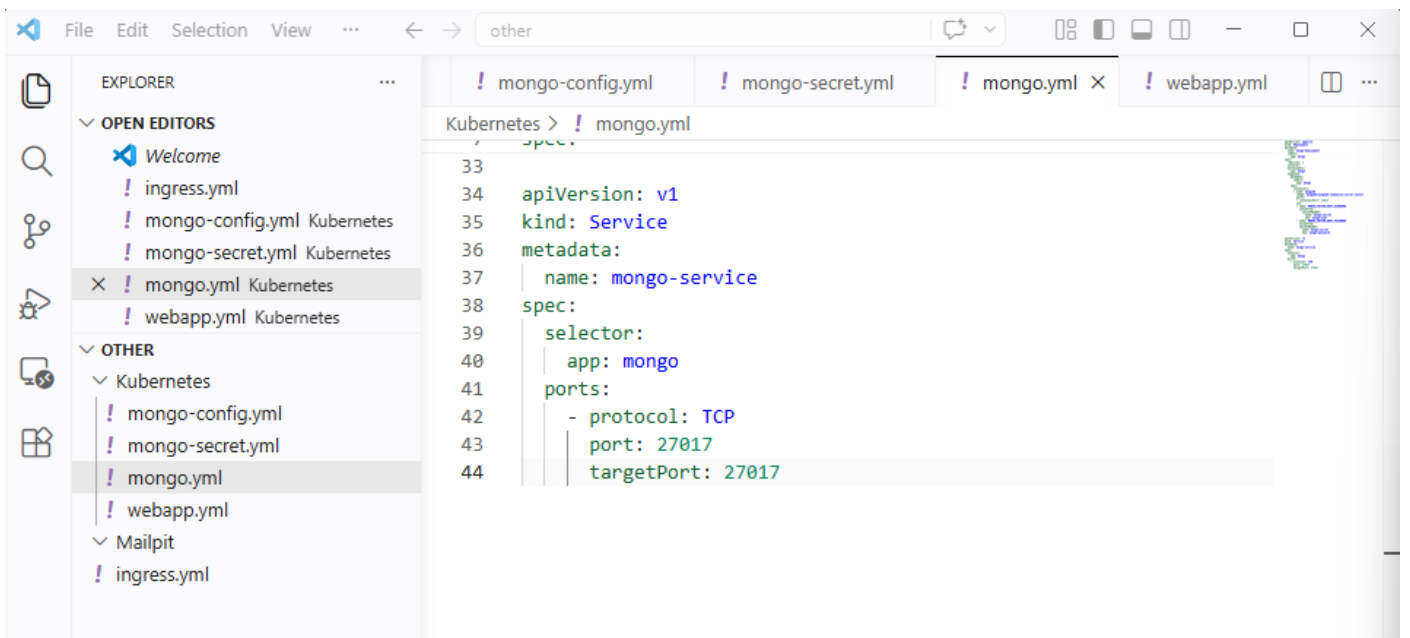
4. Automatisation de déploiement par fichier YAML.

➤ Je crée les fichiers mongo-config.yml, mongo-secret.yml, mongo.yml et webapp.yml :





- Je crée un service qui va permettre la communication entre le pod front et le pod de la base de données :



- Je crée un service externe, pod du front accessible depuis l'extérieur du cluster Kubernetes :

The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows a project structure with folders for 'OPEN EDITORS' and 'OTHER'. The Editor displays the content of 'webapp.yml' in a Kubernetes cluster, which is a Service manifest:

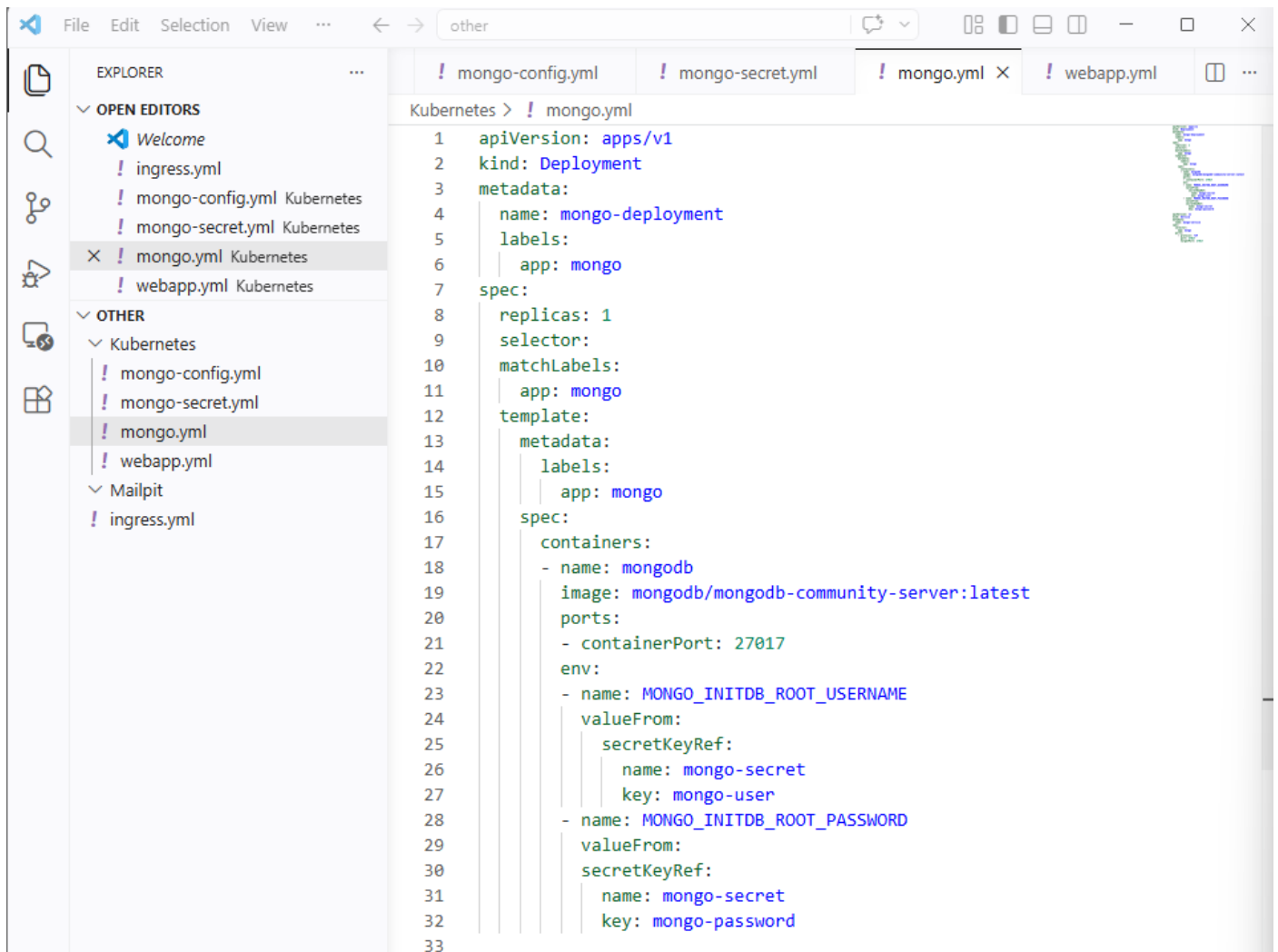
```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: webapp-service
5 spec:
6   type: NodePort
7   selector:
8     app: webapp
9   ports:
10  - protocol: TCP
11    port: 3000
12    targetPort: 3000
13    nodePort: 30100
```

➤ Je déploie l'application grâce au webapp je crée un composant de type deployment :

The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows a project structure with folders for 'OPEN EDITORS' and 'OTHER'. The Editor displays the content of 'webapp.yml' in a Kubernetes cluster, which is a Deployment manifest:

```
1 apiVersion: apps:v1
2 kind: Deployment
3 metadata:
4   name: webapp-deployment
5   labels:
6     app: webapp
7 spec:
8   replicas: 1
9   selector:
10  matchLabels:
11    app: webapp
12  template:
13    metadata:
14      labels:
15        app: webapp
16    spec:
17      containers:
18      - name: webapp
19        image: nanajanashia/k8s-demo-app:v1.0
20        ports:
21        - containerPort: 3000
22        env:
23        - name: USER_NAME
24          valueFrom:
25            secretKeyRef:
26              name: mongo-secret
27              key: mongo-user
28        - name: USER_PWD
29          valueFrom:
30            secretKeyRef:
31              name: mongo-secret
32              key: mongo-password
33        - name: DB_URL
34          valueFrom:
35            configMapKeyRef:
36              name: mongo-config
37              key: mongo-url
```

➤ Je déploie l'application mongo :



The screenshot shows the Visual Studio Code interface with a Kubernetes Deployment manifest for MongoDB. The Explorer sidebar on the left shows the file structure with 'mongo.yml' selected. The main editor displays the following YAML code:

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: mongo-deployment
5   labels:
6     app: mongo
7 spec:
8   replicas: 1
9   selector:
10    matchLabels:
11      app: mongo
12   template:
13     metadata:
14       labels:
15         app: mongo
16     spec:
17       containers:
18         - name: mongodb
19           image: mongodb/mongodb-community-server:latest
20           ports:
21             - containerPort: 27017
22           env:
23             - name: MONGO_INITDB_ROOT_USERNAME
24               valueFrom:
25                 secretKeyRef:
26                   name: mongo-secret
27                   key: mongo-user
28             - name: MONGO_INITDB_ROOT_PASSWORD
29               valueFrom:
30                 secretKeyRef:
31                   name: mongo-secret
32                   key: mongo-password
33
```

➤ Je procède à la création des composants en appliquant la commande `kubectl apply -f` aux 4 fichiers YAML dans l'ordre figurant ci-dessous :

The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a workspace with a folder named 'ingress' containing several files: 'ingress.yml', 'mongo-config.yml', 'mongo-secret.yml', 'mongo.yml', and 'webapp.yml'. The main editor displays the content of 'mongo-config.yml' with the following YAML code:

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

At the bottom, the Terminal pane shows the execution of four 'kubectl apply' commands in a PowerShell session, resulting in the creation of various Kubernetes objects:

```
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl apply -f mongo-config.yml
configmap/mongo-config created
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl apply -f mongo-secret.yml
secret/mongo-secret created
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl apply -f mongo.yml
deployment.apps/mongo-deployment created
service/mongo-service created
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl apply -f webapp.yml
deployment.apps/webapp-deployment created
service/webapp-service created
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> █
```

➤ Je liste les objets créés :

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell + - [] [] ^ X
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/mailpit-7b96b48c5d-9qsvr        1/1     Running   0           11m
pod/mailpit-7b96b48c5d-mb5xq        1/1     Running   0           8m21s
pod/mongo-deployment-744864fdd7-fp9hr 1/1     Running   0           31s
pod/webapp-deployment-5766fd95c7-ch62k 1/1     Running   0           23s

NAME                                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
service/kubernetes                   ClusterIP     10.96.0.1    <none>        443/TCP        1
3m
service/mailpit                       ClusterIP     10.96.74.122 <none>        1025/TCP,8025/TCP 1
0m
service/mongo-service                ClusterIP     10.101.97.196 <none>        27017/TCP      3
1s
service/webapp-service                NodePort      10.96.66.44  <none>        3000:30100/TCP 2
3s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mailpit              2/2     2             2           11m
deployment.apps/mongo-deployment     1/1     1             1           31s
deployment.apps/webapp-deployment    1/1     1             1           23s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/mailpit-7b96b48c5d  2         2         2       11m
replicaset.apps/mongo-deployment-744864fdd7  1         1         1       31s
replicaset.apps/webapp-deployment-5766fd95c7  1         1         1       23s
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress>

```

➤ Je liste les autres objets (configMap et Secret) :

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell + - [] [] ^ X
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl get configmap
NAME          DATA   AGE
kube-root-ca.crt  1       13m
mongo-config    1       65s
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl get secret
NAME          TYPE   DATA   AGE
mongo-secret  Opaque 2       64s
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress>

```

➤ Je décris le service webapp-service :


```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl get pod
NAME READY STATUS RESTARTS AGE
mailpit-7b96b48c5d-9qsvr 1/1 Running 0 15m
mailpit-7b96b48c5d-mb5xq 1/1 Running 0 12m
mongo-deployment-744864fdd7-fp9hr 1/1 Running 0 4m32s
webapp-deployment-5766fd95c7-ch62k 1/1 Running 0 4m24s
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl logs webapp-deployment-5766fd95c7-ch62k
app listening on port 3000!
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl logs webapp-deployment-5766fd95c7-ch62k -f
app listening on port 3000!

```

➤ Je liste les services :

```

PROBLEMS OUTPUT TERMINAL
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> minikube ip
192.168.74.32
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress>

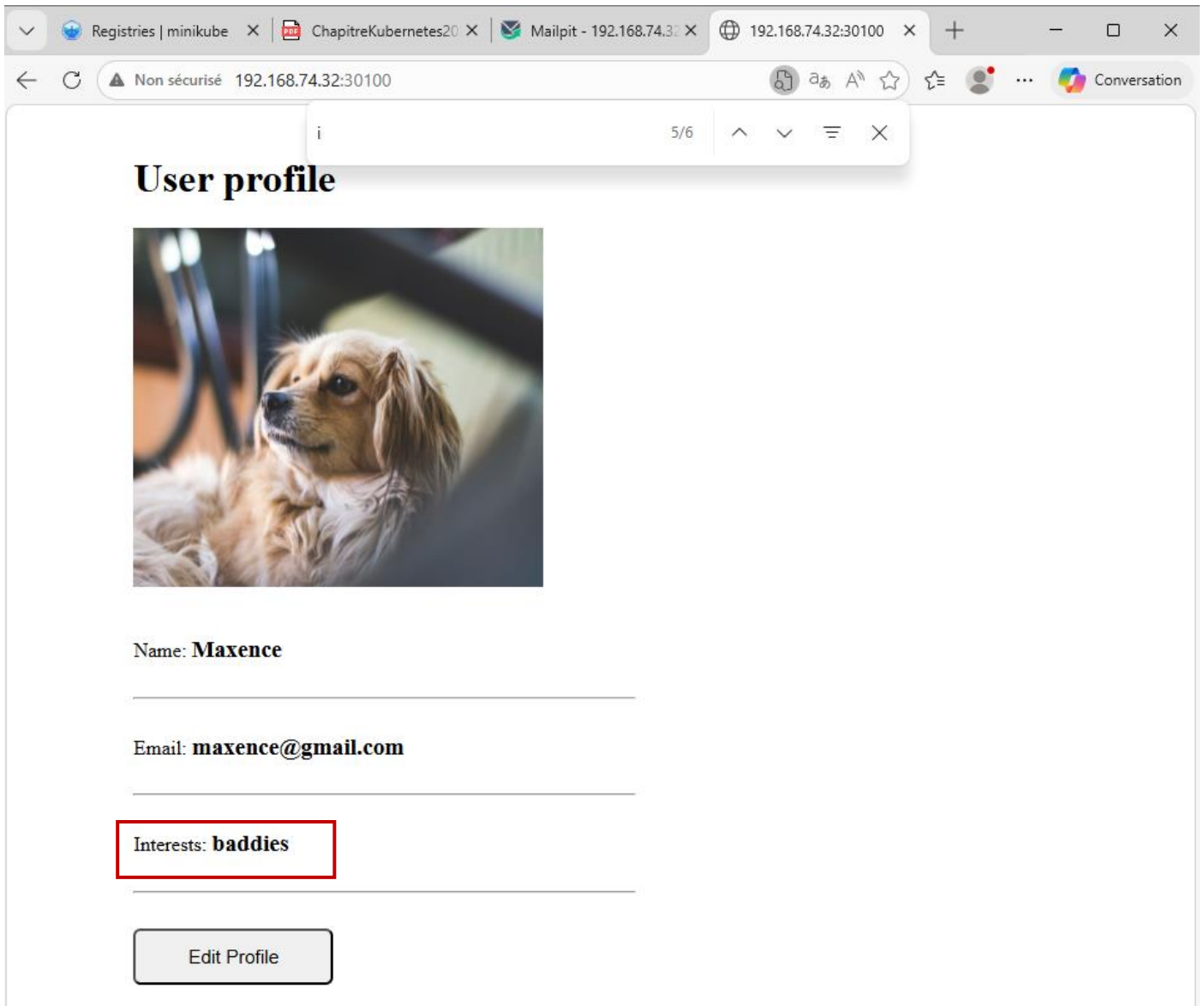
```

```

PROBLEMS OUTPUT TERMINAL
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl get node
NAME STATUS ROLES AGE VERSION
minikube Ready control-plane 20m v1.35.1
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl get node -o wide
NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE
minikube Ready control-plane 20m v1.35.1 192.168.74.32 <none> Buildroot
oot 2025.02 6.6.95 docker://28.5.2
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress>


```


➤ Je modifie le profile :



The screenshot shows a web browser window with several tabs open: 'Registries | minikube', 'ChapitreKubernetes20', 'Mailpit - 192.168.74.32', and '192.168.74.32:30100'. The address bar shows 'Non sécurisé 192.168.74.32:30100'. The page content includes a profile header 'User profile', a profile picture of a dog, and fields for 'Name: Maxence', 'Email: maxence@gmail.com', and 'Interests: baddies'. The 'Interests' field is highlighted with a red border. An 'Edit Profile' button is located at the bottom of the profile section.

User profile



Name: **Maxence**

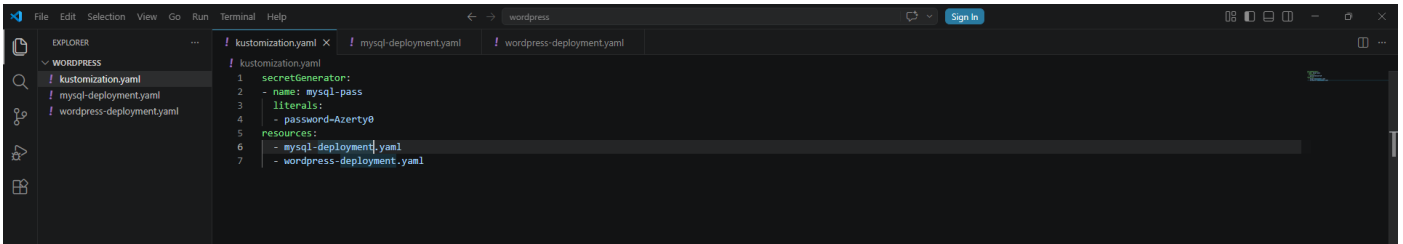
Email: **maxence@gmail.com**

Interests: **baddies**

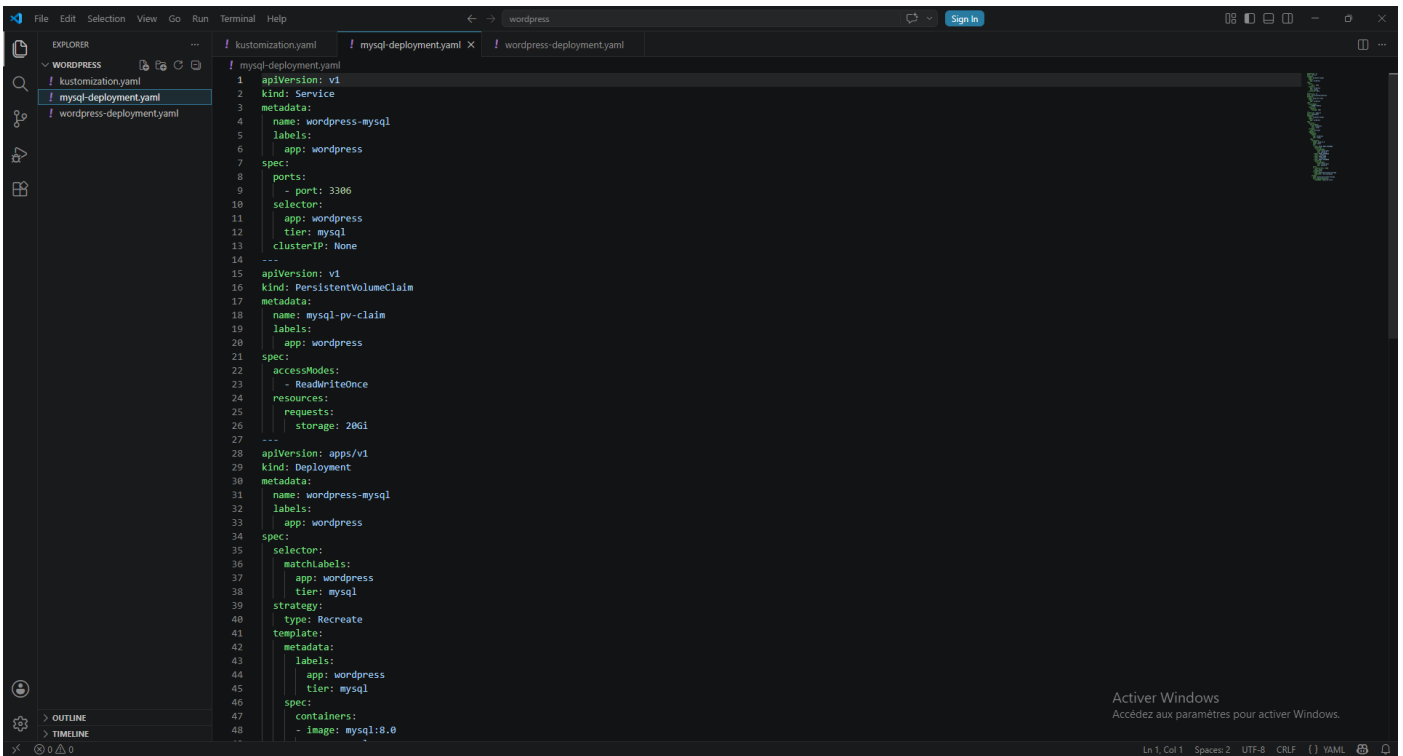
Edit Profile

5. Application.

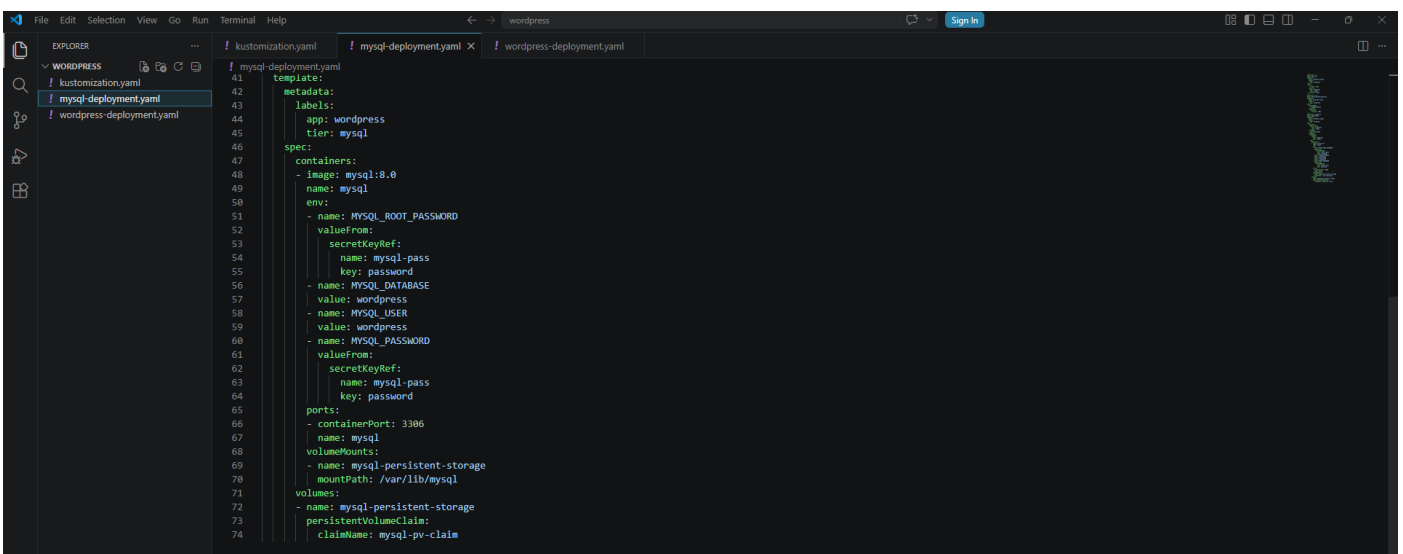
➤ J'ajoute les ressources de configurations pour MySQL et Wordpress :



```
1 secretGenerator:
2   - name: mysql-pass
3     literals:
4       - password-Azerty0
5   resources:
6     - mysql-deployment.yaml
7     - wordpress-deployment.yaml
```



```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: wordpress-mysql
5   labels:
6     app: wordpress
7 spec:
8   ports:
9     - port: 3306
10  selector:
11    app: wordpress
12    tier: mysql
13  clusterIP: None
14 ---
15 apiVersion: v1
16 kind: PersistentVolumeClaim
17 metadata:
18   name: mysql-pv-claim
19   labels:
20     app: wordpress
21 spec:
22   accessModes:
23     - ReadWriteOnce
24   resources:
25     requests:
26       storage: 20Gi
27 ---
28 apiVersion: apps/v1
29 kind: Deployment
30 metadata:
31   name: wordpress-mysql
32   labels:
33     app: wordpress
34 spec:
35   selector:
36     matchLabels:
37       app: wordpress
38       tier: mysql
39   strategy:
40     type: Recreate
41   template:
42     metadata:
43       labels:
44         app: wordpress
45         tier: mysql
46     spec:
47       containers:
48         - image: mysql:8.0
```



```
41 template:
42   metadata:
43     labels:
44       app: wordpress
45       tier: mysql
46   spec:
47     containers:
48       - image: mysql:8.0
49         name: mysql
50         env:
51           - name: MYSQL_ROOT_PASSWORD
52             valueFrom:
53               secretKeyRef:
54                 name: mysql-pass
55                 key: password
56           - name: MYSQL_DATABASE
57             value: wordpress
58           - name: MYSQL_USER
59             value: wordpress
60           - name: MYSQL_PASSWORD
61             valueFrom:
62               secretKeyRef:
63                 name: mysql-pass
64                 key: password
65         ports:
66           - containerPort: 3306
67             name: mysql
68         volumeMounts:
69           - name: mysql-persistent-storage
70             mountPath: /var/lib/mysql
71         volumes:
72           - name: mysql-persistent-storage
73             persistentVolumeClaim:
74               claimName: mysql-pv-claim
```

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: wordpress
5   labels:
6     app: wordpress
7 spec:
8   ports:
9     - port: 80
10  selector:
11    app: wordpress
12    tier: frontend
13  type: LoadBalancer
14 ---
15 apiVersion: v1
16 kind: PersistentVolumeClaim
17 metadata:
18   name: wp-pv-claim
19   labels:
20     app: wordpress
21 spec:
22   accessModes:
23     - ReadWriteOnce
24   resources:
25     requests:
26       storage: 20Gi
27 ---
28 apiVersion: apps/v1
29 kind: Deployment
30 metadata:
31   name: wordpress
32   labels:
33     app: wordpress
34 spec:
35   selector:
36     matchLabels:
37       app: wordpress
38       tier: frontend
39   strategy:
40     type: Recreate
41   template:
42     metadata:
43       labels:
44         app: wordpress
45         tier: frontend
46     spec:
47       containers:
48         - image: wordpress:6.2.1-apache
```

```
51   - name: WORDPRESS_DB_HOST
52     value: wordpress-mysql
53   - name: WORDPRESS_DB_PASSWORD
54     valueFrom:
55       secretKeyRef:
56         name: mysql-pass
57         key: password
58   - name: WORDPRESS_DB_USER
59     value: wordpress
60   ports:
61     - containerPort: 80
62     name: wordpress
63   volumeMounts:
64     - name: wordpress-persistent-storage
65       mountPath: /var/www/html
66   volumes:
67     - name: wordpress-persistent-storage
68       persistentVolumeClaim:
69         claimName: wp-pv-claim
```

➤ Puis j'applique le répertoire pour déployer mon site WordPress :

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a folder named 'WORDPRESS' with files 'kustomization.yaml', 'mysql-deployment.yaml', and 'wordpress-deployment.yaml'. The main editor displays the 'kustomization.yaml' file with the following content:

```
1 secretGenerator:
2   - name: mysql-pass
3     literals:
4       - password=Azerty0
5 resources:
6   - mysql-deployment.yaml
7   - wordpress-deployment.yaml
```

The terminal at the bottom shows the execution of the following command and its output:

```
PS D:\ECOLE\Annee2\b1boucly\kubernetes\wordpress> kubectl apply -k ./
secret/mysql-pass-4m7mfmt2m6 created
service/wordpress created
service/wordpress-mysql created
persistentvolumeclaim/mysql-pv-claim created
persistentvolumeclaim/wp-pv-claim created
deployment.apps/wordpress created
deployment.apps/wordpress-mysql created
PS D:\ECOLE\Annee2\b1boucly\kubernetes\wordpress>
```

➤ Je vérifie que le secret existe :

The screenshot shows the Visual Studio Code interface with the same 'kustomization.yaml' file open. The terminal at the bottom shows the execution of the following command and its output:

```
PS D:\ECOLE\Annee2\b1boucly\kubernetes\wordpress> kubectl get secrets
NAME                TYPE      DATA  AGE
mongo-secret        Opaque    2      15m
mysql-pass-4m7mfmt2m6 Opaque    1      33s
PS D:\ECOLE\Annee2\b1boucly\kubernetes\wordpress>
```

➤ Je vérifie qu'un volume persistant a été provisionné dynamiquement :

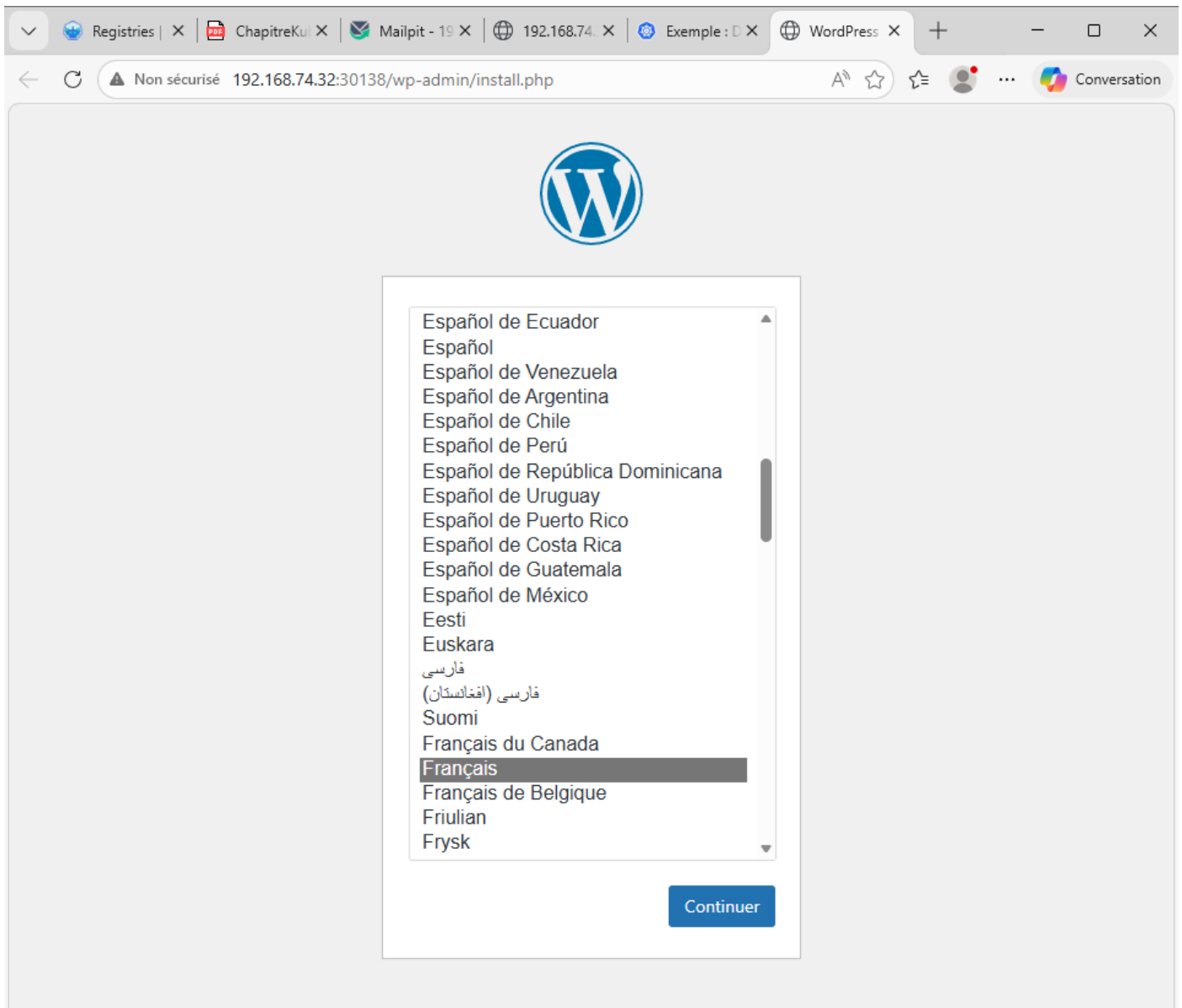
```
kustomization.yaml
1 secretGenerator:
2   - name: mysql-pass
3     literals:
4       - password=Azerty0
5 resources:
6   - mysql-deployment.yaml
7   - wordpress-deployment.yaml
```

```
PS D:\ECOLE\Annee2\b1boucly\kubernetes\wordpress> kubectl get pvc
NAME                                STATUS    VOLUME                                     CAPACITY   ACCESS MODES
mysql-pv-claim                       Bound    pvc-ffb539f6-3fbf-4b5b-9b8a-739f312b26ed  20Gi       RWO
wp-pv-claim                           Bound    pvc-32873449-3676-4707-813e-aa7e0ad7bc17  20Gi       RWO
```

➤ Je vérifie que le pod fonctionne :

```
kustomization.yaml
1 secretGenerator:
2   - name: mysql-pass
3     literals:
4       - password=Azerty0
5 resources:
6   - mysql-deployment.yaml
7   - wordpress-deployment.yaml
```

```
PS D:\ECOLE\Annee2\b1boucly\kubernetes\wordpress> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-9qsvr            1/1    Running   0           27m
mailpit-7b96b48c5d-mb5xq            1/1    Running   0           24m
mongo-deployment-744864fdd7-fp9hr   1/1    Running   0           16m
webapp-deployment-5766fd95c7-ch62k  1/1    Running   0           16m
wordpress-69b66f6445-f4mpg          1/1    Running   0           89s
wordpress-mysql-77f7cdb7df-z8rdk    1/1    Running   0           89s
```

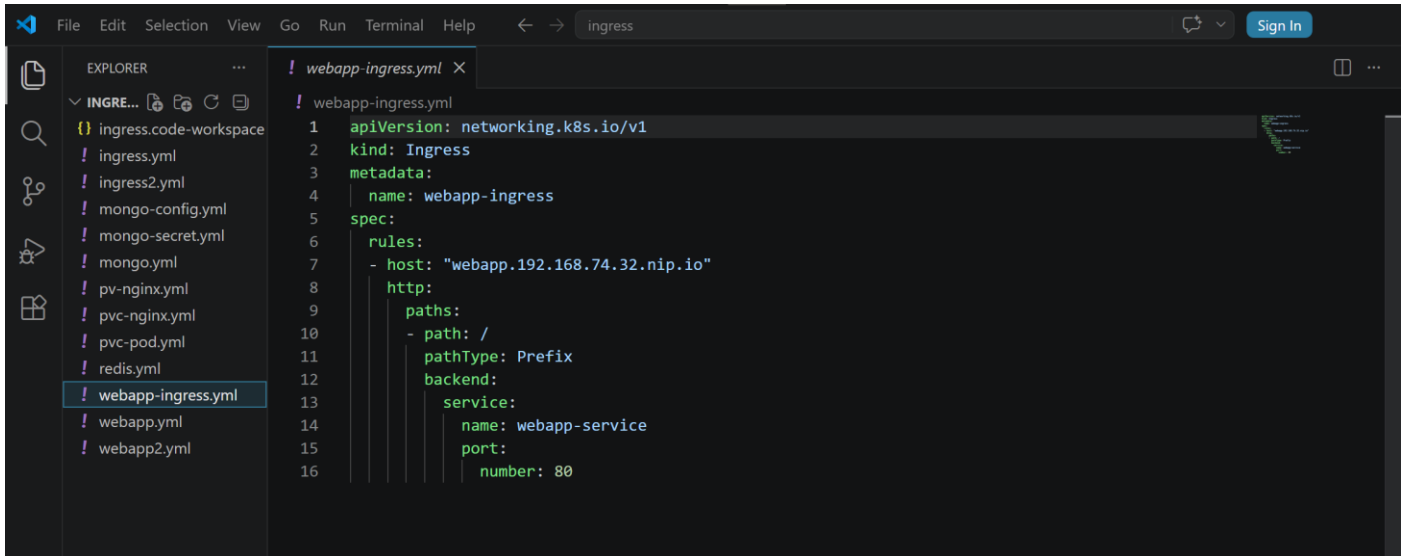



6. Suite Ingress (Mailpit) : hôtes virtuels et nom de domaine nip.io

- Je récupère les informations sur les objets ingress :

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

- Je crée un objet Ingress avec le champ host spécifiant le nom d'hôte virtuel :

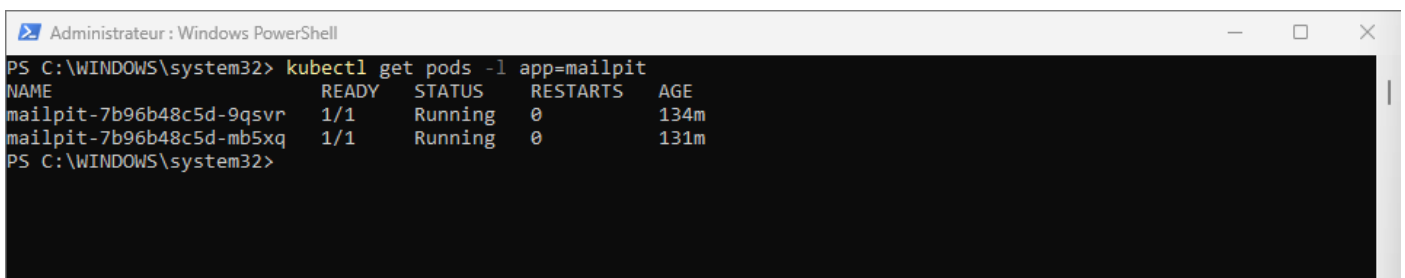


```
1  apiVersion: networking.k8s.io/v1
2  kind: Ingress
3  metadata:
4    name: webapp-ingress
5  spec:
6    rules:
7    - host: "webapp.192.168.74.32.nip.io"
8      http:
9        paths:
10       - path: /
11         pathType: Prefix
12         backend:
13           service:
14             name: webapp-service
15             port:
16               number: 80
```

Je n'ai pas les 2 screens de la suite, car la connexion sur le nip.io n'a pas fonctionné

8. Cycle de vie d'un conteneur dans Kubernetes

- Je consulte l'état des pods de l'application portant le label app=mailpit :



```
PS C:\WINDOWS\system32> kubectl get pods -l app=mailpit
NAME                                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-9qsvr            1/1     Running  0           134m
mailpit-7b96b48c5d-mb5xq            1/1     Running  0           131m
PS C:\WINDOWS\system32>
```

- Je lance à nouveau une connexion sur le conteneur Mailpit :

```
Administrateur : Windows PowerShell
PS C:\WINDOWS\system32> kubectl exec -it deployment/mailpit -- sh
/# ps -ef
PID  USER      TIME  COMMAND
   1   root      0:00  /mailpit
  13   root      0:00  sh
  19   root      0:00  ps -ef
/# mkdir /tmp/test
/# ls -ld /tmp/test
drwxr-xr-x  2 root    root      4096 May  6 13:38 /tmp/test
/# kill 1
/# command terminated with exit code 137
PS C:\WINDOWS\system32>
```

- Je consulte l'état des pods de l'application :

```
Administrateur : Windows PowerShell
PS C:\WINDOWS\system32> kubectl get pods -l app=mailpit
NAME                                READY  STATUS   RESTARTS  AGE
mailpit-7b96b48c5d-9qsvr            1/1    Running  1 (25s ago)  135m
mailpit-7b96b48c5d-mb5xq           1/1    Running  0              132m
PS C:\WINDOWS\system32>
```

- Je lance à nouveau une connexion sur le conteneur Mailpit :

```
Administrateur : Windows PowerShell
PS C:\WINDOWS\system32> 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:\WINDOWS\system32>
```

9. Persistance des données

- Je crée le fichier de configuration du Pod redis.yaml et je crée le pod avec la commande `kubectl apply -f` :


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS kubectl + - [ ] [ ] ... [ ] [ ] X
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl exec -it redis -- /bin/bash
root@redis:/data# echo Hello > test-file
root@redis:/data#
```

➤ J'installe procs puis je liste les processus en cours d'exécution :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS kubectl + - [ ] [ ] ... [ ] [ ] X
root@redis:/data# 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 [130 kB]
Fetched 10.0 MB in 1s (9119 kB/s)
Reading package lists... Done
root@redis:/data#
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS kubectl + - [ ] [ ] ... [ ] [ ] X
root@redis:/data# 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
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS kubectl + - [ ] [ ] ... [ ] [ ] X
root@redis:/data# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
redis     1  0.1  0.3 160376 23552 ?        Ssl  13:42   0:00 redis-server *:6379
root      29  0.0  0.0   4320  3712 pts/0    Ss   13:43   0:00 /bin/bash
root      69  0.0  0.0     0     0 pts/0    Z    13:44   0:00 [dpkg-preconfigu] <defun
root     159  0.0  0.0   6388  3712 pts/0    R+   13:44   0:00 ps aux
root@redis:/data#
```


➤ Je créer un volume persistant de type hostpath :

The screenshot shows the Visual Studio Code interface with a file explorer on the left and a code editor on the right. The file explorer shows a directory named 'INGRESS' containing several YAML files. The code editor is open to 'pv-nginx.yml' and displays the following content:

```
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"
```

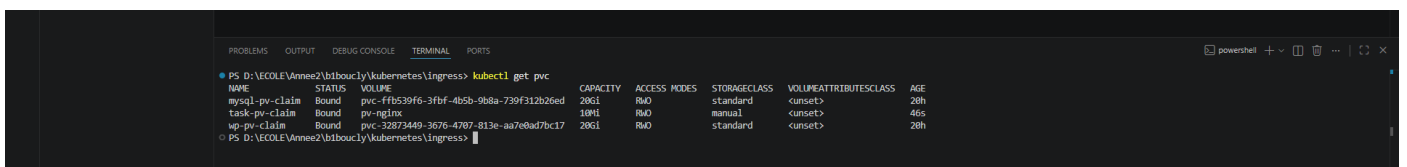
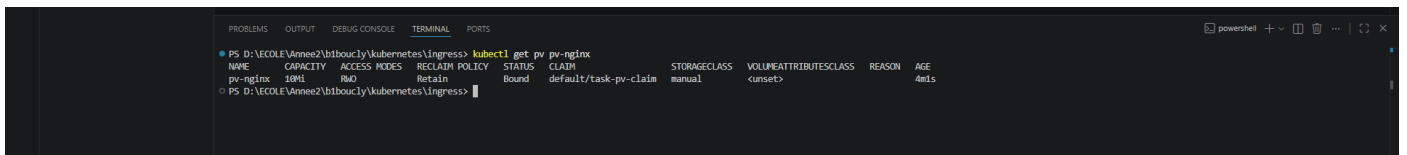
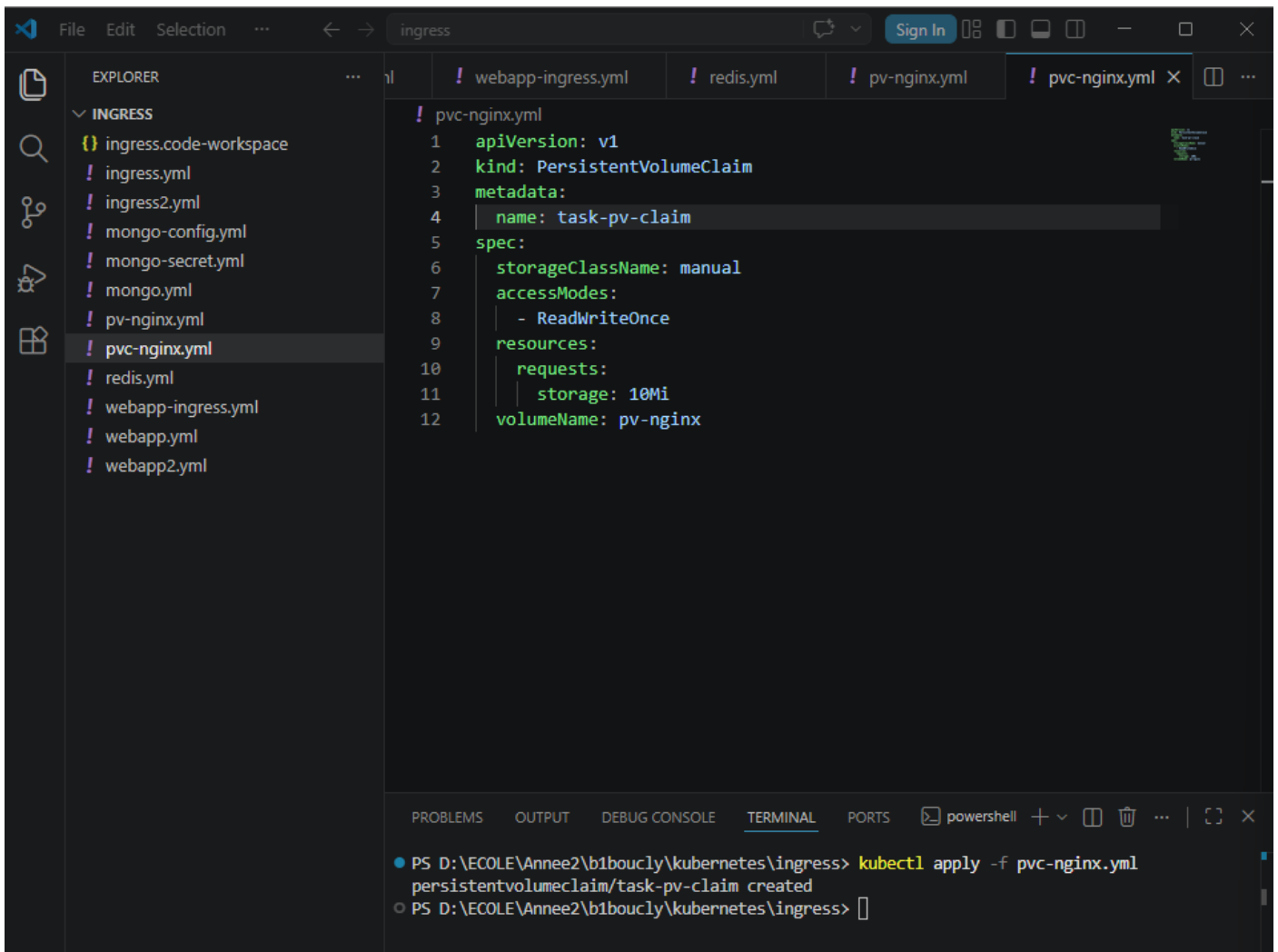
Below the code editor, the terminal window shows the following commands and output:

```
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl apply -f pv-nginx.yml
persistentvolume/pv-nginx created
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress>
```

The screenshot shows the terminal window with the following output:

```
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl get pv pv-nginx
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM  STORAGECLASS  VOLUMEATTRIBUTESCLASS  REASON  AGE
pv-nginx      10Mi     RWO          Retain          Available  Claim  manual        <unset>                 23s
```

➤ Je crée un PersistentVolumeClaim (demande de stockage) :



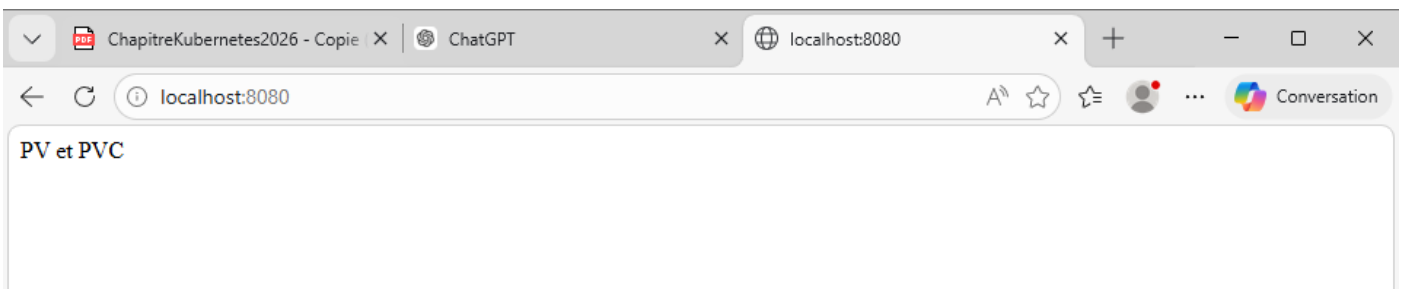
➤ Je décris le pod task-pv-pod :

```
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl describe pod task-pv-pod
Name: task-pv-pod
Namespace: ingress
Node: minikube
Pod IP: 10.0.2.15
Initiated: 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
    ReadOnly: false
  kube-api-access-svtgg:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigName: 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 100s default-scheduler Successfully assigned default/task-pv-pod to minikube
Normal Pulling 99s kubelet Pulling image "nginx:latest"
Normal Pulled 96s kubelet Successfully pulled image "nginx:latest" in 3.923s (3.923s including waiting). Image size: 160939956 bytes.
Normal Created 95s kubelet Container created
Normal Started 95s kubelet Container started
```

```
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl exec -it task-pv-pod -- /bin/bash
root@task-pv-pod:/#
```

```
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl port-forward task-pv-pod 80
Unable to listen on port 80: Listeners failed to create with the following errors: [unable to create listener: Error listen tcp4 127.0.0.1:80: bind: An attempt was made to access a socket in a way forbidden by its access permissions. unable to create listener: Error listen tcp6 [::]:80: bind: An attempt was made to access a socket in a way forbidden by its access permissions.]
error: unable to listen on any of the requested ports: [{80 80}]
PS D:\ECOLE\Annee2\b1boucly\kubernetes\ingress> kubectl port-forward task-pv-pod 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::]:8080 -> 80
```

➤ Je me connecte en localhost :



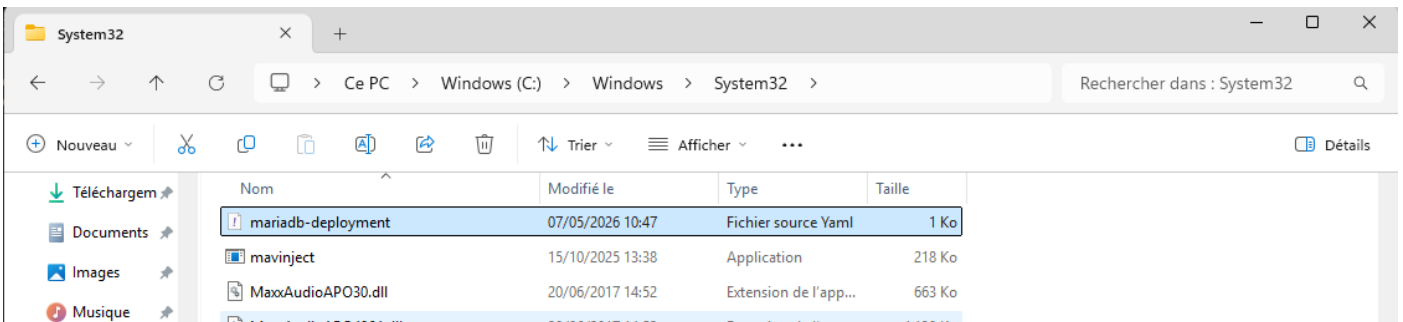
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
root@task-pv-pod:/# apt-get update
Hit:1 http://deb.debian.org/debian trixie InRelease
Hit:2 http://deb.debian.org/debian trixie-updates InRelease
Hit:3 http://deb.debian.org/debian-security trixie-security InRelease
Reading package lists... Done
root@task-pv-pod:/# curl http://localhost:80
PV et PVC
root@task-pv-pod:/#
```

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

- Je déploie une image mariadb :latest :

```
Sélection Administrateur : Windows PowerShell
PS C:\WINDOWS\system32> kubectl create deployment mariadb --image=mariadb:latest --dry-run=client --output yml > mariadb-deployment.yml
PS C:\WINDOWS\system32>
```

- Je me rends dans le system32 j'y retrouve mon fichier mariadb-deployment :



- J'ouvre le fichier avec VSCode, et je supprime les champs inutiles :

```
File Edit Selection ... ingress Sign In
gress.yml ! redis.yml ! pv-nginx.yml ! pvc-nginx.yml ! pvc-pod.yml ! mariadb-deployment.yml x ! ...
C: > Windows > System32 > ! mariadb-deployment.yml
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
```

```
File Edit Selection ... ingress Sign In
gress.yml ! redis.yml ! pv-nginx.yml ! pvc-nginx.yml ! pvc-pod.yml ! mariadb-deployment.yml x ! ...
C: > Windows > System32 > ! mariadb-deployment.yml
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
```

- Afin de conserver les données de la base, je crée un objet PVC :

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing 'mariadb-deployment.yaml' and 'pvc.yaml'. The main editor displays the 'pvc.yaml' file with the following content:

```
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
```

The 'name: mariadb-data' line is highlighted with a red box. Below the editor, the terminal shows the command `kubectl apply -f pvc.yaml` and its output: `persistentvolumeclaim/mariadb-data created`.

- Voici l'état de la demande de volume persistant :

The terminal window shows the command `kubectl get pvc mariadb-data` and its output:

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	VOLUMEATTRIBUTESCLASS	AGE
mariadb-data	Bound	pvc-78d1535b-c786-4054-9d77-7bbe799f18f2	100Mi	RWO	Standard	<unset>	31s

- Dans le fichier deployment je précise au conteneur de MariaDB d'utiliser le volume persistant :

```
1 kind: Deployment
2 metadata:
3   labels:
4     app: mariadb
5   name: mariadb
6 spec:
7   selector:
8     matchLabels:
9       app: mariadb
10  template:
11    metadata:
12      labels:
13        app: mariadb
14    spec:
15      volumes:
16        - name: mariadb-data
17          PersistentVolumeClaim:
18            claimName: mariadb-data
19      containers:
20        - image: mariadb:latest
21          name: mariadb
22          volumeMounts:
23            - mountPath: /var/lib/mysql
24              name: mariadb-data
25
26
```

➤ J'applique le fichier mariadb-deployment.yaml

```
PS C:\Users\mmalaussena\Desktop\pvc> kubectl apply -f mariadb-deployment.yaml
deployment.apps/mariadb created
PS C:\Users\mmalaussena\Desktop\pvc>
```

Lors de la première inscription le conteneur de met en statut « ContainerCreating », que je n'ai pas réussi à prendre en screen, ensuite il se mets en error :

```
PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pod -l app=mariadb
NAME                                READY   STATUS    RESTARTS   AGE
mariadb-5c476b5c79-t2c7t            0/1    Error     3 (52s ago)  71s
PS C:\Users\mmalaussena\Desktop\pvc>
```

➤ J'observe les logs et je me rends compte qu'il y a une erreur :

- Je me connecte au conteneur afin de tester le fonctionnement de la base de données :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mmalaussena\Desktop\pvc> kubectl exec -it deployment/mariadb -- bash
root@mariadb-5bb9c9b6d-sd64g:/# mariadb-admin status -p$MARIADB_ROOT_PASSWORD
Uptime: 114 Threads: 1 Questions: 1 Slow queries: 0 Opens: 17 Open tables: 10 Queries per second avg: 0.008
root@mariadb-5bb9c9b6d-sd64g:/# exit
exit
PS C:\Users\mmalaussena\Desktop\pvc>
```

- Je crée le fichier service.yaml :

```
File Edit Selection ... pvc Sign In
EXPLORER mariadb-deployment.yaml mariadb-deployment.yaml service.yaml
PVC mariadb-deployment.yaml mariadb-deployment.yaml service.yaml
! mariadb-deployment.yaml ! mariadb-deployment.yaml ! service.yaml
! 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
```

- Puis je l'applique :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + ...
PS C:\Users\mmalaussena\Desktop\pvc> kubectl apply -f service.yaml
service/mariadb created
PS C:\Users\mmalaussena\Desktop\pvc>
```

- Puis je mets en place la surveillance de la base de données :

➤ Je décris le pod mariadb :

```

kubect1 describe pods mariadb-6574478f7f-49q51
Name:          mariadb-6574478f7f-49q51
Namespace:    default
Priority:      0
Service Account: default
Node:         minikube/192.168.76.42
Start Time:   Thu, 07 May 2026 11:15:54 +0200
Labels:       app=mariadb
              pod-template-hash=6574478f7f
Annotations:  <none>
Status:       Running
IP:           10.244.0.28
IPs:          <none>
IP Address(es): 10.244.0.28
Controlled By: ReplicaSet/mariadb-6574478f7f
Containers:
  mariadb:
    Container ID:  docker://a36790adb4173f893436b58556789f2f34a8238f45cbc7222a9ccce78839773c
    Image:          mariadb:latest
    Image ID:      docker-pullable://mariadb@sha256:e8236fc6386e7eacd9359e59d8a878bd7aa8d18288d36d13861121bedeae983
    Port:          <none>
    Host Port:     <none>
    State:         Running
      Started:     Thu, 07 May 2026 11:16:59 +0200
    Last State:    Terminated
      Reason:      Error
      Exit Code:   1
      Started:     Thu, 07 May 2026 11:16:27 +0200
      Finished:    Thu, 07 May 2026 11:16:58 +0200
    Ready:        False
    Restart Count: 2
    Liveness:      exec [sh -c mariadb-admin status -pMARIADB_ROOT_PASSWORD] delay=0s timeout=1s period=10s #success=1 #failure=3
    Readiness:     exec [sh -c mariadb-admin status -pMARIADB_ROOT_PASSWORD] delay=0s timeout=1s period=10s #success=1 #failure=3
    Startup:       exec [sh -c mariadb-admin status -pMARIADB_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-g8n6h (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-g8n6h:
      Type:            Projected (a volume that contains injected data from multiple sources)
      TokenExpirationSeconds: 3607
      ConfigMapName:    kube-root-ca.crt
      Optional:         false
      DownwardAPI:     true
    OS 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   88s   default-scheduler  Successfully assigned default/mariadb-6574478f7f-49q51 to minikube
  Normal  Pulled      79s   kubelet            Successfully pulled image "mariadb:latest" in 931ms (931ms including waiting). Image size: 335568219 bytes.
  Normal  Pulled      47s   kubelet            Successfully pulled image "mariadb:latest" in 821ms (821ms including waiting). Image size: 335568219 bytes.
  Normal  Killing     28s (x2 over 58s)  kubelet            Container mariadb failed startup probe, will be restarted
  Normal  Pulling     16s (x3 over 79s)  kubelet            Pulling image "mariadb:latest"
  Normal  Created     15s (x3 over 79s)  kubelet            Container created
  Normal  Started     15s (x3 over 78s)  kubelet            Container started
  Normal  Pulled      15s   kubelet            Successfully pulled image "mariadb:latest" in 878ms (878ms including waiting). Image size: 335568219 bytes.
  Warning  Failed      20s (x2 over 20s)  kubelet            Failed to start container: Error: exec [sh -c mariadb-admin status -pMARIADB_ROOT_PASSWORD] failed: exit status 1: 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!
  
```

➤ Je regarde les pods et je me rends compte que l'un d'eux est en CrashLoopBackOff :

```

kubect1 get pod -l app=mariadb
NAME                                READY   STATUS             RESTARTS   AGE
mariadb-5bbb0c9b6d-sd64g            1/1     Running            0          0m28s
mariadb-6574478f7f-49q51            0/1     CrashLoopBackOff   4 (23s ago) 3m2s
  
```

➤ Je regarde les logs, je constate la présence d'une erreur :

```
PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pod -l app=mariadb
NAME                                READY   STATUS    RESTARTS   AGE
mariadb-5bb9c9bd-sd64g              1/1     Running   0           9m55s
mariadb-6574478f7f-49q5l            0/1     CrashLoopBackOff   4 (58s ago)   3m29s

PS C:\Users\mmalaussena\Desktop\pvc> kubectl logs mariadb-6574478f7f-49q5l
2026-05-07 09:19:25+08:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2-maria-ubu2404 started.
2026-05-07 09:19:25+08:00 [Warn] [Entrypoint]: /sys/fs/cgroup/memory.pressure not writable, functionality unavailable to MariaDB
2026-05-07 09:19:25+08:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2026-05-07 09:19:25+08:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2-maria-ubu2404 started.
2026-05-07 09:19:25+08:00 [Note] [Entrypoint]: MariaDB upgrade not required
2026-05-07 09:19:25 0 [Note] Starting MariaDB 12.2.2-MariaDB-ubu2404 source revision d26a6f4c1f2119277e79b9548886c6d8c01472f server_uid Fc9P8lAYe2YE189525CrJl0yZrE= as process 1
2026-05-07 09:19:25 0 [ERROR] mariadb: Can't lock aria control file '/var/lib/mysql/aria_log_control' for exclusive use, error: 11. Will retry for 30 seconds
PS C:\Users\mmalaussena\Desktop\pvc>
```

➤ Le problème vient du fait qu'une base de données ne peut pas partager ses fichiers :

```
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:
```

Le RollingUpdate correspond à un remplacement par vague des pods existants (mécanisme par défaut)

Le Recreate correspond à une suppression des anciens pods avant création (no rolling update)

```
PS C:\Users\mmalaussena\Desktop\pvc> kubectl apply -f mariadb-deployment.yaml
deployment.apps/mariadb configured
PS C:\Users\mmalaussena\Desktop\pvc>
```

```
PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pod -l app=mariadb --watch
h
NAME                                READY   STATUS    RESTARTS   AGE
mariadb-6574478f7f-49q5l            0/1     CrashLoopBackOff   6 (79s ago)   5m52s
PS C:\Users\mmalaussena\Desktop\pvc>
```

- Dans le cas si dessus ou le pod reste dans l'état CrashLoopBackOff, il faut supprimer le pod :

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pod -l app=mariadb
NAME READY STATUS RESTARTS AGE
mariadb-6574478f7f-49q5l 0/1 CrashLoopBackOff 6 (2m8s ago) 6m41s
PS C:\Users\mmalaussena\Desktop\pvc> kubectl delete pods -l app=mariadb
pod "mariadb-6574478f7f-49q5l" deleted from default namespace
PS C:\Users\mmalaussena\Desktop\pvc>

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pod -l app=mariadb
NAME READY STATUS RESTARTS AGE
mariadb-6574478f7f-t6rg6 1/1 Running 0 40s
PS C:\Users\mmalaussena\Desktop\pvc>

```

- Dans le Powershell j'ai augmenté le nombre de pods associés au déploiement et la montée en charge automatique :

```

Administrateur : Windows PowerShell
PS C:\WINDOWS\system32> kubectl scale deployment mariadb --replicas=2
deployment.apps/mariadb scaled
PS C:\WINDOWS\system32> kubectl get pods -l app=mariadb --watch
NAME READY STATUS RESTARTS AGE
mariadb-6574478f7f-fl2xp 0/1 Running 0 11s
mariadb-6574478f7f-t6rg6 1/1 Running 0 108s

```

- Mais le nouveau pod n'arrive pas à démarrer :

```

Administrateur : Windows PowerShell
PS C:\WINDOWS\system32> kubectl describe pods mariadb-6574478f7f-fl2xp
Name: mariadb-6574478f7f-fl2xp
Namespace: default
Priority: 0
Service Account: default
Node: minikube/192.168.76.42
Start Time: Thu, 07 May 2026 11:24:17 +0200
Labels: app=mariadb
pod-template-hash=6574478f7f
Annotations: <none>
Status: Running
IP: 10.244.0.30
IPs:
IP: 10.244.0.30
Controlled By: ReplicaSet/mariadb-6574478f7f
Containers:
  mariadb:
    Container ID: docker://14c6a9d2c-fb5f40c643e75f7a2ce9d191216dadd7384d88199c46f6e73527f0e
    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 11:26:25 +0200
      Finished: Thu, 07 May 2026 11:26:56 +0200
    Ready: False
    Restart Count: 4
    Liveness: exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s timeout=1s period=10s #succes
s=1 #failure=3
    Readiness: exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s timeout=1s period=10s #succes
s=1 #failure=3
    Startup: exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s timeout=1s period=10s #succes
s=1 #failure=3

```

```
node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type            Reason            Age             From              Message
  ----            -
  Normal          Scheduled         3m26s          default-scheduler Successfully assigned default/mariadb-6574478f7f-fl2xp
  to minikube
  Normal          Pulled            3m25s          kubelet           Successfully pulled image "mariadb:latest" in 891ms (8
  91ms including waiting). Image size: 335568219 bytes.
  Normal          Pulled            2m53s          kubelet           Successfully pulled image "mariadb:latest" in 889ms (8
  89ms including waiting). Image size: 335568219 bytes.
  Normal          Pulled            2m21s          kubelet           Successfully pulled image "mariadb:latest" in 872ms (8
  72ms including waiting). Image size: 335568219 bytes.
  Normal          Pulled            110s          kubelet           Successfully pulled image "mariadb:latest" in 906ms (9
  06ms including waiting). Image size: 335568219 bytes.
  Normal          Pulling           79s (x5 over 3m26s) kubelet           Pulling image "mariadb:latest"
  Normal          Created           78s (x5 over 3m25s) kubelet           Container created
  Normal          Started           78s (x5 over 3m25s) kubelet           Container started
  Normal          Pulled            78s           kubelet           Successfully pulled image "mariadb:latest" in 885ms (8
  85ms including waiting). Image size: 335568219 bytes.
  Warning          Unhealthy         56s (x15 over 3m16s) 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           56s (x5 over 2m56s) kubelet           Container mariadb failed startup probe, will be restarted
  Warning          BackOff           46s (x3 over 47s)   kubelet           Back-off restarting failed container mariadb in pod mariadb-6574478
  f7f-fl2xp_default(178ddca9-3ced-4d90-b7f1-ffe329687dbf)
PS C:\WINDOWS\system32>
```

Le problème est tout à fait normal puisqu'une base de données s'attend à avoir un accès exclusif à ses fichiers.

Le mécanisme de déploiement n'est pas adapté pour gérer plusieurs pods de base de données. Kubernetes propose pour cela un autre type d'objet : le type StatefulSet

- J'évince l'ancien déploiement de MariaDB :

```
Administrateur : Windows PowerShell
PS C:\WINDOWS\system32> kubectl delete deployment mariadb
deployment.apps "mariadb" deleted from default namespace
PS C:\WINDOWS\system32> kubectl get pods -l app=mariadb
No resources found in default namespace.
PS C:\WINDOWS\system32>
```

- Je modifie le mariadb-deployment.yaml et sauvegarde sous le nom mariadb statefulset.yaml :

```

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

```

➤ Je crée le StatefulSet et je consulte l'état des pods associés au StatefulSet de MariaDB :

```

PS C:\Users\mmalaussena\Desktop\pvc> kubectl apply -f mariadb-statefulset.yaml
statefulset.apps/mariadb created
PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pods -l app=mariadb --watch
NAME          READY   STATUS    RESTARTS   AGE
mariadb-0     1/1     Running   0           45s

```

➤ Je consulte l'état des demandes de volumes persistants :

```

PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pvc
NAME          STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mariadb-data  Bound   pvc-7d41535b-c786-4054-9a77-7bbe799f18f2  100Mi      RWO            standard      <unset>                 47m
mariadb-data-mariadb-0  Bound   pvc-3c777164-1c00-47ff-a740-9dd1b2d7838b  100Mi      RWO            standard      <unset>                 68s
mysql-pv-claim  Bound   pvc-ffb539f6-3fbf-4b5b-9b8a-739f312b26ed  20Gi       RWO            standard      <unset>                 21h
task-pv-claim  Bound   pv-nglnrx                                 10Mi       RWO            manual        <unset>                 88s
wp-pv-claim    Bound   pvc-32873449-3676-4707-813e-aa7e0ad7bc17  20Gi       RWO            standard      <unset>                 21h

```

➤ Je consulte l'état des volumes persistants :

```

PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pv
NAME          CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
pv-nglnrx    10Mi       RWO            Retain           Bound   default/task-pv-claim  manual        <unset>                 84m
pvc-32873449-3676-4707-813e-aa7e0ad7bc17  20Gi       RWO            Delete          Bound   default/wp-pv-claim    standard      <unset>                 21h
pvc-3c777164-1c00-47ff-a740-9dd1b2d7838b  100Mi      RWO            Delete          Bound   default/mariadb-data-mariadb-0  standard      <unset>                 89s
pvc-7d41535b-c786-4054-9a77-7bbe799f18f2  100Mi      RWO            Delete          Bound   default/mariadb-data      standard      <unset>                 48m
pvc-ffb539f6-3fbf-4b5b-9b8a-739f312b26ed  20Gi       RWO            Delete          Bound   default/mysql-pv-claim  standard      <unset>                 21h

```

- Je supprime l'ancien objet mariadb-data de type PVC :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mmalaussena\Desktop\pvc> kubectl delete pvc/mariadb-data
persistentvolumeclaim "mariadb-data" deleted from default namespace
PS C:\Users\mmalaussena\Desktop\pvc> 
```

- Je mets en place la scalabilité de l'objet StatefulSet :

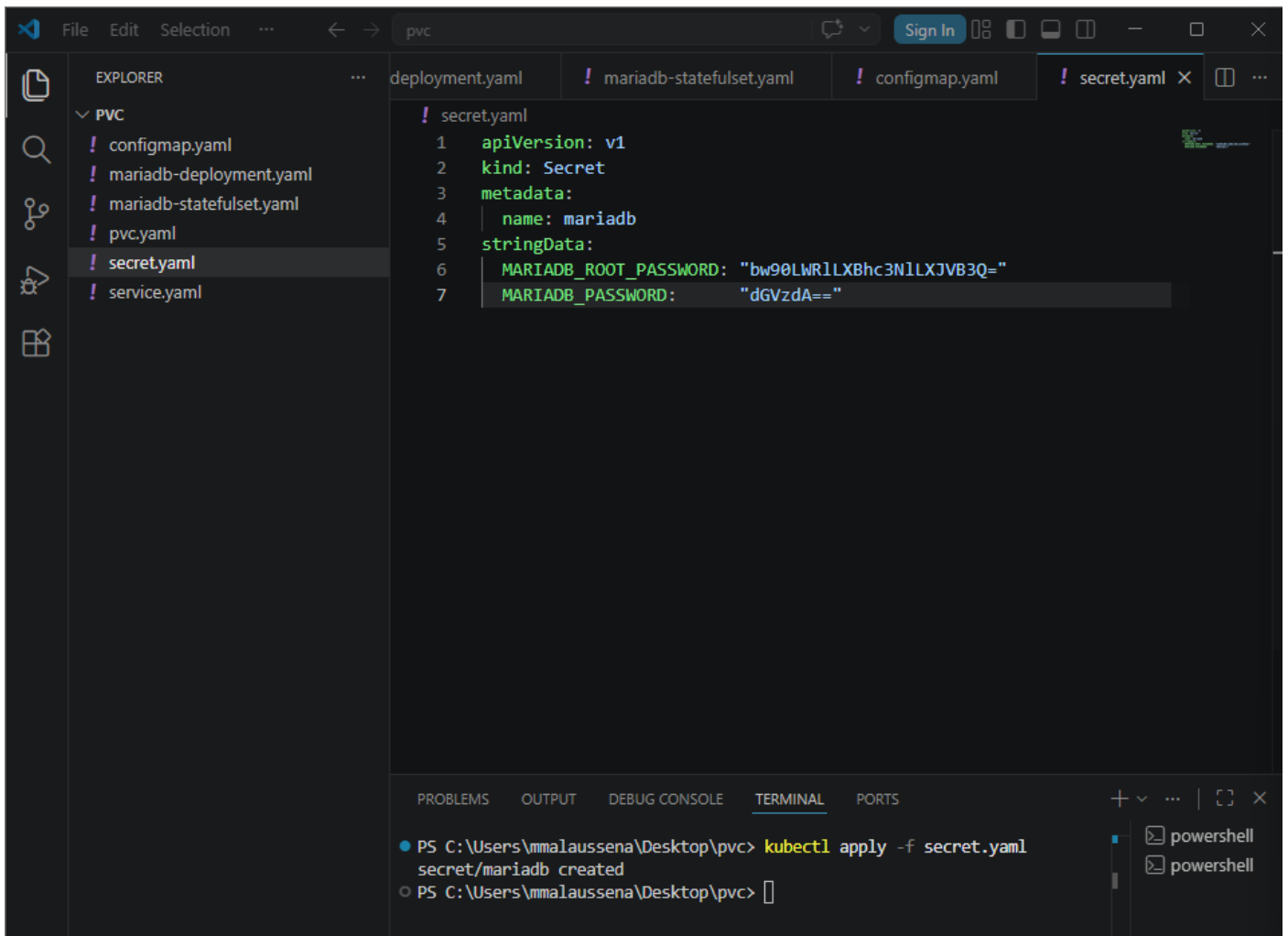
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mmalaussena\Desktop\pvc> kubectl scale sts mariadb --replicas=
2
statefulset.apps/mariadb scaled
PS C:\Users\mmalaussena\Desktop\pvc> 
```

- Je consulte les demandes de volumes persistants et des pods :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pvc -l app=mariadb
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS VOLUMEATTRIBUTESCLASS AGE
mariadb-data-mariadb-0 Bound pvc-3c777164-1c80-47ff-a740-9dd1b2d7838b 100Mi RWO standard <unset> 3m16s
mariadb-data-mariadb-1 Bound pvc-33f1ab45-ba04-4110-9b4f-bb629260a792 100Mi RWO standard <unset> 23s
PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pods -l app=mariadb
NAME READY STATUS RESTARTS AGE
mariadb-0 1/1 Running 0 3m27s
mariadb-1 1/1 Running 0 34s
PS C:\Users\mmalaussena\Desktop\pvc> kubectl get pods,pvc -l app=mariadb
NAME READY STATUS RESTARTS AGE
pod/mariadb-0 1/1 Running 0 3m34s
pod/mariadb-1 1/1 Running 0 41s
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS VOLUMEATTRIBUTESCLASS AGE
persistentvolumeclaim/mariadb-data-mariadb-0 Bound pvc-3c777164-1c80-47ff-a740-9dd1b2d7838b 100Mi RWO standard <unset> 3m34s
persistentvolumeclaim/mariadb-data-mariadb-1 Bound pvc-33f1ab45-ba04-4110-9b4f-bb629260a792 100Mi RWO standard <unset> 41s
PS C:\Users\mmalaussena\Desktop\pvc> 
```

- Je crée les fichiers configmap.yaml, secret.yaml et mariadb-statefulset2.yaml et j'applique les fichiers :

```
File Edit Selection ... mariadb-deployment.yaml mariadb-statefulset.yaml configmap.yaml
EXPLORER
pvc
! configmap.yaml
! mariadb-deployment.yaml
! mariadb-statefulset.yaml
! pvc.yaml
! service.yaml
! 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\mmalaussena\Desktop\pvc> kubectl apply -f configmap.yaml
configmap/mariadb created
PS C:\Users\mmalaussena\Desktop\pvc> 
```



```
File Edit Selection ... pvc Sign In
EXPLORER
PVC
! configmap.yaml
! mariadb-deployment.yaml
! mariadb-statefulset.yaml
! mariadb-statefulset2.yaml
! pvc.yaml
! secret.yaml
! service.yaml
! mariadb-deployment.yaml
! mariadb-statefulset.yaml
! mariadb-statefulset2.yaml
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:
```

```
File Edit Selection ... pvc Sign In
EXPLORER
PVC
! configmap.yaml
! mariadb-deployment.yaml
! mariadb-statefulset.yaml
! mariadb-statefulset2.yaml
! pvc.yaml
! secret.yaml
! service.yaml
! mariadb-statefulset2.yaml
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
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mmalaussena\Desktop\pvc> kubectl apply -f mariadb-statefulset2
.yaml
statefulset.apps/mariadb configured
PS C:\Users\mmalaussena\Desktop\pvc> 
```