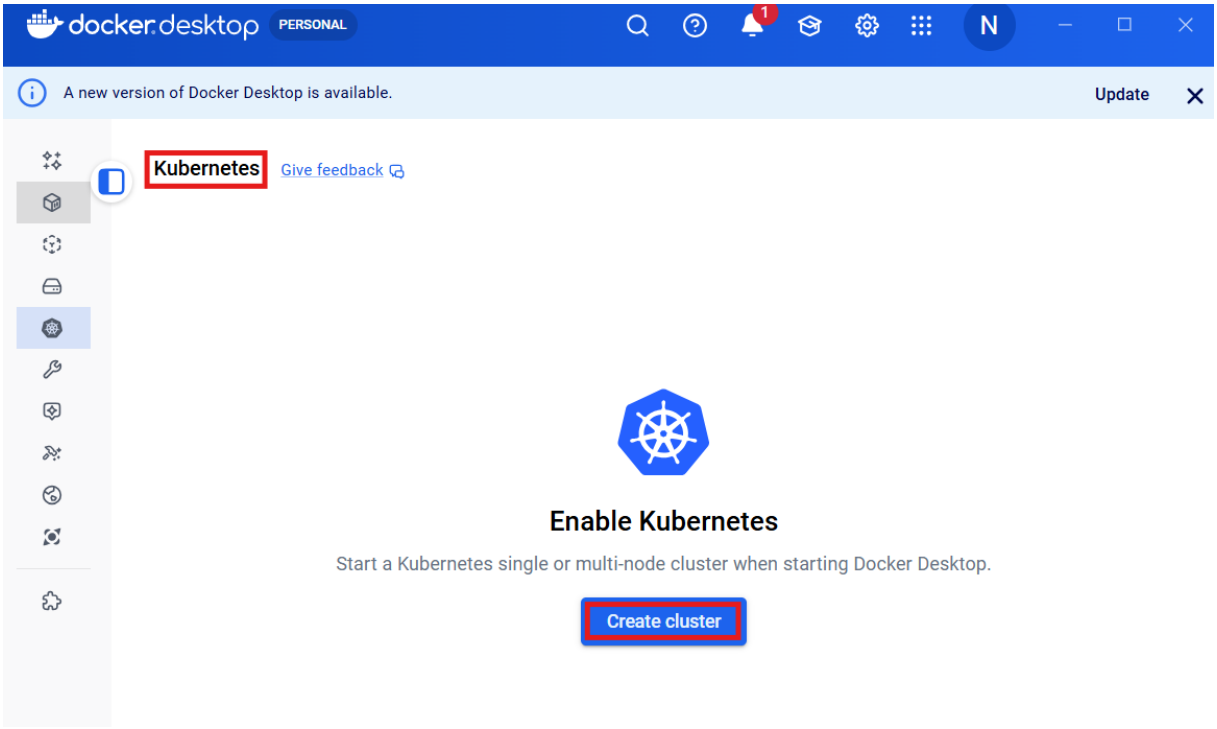
Chapitre 5 – Kubernetes : plateforme de déploiement des applications conteneurisées

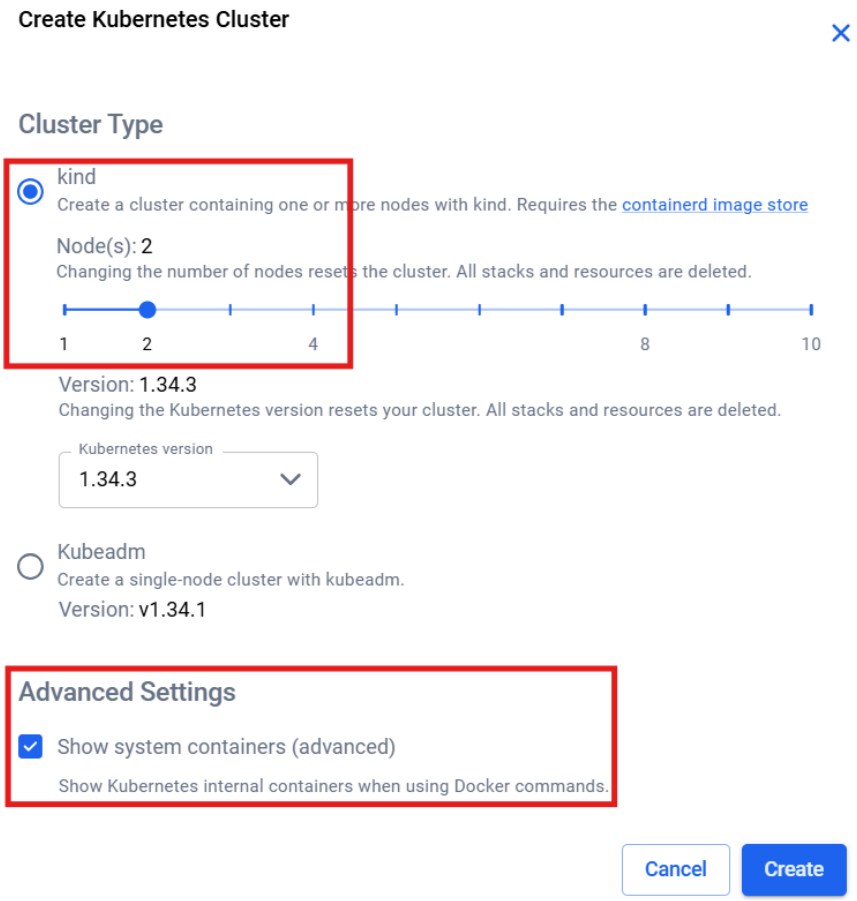
Table des matières

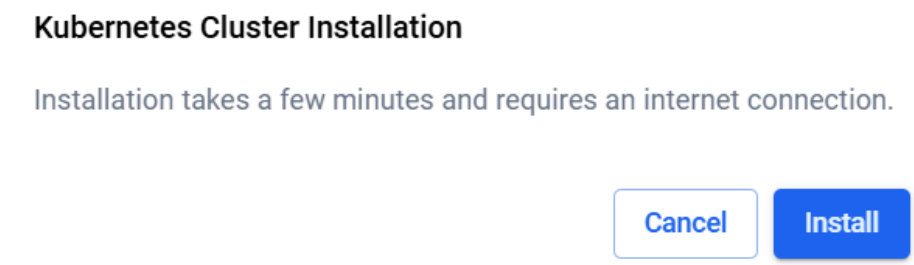
1. Kubernetes avec Docker Desktop.

Nous lançons Kubernetes depuis Docker.

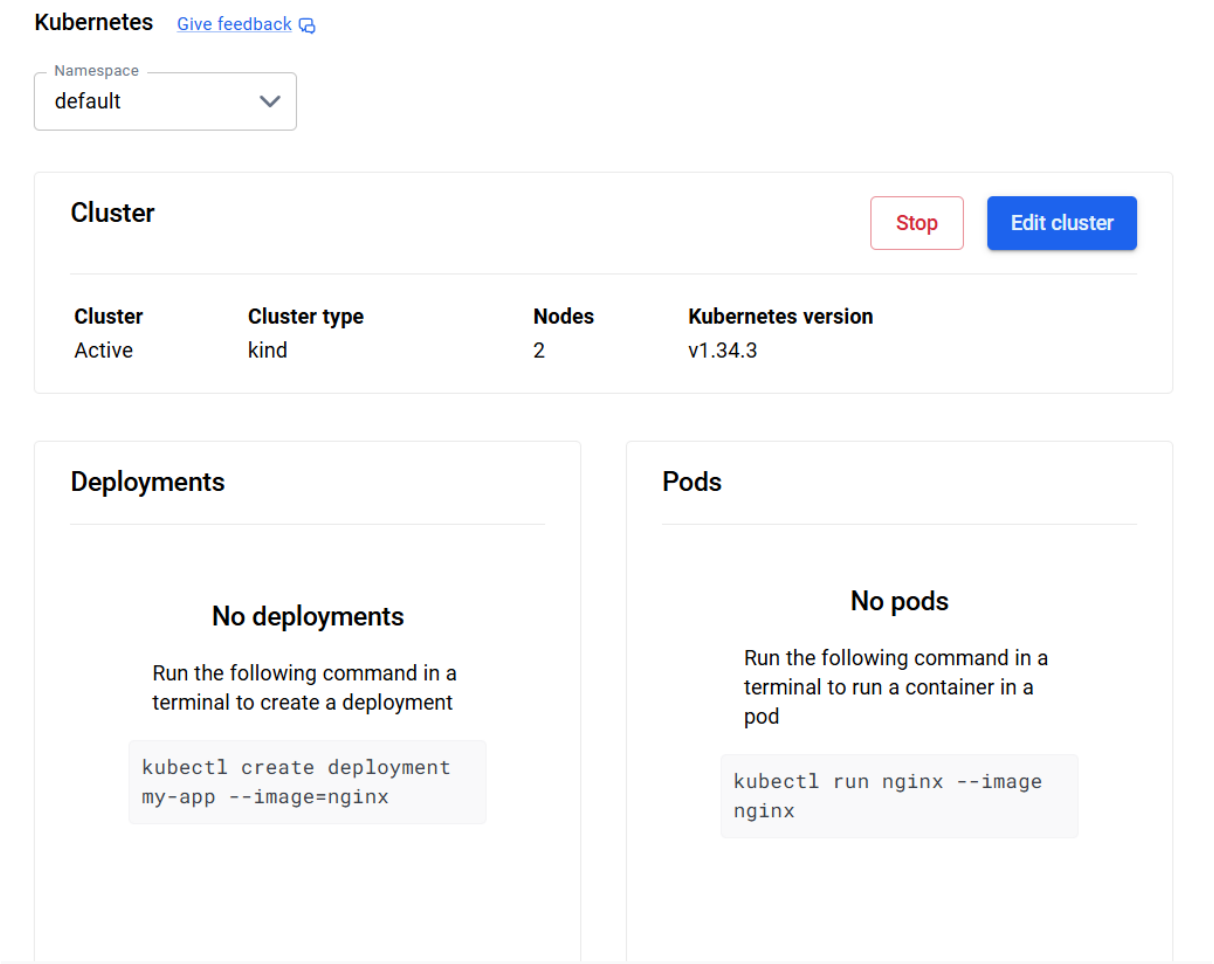


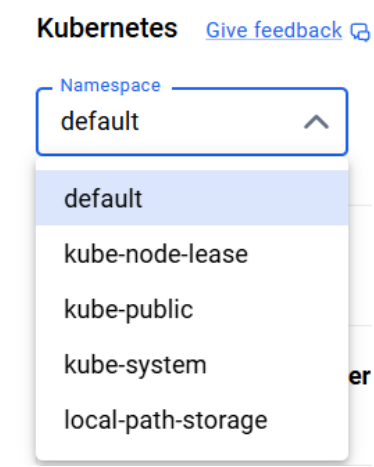
Nous créons un cluster, nous sélectionnons kind avec deux nœuds.



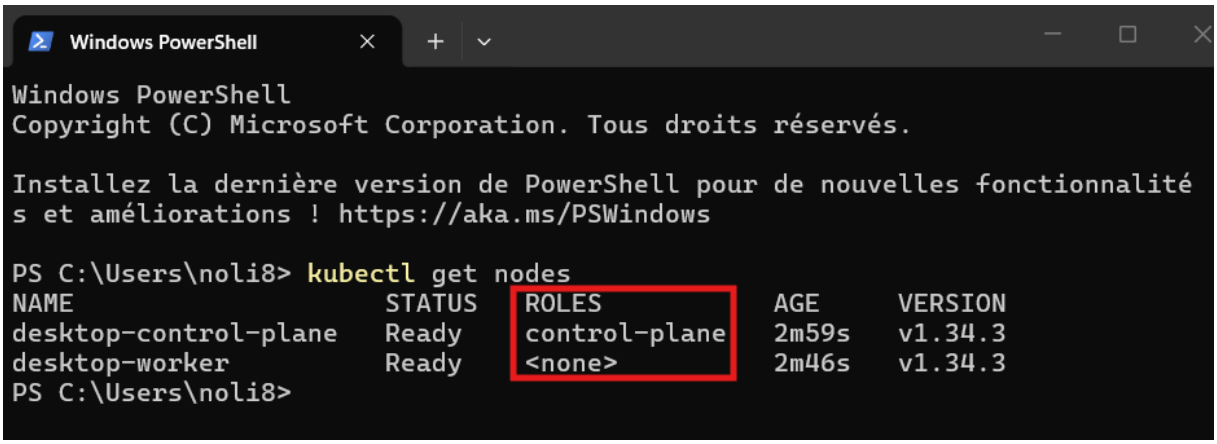


Nous pouvons constater que notre cluster a bien été créé et que notre nœud ne possède aucun pods.

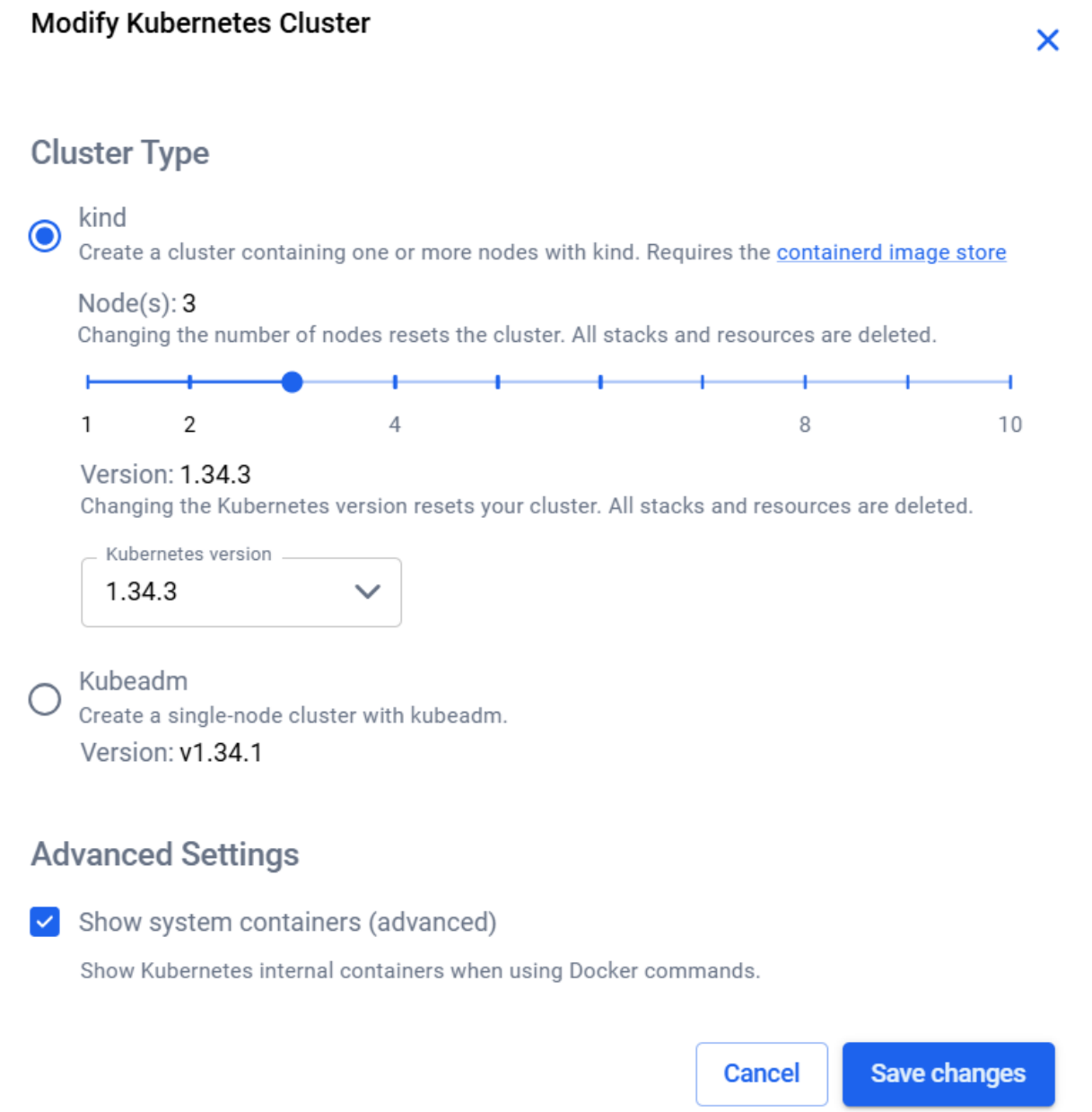


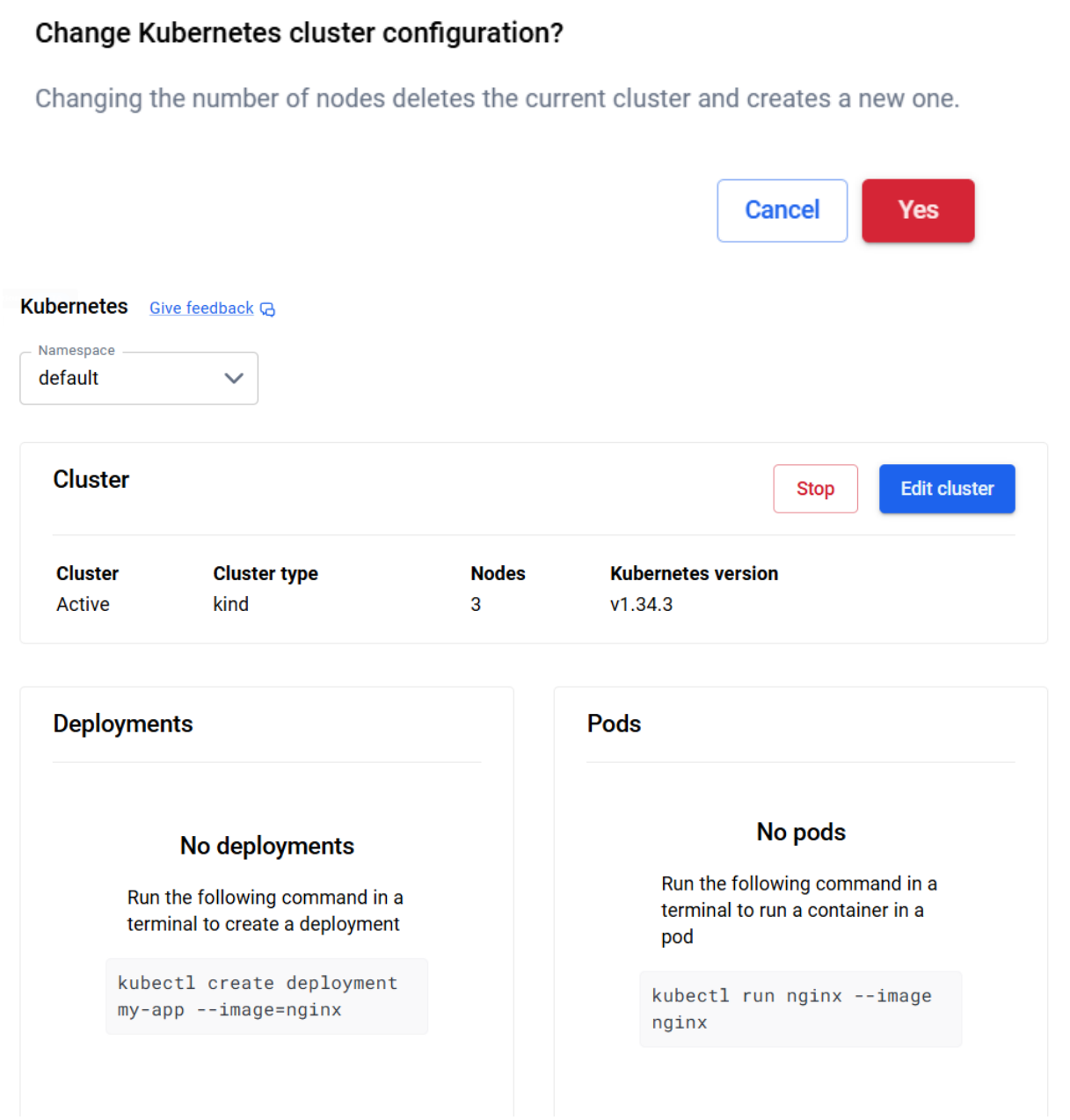


Cette commande nous permet de remarquer qu'un des deux nœuds est maître et l'autre un worker.



Nous y ajoutons un troisième nœud.

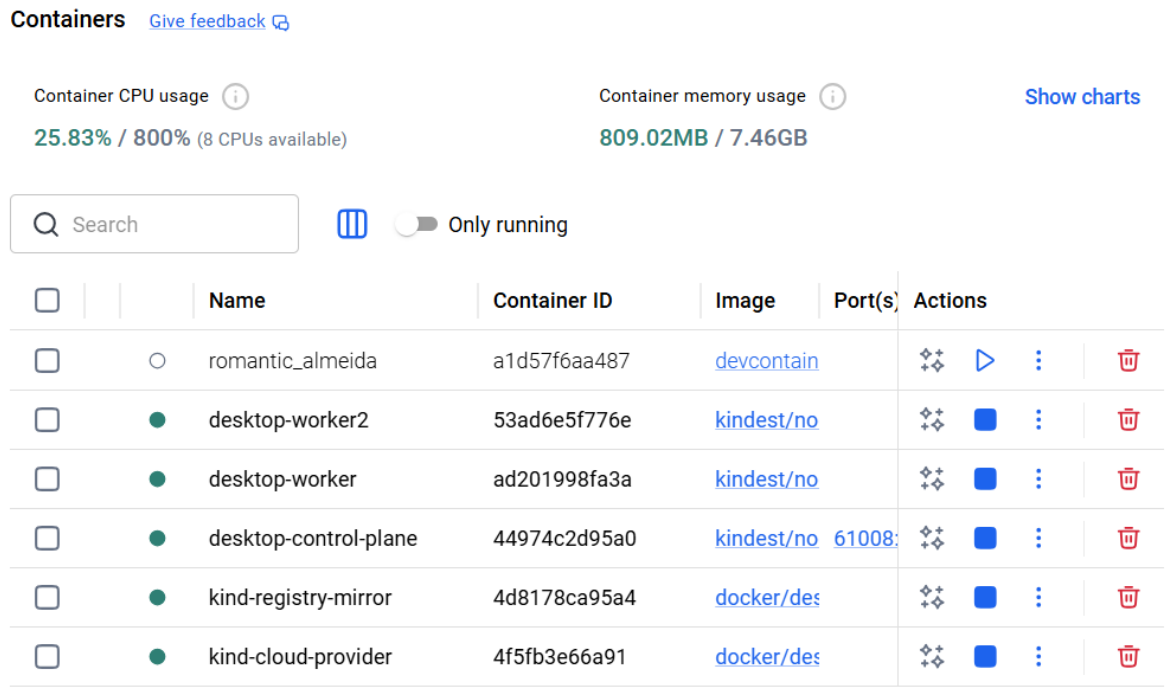




Nous constatons qu'il n'y a toujours qu'un seul maître.

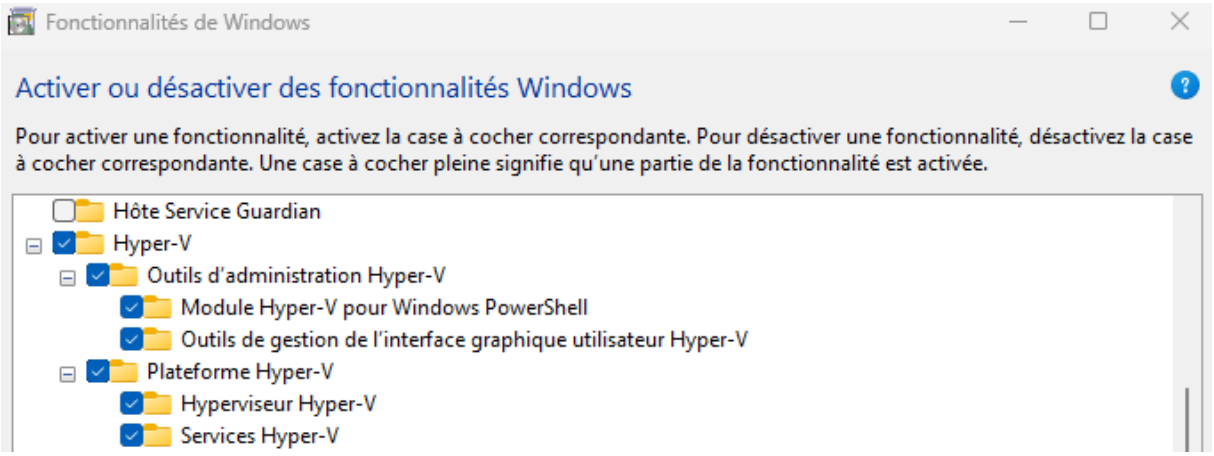
```
PS C:\Users\noli8> kubectl get nodes
NAME                   STATUS   ROLES           AGE    VERSION
desktop-control-plane   Ready    control-plane   107s   v1.34.3
desktop-worker          Ready    <none>          94s    v1.34.3
desktop-worker2         Ready    <none>          93s    v1.34.3
PS C:\Users\noli8>
```

Les nœuds apparaissent dans les containers.



2. Installation de Minikube.

Nous activons Hyper-V.



Une fois Hyper-V activé, nous installons Minikube.

```
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-----        29/04/2026     17:31                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>
```

```
PS C:\WINDOWS\system32> $oldPath = [Environment]::GetEnvironmentVariable('Path', [EnvironmentVariableTarget]::Machine)
PS C:\WINDOWS\system32> if ($oldPath.Split(';') -inotcontains 'C:\minikube'){
>>   [Environment]::SetEnvironmentVariable('Path', $('{0};C:\minikube' -f $oldPath), [EnvironmentVariableTarget]::Machine)
>> }
PS C:\WINDOWS\system32>
```

Nous lançons minikube qui est la machine.

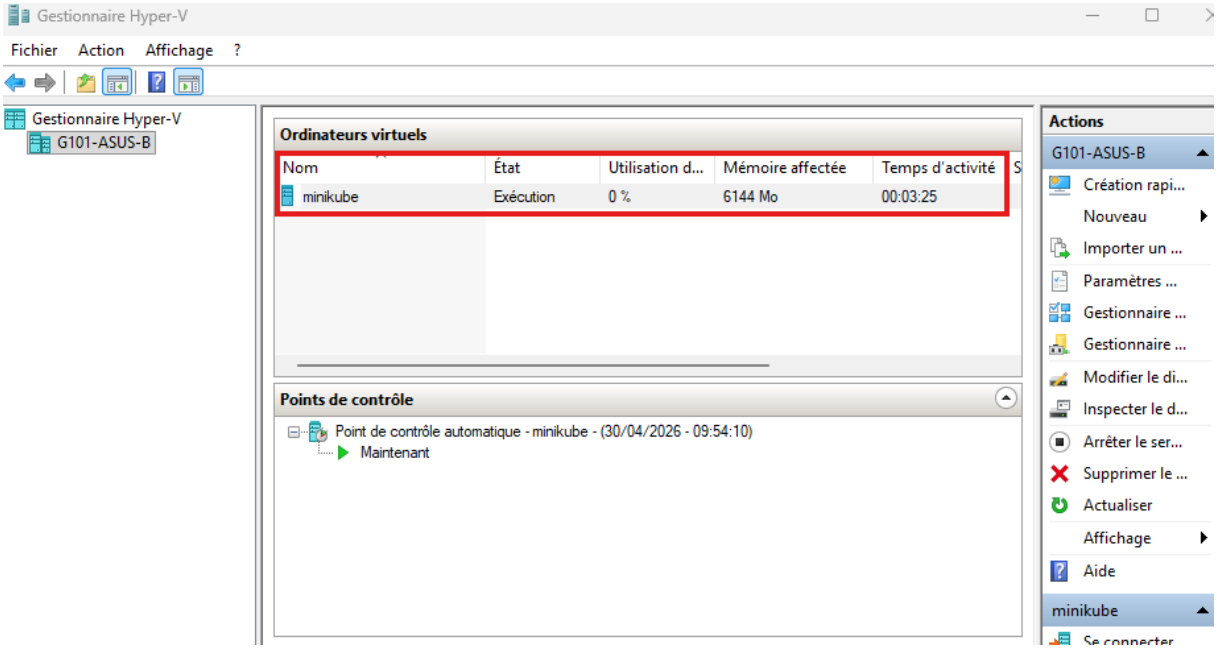
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

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

PS C:\WINDOWS\system32> minikube start
* minikube v1.38.1 sur Microsoft Windows 11 Pro 25H2
* Choix automatique du pilote virtualbox
! 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% 36.44 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% 34.99 M
* Création de VM virtualbox (CPUs=2, Mémoire=6144MB, Disque=20000MB)...
* Suppression de "minikube" dans virtualbox...
! StartHost a échoué, mais va réessayer : creating host: create: creating: C:\Program Files\Oracle\VirtualBox\VBoxManage.exe modifyvm minikube --firmware bios --bioslogofadein off --bioslogofadeout off --bioslogodisplaytime 0 --biosbootmenu disabled --ostype Linux26_64 --cpus 2 --memory 6144 --acpi on --ioapic on --rtcuseutc on --natdnshostresolver1 on --natdnsproxy1 off --cpuhotplug off --pae on --hpet on --hwvirtex on --nestedpaging on --largepages on --vtxvpid on --accelerate3d off --boot1 dvd --natlocalhostreachable1 on failed:
VBoxManage.exe: error: The graphics controller does not support the given feature
VBoxManage.exe: error: Details: code VBOX_E_NOT_SUPPORTED (0x80bb0009), component GraphicsAdapterWrap, interface IGraphicsAdapter, calleeIUnknown
VBoxManage.exe: error: Context: "SetFeature(GraphicsFeature_Acceleration3D, ValueUnion.f)" at line 1108 of file VBoxManageModifyVM.cpp

* Échec du démarrage de virtualbox VM. L'exécution de "minikube delete" peut résoudre le problème : error loading existing host. Please try running [minikube delete], then run [minikube start] again: filestore "minikube": open C:\Users\standard\.minikube\machines\minikube\config.json: The system cannot find the file specified.

X Fermeture en raison de GUEST_NOT_FOUND : Failed to start host: error loading existing host. Please try running [minikube delete], then run [minikube start] again: filestore "minikube": open C:\Users\standard\.minikube\machines\minikube\config.json: The system cannot find the file specified.
* Suggestion : minikube manque des fichiers relatifs à votre environnement invité. Cela peut être corrigé en exécutant 'minikube delete'
* Problème connexe: https://github.com/kubernetes/minikube/issues/9130

PS C:\WINDOWS\system32>
```

```
PS C:\Users\nnicolau> minikube start
😄  minikube v1.38.1 sur Microsoft Windows 11 Education 25H2
✨  Choix automatique du pilote hyperv. Autres choix: virtualbox, ssh
❗  Starting v1.39.0, minikube will default to "containerd" container runtime. See #21973 for more info.
💿  Téléchargement de l'image de démarrage de la VM...
    > minikube-v1.38.0-amd64.iso....:  65 B / 65 B [---------] 100.00% ? p/s 0s
    > minikube-v1.38.0-amd64.iso:  370.55 MiB / 370.55 MiB  100.00% 40.07 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% 36.93 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\nnicolau>
```

Cette commande nous permet d'afficher le nombre de nœuds.

```
PS C:\Users\nnicolau> kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   80s   v1.35.1
PS C:\Users\nnicolau>
```

Nous constatons que la machine a bien été créée dans Hyper-V.



La commande get all nous permet d'afficher la liste des objets.

```
PS C:\Users\nnicolau> kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   5m15s
```

Nous créons un pod nginx.

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

Nous vérifions que le pod a bien été créé.

```
PS C:\Users\nnicolau> kubectl get pods
NAME     READY   STATUS    RESTARTS   AGE
monpod   1/1     Running   0          64s
PS C:\Users\nnicolau>
```

La commande get all permet d'afficher tous les pods et les services.

```
PS C:\Users\nnicolau> kubectl get all
NAME         READY   STATUS    RESTARTS   AGE
pod/monpod   1/1     Running   0          2m8s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   13m
PS C:\Users\nnicolau>
```

Nous affichons les logs du pod monpod plus en détail.

```
PS C:\Users\nnicolau> kubectl logs pod/monpod
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2026/04/30 08:06:32 [notice] 1#1: using the "epoll" event method
2026/04/30 08:06:32 [notice] 1#1: nginx/1.29.8
2026/04/30 08:06:32 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2026/04/30 08:06:32 [notice] 1#1: OS: Linux 6.6.95
2026/04/30 08:06:32 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2026/04/30 08:06:32 [notice] 1#1: start worker processes
2026/04/30 08:06:32 [notice] 1#1: start worker process 29
2026/04/30 08:06:32 [notice] 1#1: start worker process 30
PS C:\Users\nnicolau>
```

Nous affichons les espaces de nom dans lesquels sont les objets.

```
PS C:\Users\nnicolau> kubectl get namespace
NAME              STATUS   AGE
default           Active   17m
kube-node-lease   Active   17m
kube-public       Active   17m
kube-system       Active   17m
PS C:\Users\nnicolau>
```

Nous constatons que les commandes pour afficher les pod et les node fonctionnent au pluriel ou au singulier.

```
PS C:\Users\nnicolau> kubectl get pods
NAME     READY   STATUS    RESTARTS   AGE
monpod   1/1     Running   0          7m38s
PS C:\Users\nnicolau> kubectl get pod
NAME     READY   STATUS    RESTARTS   AGE
monpod   1/1     Running   0          7m40s
PS C:\Users\nnicolau>
```

```
PS C:\Users\nnicolau> kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   19m   v1.35.1
PS C:\Users\nnicolau> kubectl get node
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   19m   v1.35.1
PS C:\Users\nnicolau>
```

Le paramètre -O permet d'afficher des champs supplémentaires.

```
PS C:\Users\nnicolau> kubectl get node -o wide
NAME       STATUS   ROLES           AGE   VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE           KERNEL-VERSION   CONTAINER-RUNTIME
minikube   Ready    control-plane   20m   v1.35.1   172.23.63.113   <none>        Buildroot 2025.02   6.6.95           docker://28.5.2
PS C:\Users\nnicolau>
```

Nous pouvons entrer dans la machine minikube en ssh et vérifier les containers qui y tournent avec les services.

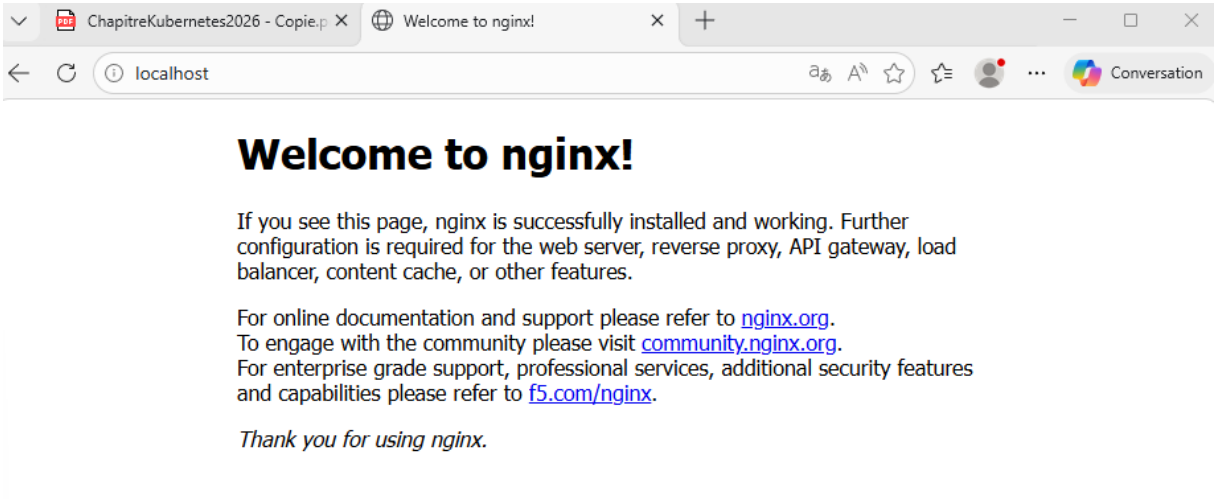
```
PS C:\Users\nnicolau> minikube ssh
                       _             _
            _         _ ( )           ( )
  ___ ___  (_)  ___  (_)| |/')  _   _ | |_      __
/' _ ` _ `\| |/' _ `\| || , <  ( ) ( )| '_`\  /'__`\
| ( ) ( ) || || ( ) || || |\`\ | (_) || |_) )(  ___/
(_) (_) (_)(_)(_) (_)(_)(_)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

$ sudo ctr namespace ls
NAME LABELS
moby
$ sudo ctr -n moby container ls
CONTAINER                                                           IMAGE    RUNTIME
04d1358f0b90780e2ce85f73aa0ac9a6e328bbe9c7123c4be044cf028fb9ec6e    -        io.containerd.runc.v2
0d7ee30800ac68fccf1aec7cf8c91c4ab6bc50d66c1eaa772a86c77136532c87    -        io.containerd.runc.v2
2108b1a08caf3d2e8b803372cd1d5683d8d2e26f311e4ebce1fb054f15ee2b96    -        io.containerd.runc.v2
31ba2de516182c59f50985fff21ba5095cc3fd9c2ec505c91597ad59e3a3d477    -        io.containerd.runc.v2
324491ec7925fc59dbcd2f706c3b2366068 42c610deede9b52accb4a5bf1ac29    -        io.containerd.runc.v2
3bcf6615bf38b2816c5a685a0ac7764c49e6400bdb99a969061400cce8f4cd38    -        io.containerd.runc.v2
4303de203167fa6cf75755cfc4399fed2831d4f154ebdc0eaf06935ce6d310ee    -        io.containerd.runc.v2
47f1bcdb5fbb11825a8b58b964eeaf85adc381ce631f7004bba29405731aed8b    -        io.containerd.runc.v2
6f14807e42c7494f460a51e6450c1cd2a170165d65d1a997ae0997f158328a79    -        io.containerd.runc.v2
7da02bcfda43c18ac2358886f68071aa741e4a3a69553614d365608508f08ef4    -        io.containerd.runc.v2
86c0a1eabbb53709a6fb4e57541c557fde7a1b5aaf3cffe20a5e6e49486111e7    -        io.containerd.runc.v2
a4b31c71dd976e36850bdf199d37240398801a0f30a36069a30e315533cb12aa    -        io.containerd.runc.v2
a8311b085b98e9beb9384beca0a7662f112bcf4c43ab38737a707c3063bea0de    -        io.containerd.runc.v2
c443d40b6cf5bbe3409495b1fad4148d651129b241001aa8f8acbb90c648cb59    -        io.containerd.runc.v2
d7646d5c92a65762e9cb0d2d71851c4be46c560e5360ce7a8c7c7358d9f358e7    -        io.containerd.runc.v2
f06012eb0065f4349cd0b6d10e236c75e9f8095581e004ffa091e9f710f6e066    -        io.containerd.runc.v2
$
```

Nous pouvons afficher la description d'un pod.

```
PS C:\Users\nnicolau> kubectl describe pod/monpod
Name:             monpod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.23.63.113
Start Time:       Thu, 30 Apr 2026 10:06:25 +0200
Labels:           run=monpod
Annotations:      <none>
Status:           Running
IP:               10.244.0.3
IPs:
  IP:  10.244.0.3
Containers:
  monpod:
    Container ID:   docker://3bcf6615bf38b2816c5a685a0ac7764c49e6400bdb99a969061400cce8f4cd38
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:6e23479198b998e5e25921dff8455837c7636a67111a04a635cf1bb363d199dc
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 30 Apr 2026 10:06:32 +0200
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-fr664 (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-fr664:
    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  13m   default-scheduler  Successfully assigned default/monpod to minikube
  Normal  Pulling    13m   kubelet            Pulling image "nginx:latest"
  Normal  Pulled     13m   kubelet            Successfully pulled image "nginx:latest" in 5.305s (5.305s including wait
ing). Image size: 160939956 bytes.
  Normal  Created    13m   kubelet            Container created
  Normal  Started    13m   kubelet            Container started
PS C:\Users\nnicolau>
```

Nous activons le port-forward pour accéder au site depuis l'extérieur.

```
PS C:\Users\nnicolau> 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
```



Après être accédé au site, nous détruisons le pod.

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

3. Le tableau de bord de Kubernetes.

Nous installons le module dashboard pour pouvoir gérer le tout via une interface.

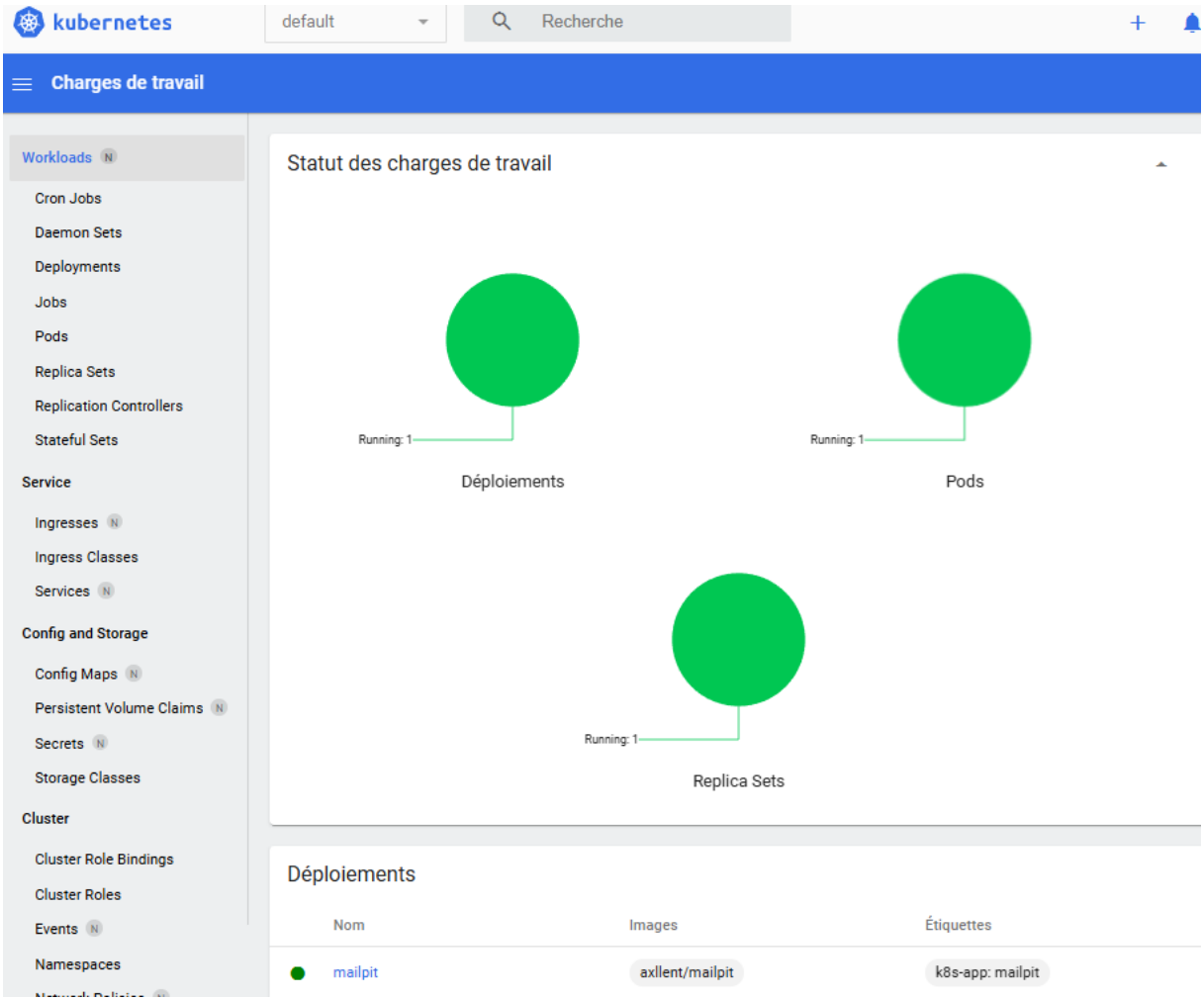
```
PS C:\Users\nnicolau> minikube addons enable dashboard
    dashboard est un addon maintenu par Kubernetes. Pour toute question, contactez minikube sur GitHub.
Vous pouvez consulter la liste des mainteneurs de minikube sur : https://github.com/kubernetes/minikube/blob/master/OWNERS
    ▪ Utilisation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

        minikube addons enable metrics-server

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

Nous activons la métrique pour bénéficier des fonctionnalités.

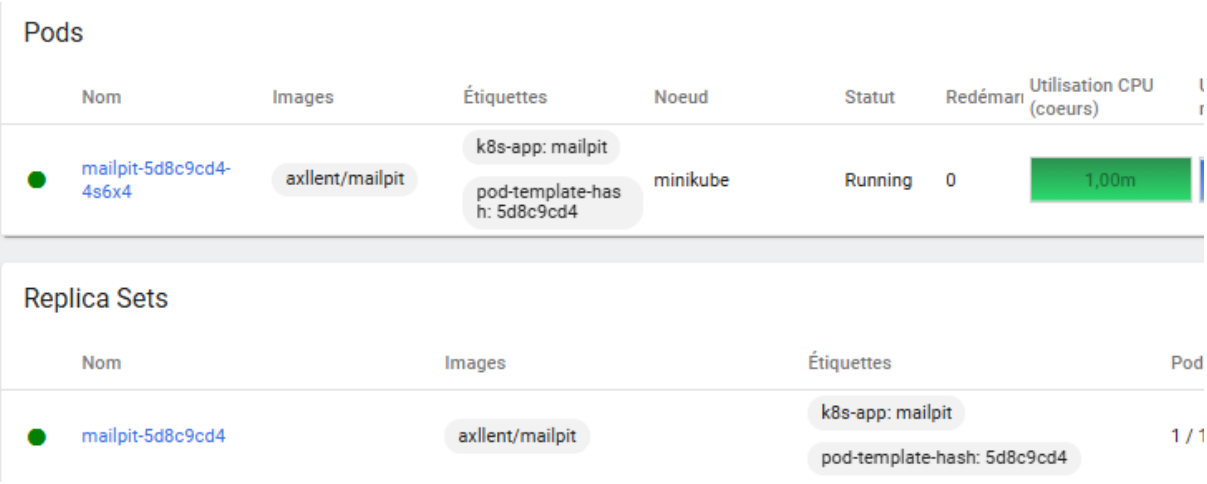
```
PS C:\Users\nnicolau> minikube addons enable metrics-server
    metrics-server est un addon maintenu par Kubernetes. Pour toute question, contactez minikube sur GitHub.
Vous pouvez consulter la liste des mainteneurs de minikube sur : https://github.com/kubernetes/minikube/blob/master/OWNERS
    ▪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

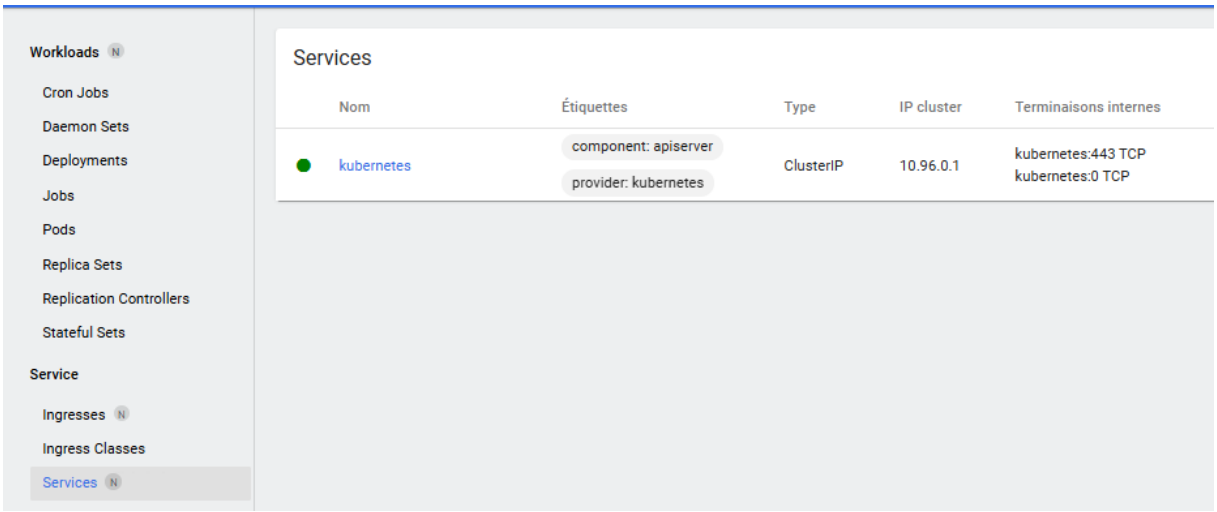
Une fois le module lancé, nous accédons au tableau de bord.

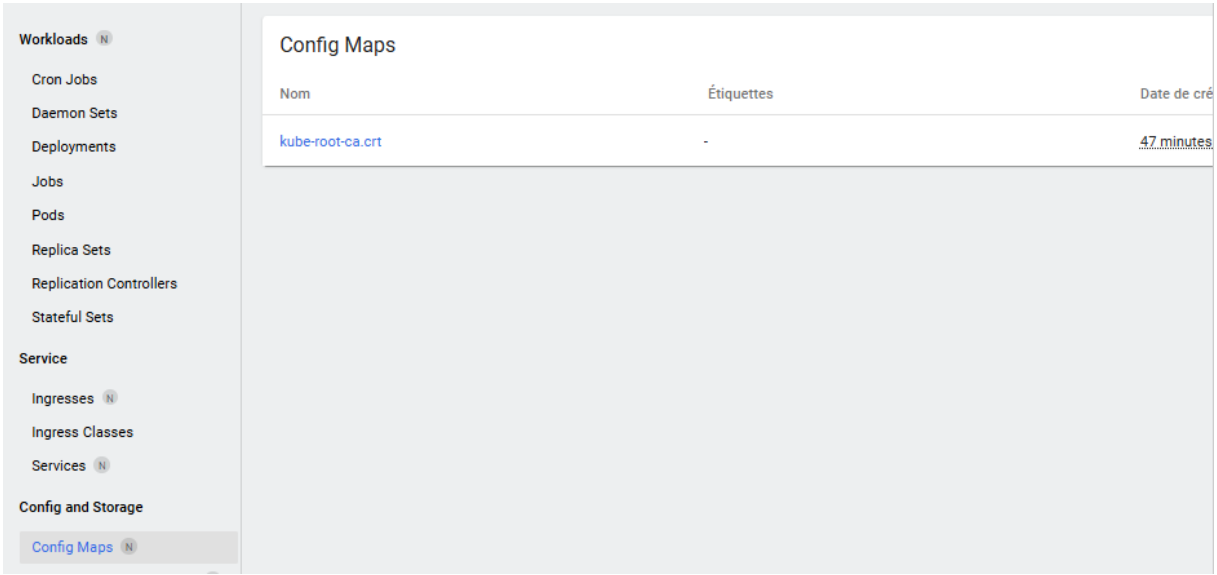
```
PS C:\Users\nnicolau> minikube dashboard
🤔  Vérification de l'état du tableau de bord...
🚀  Lancement du proxy...
🤔  Vérification de l'état du proxy...
🎉  Ouverture de http://127.0.0.1:60323/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes
-dashboard:/proxy/ dans votre navigateur par défaut...
```

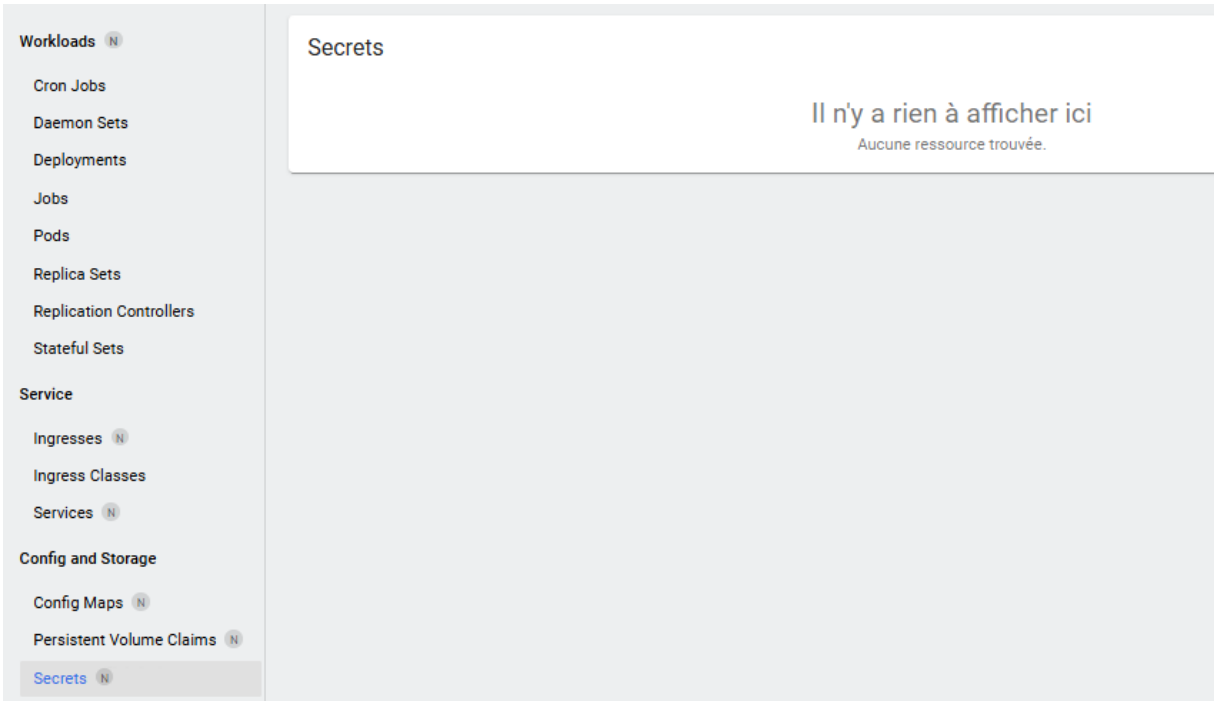


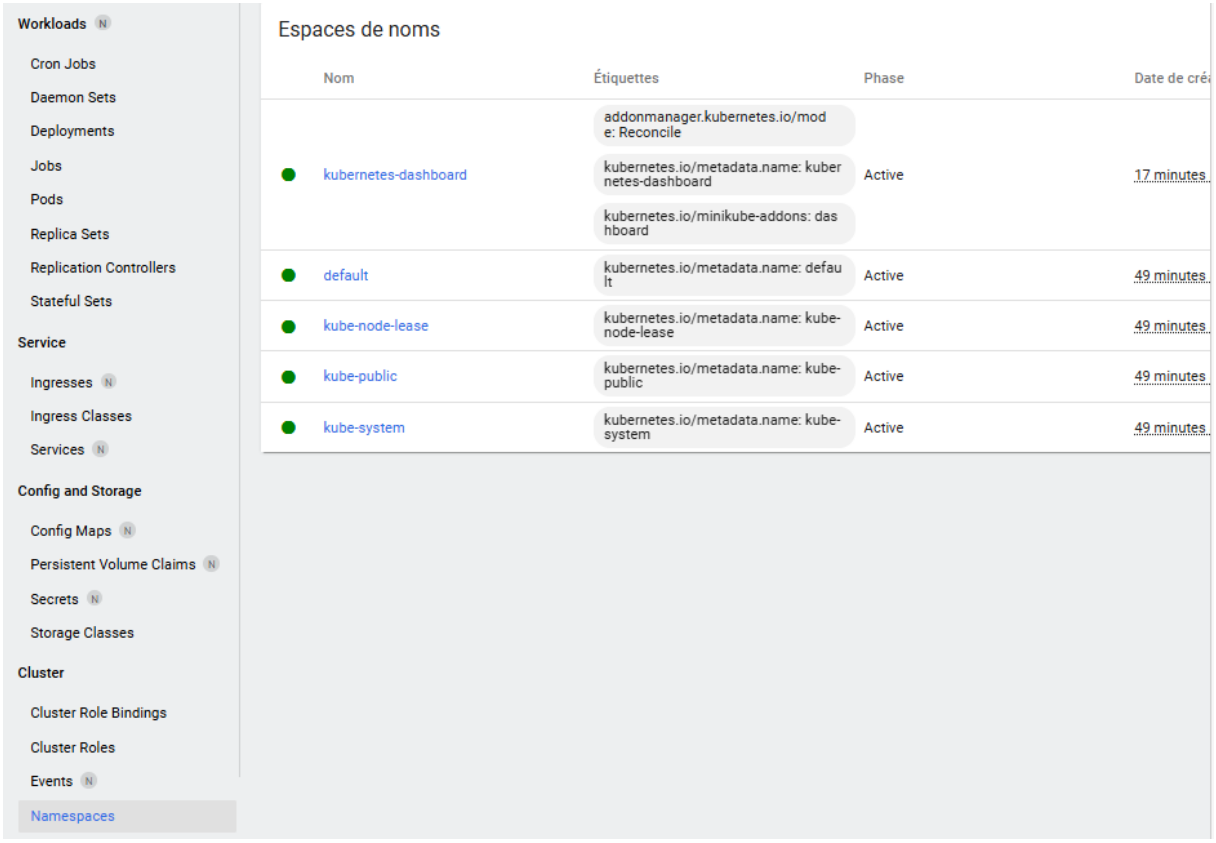
Depuis celui-ci, nous pouvons consulter les différents éléments comme les pods, réplicas, services…



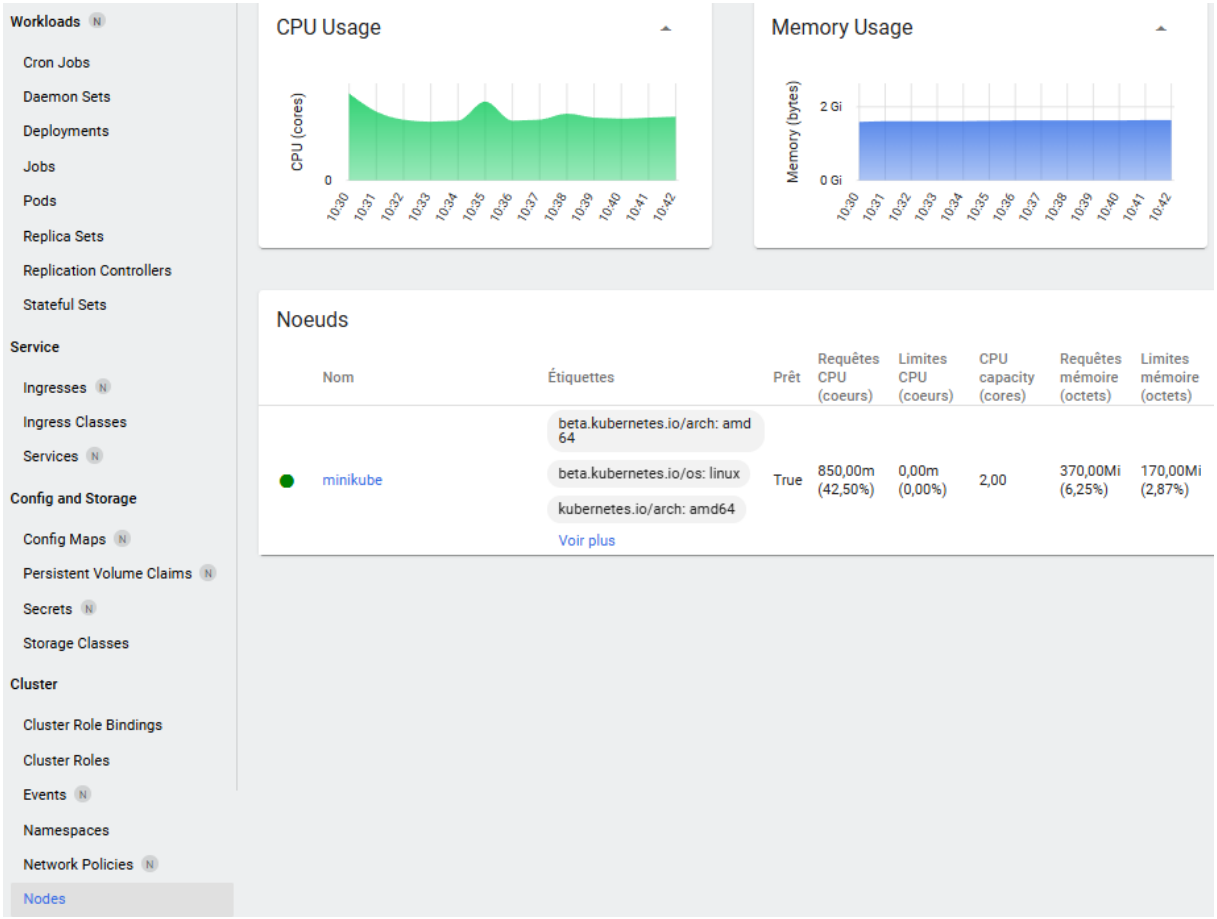

Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Services
Nom
Étiquettes
Type
IP cluster
Terminaisons internes
kubernetes
component: apiserver
provider: kubernetes
ClusterIP
10.96.0.1
kubernetes:443 TCP
kubernetes:0 TCP


Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Config Maps
Nom
Étiquettes
kube-root-ca.crt
-
47 minutes

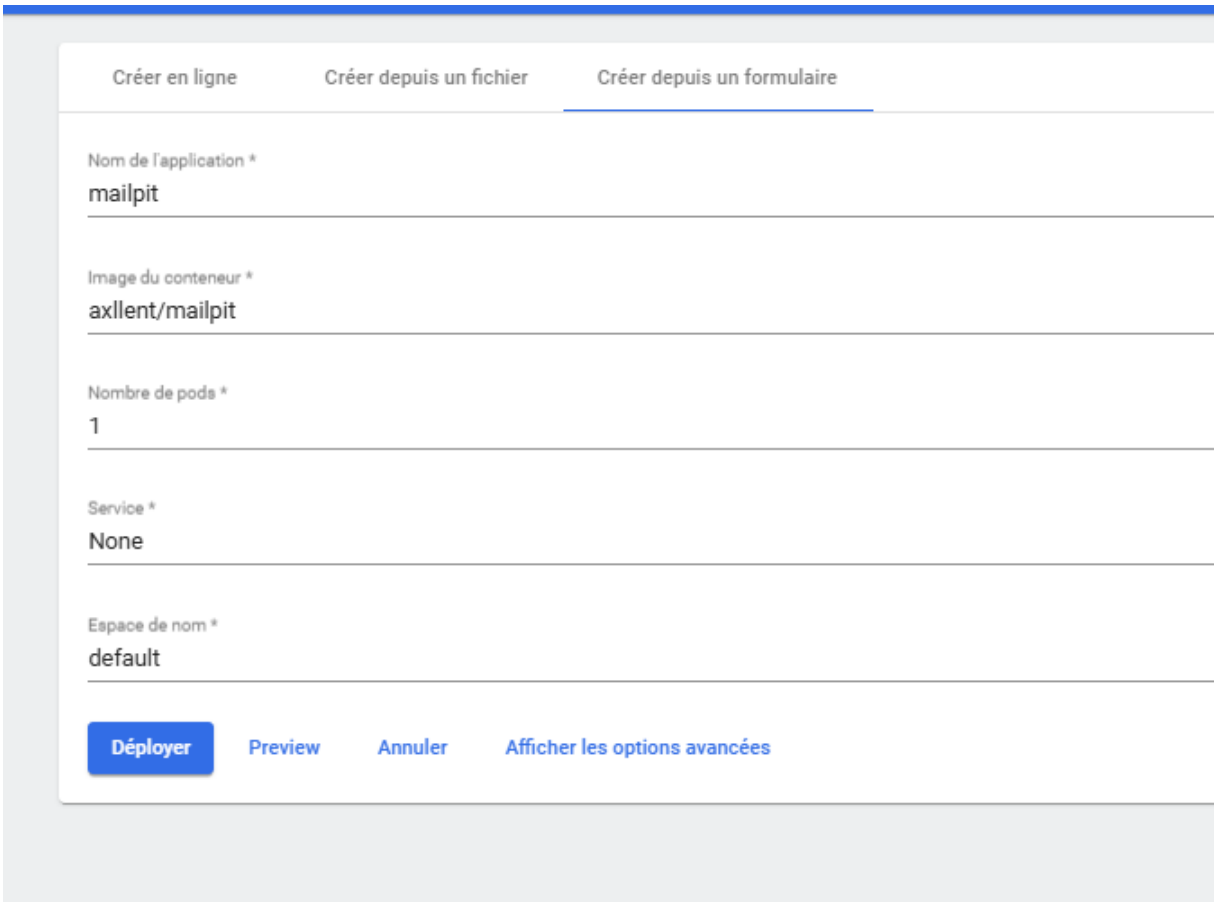

Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Secrets
Il n'y a rien à afficher ici
Aucune ressource trouvée.

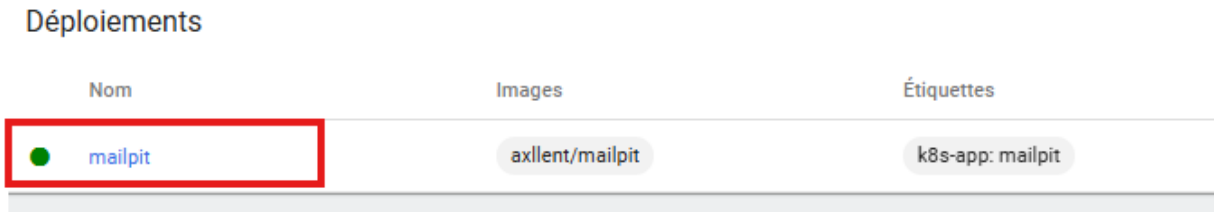


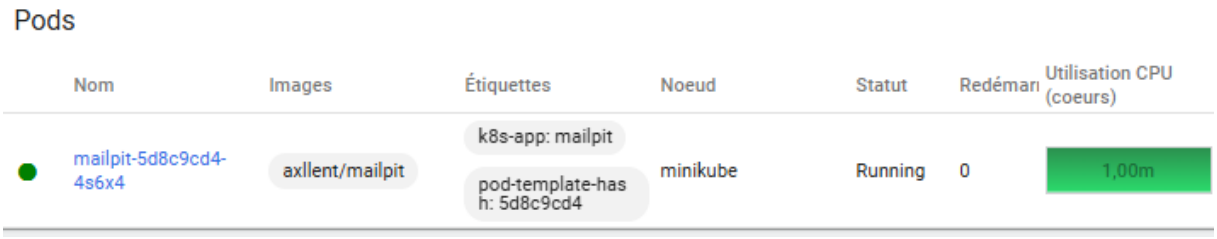
Dans l'onglet nodes, nous pouvons vérifier la consommation de ceci sur la machine.

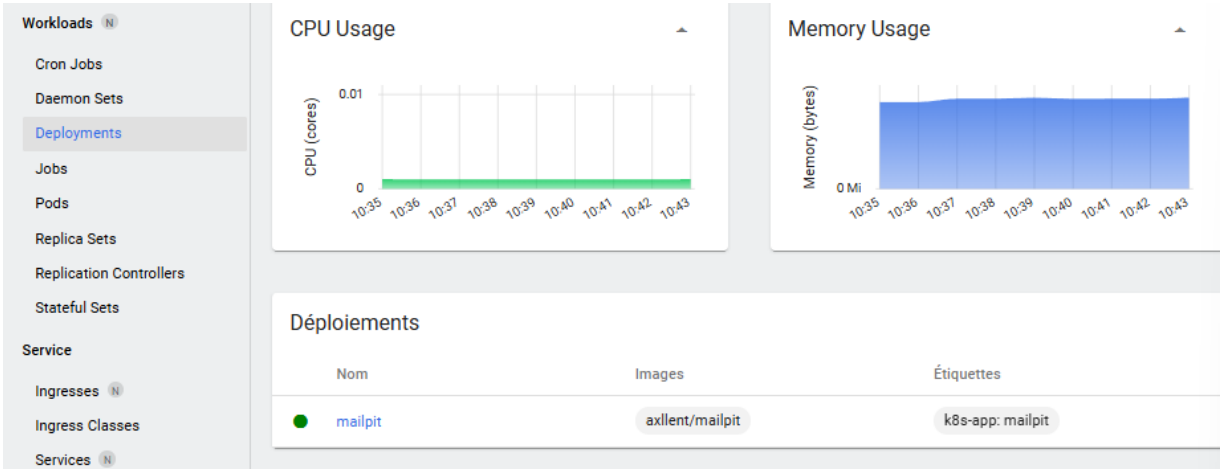


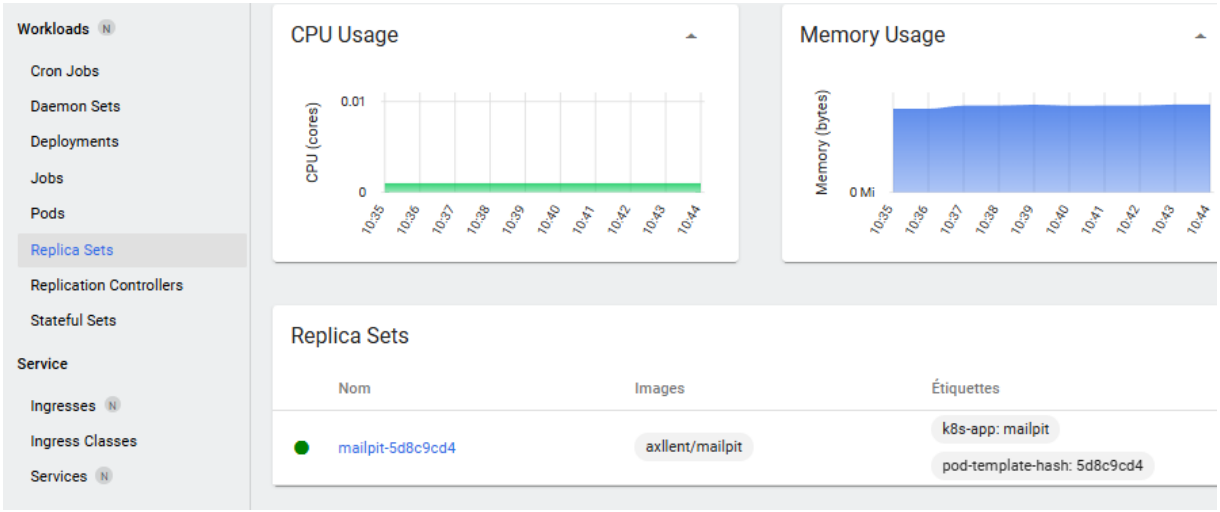
Nous créons le déploiement Mailpit.



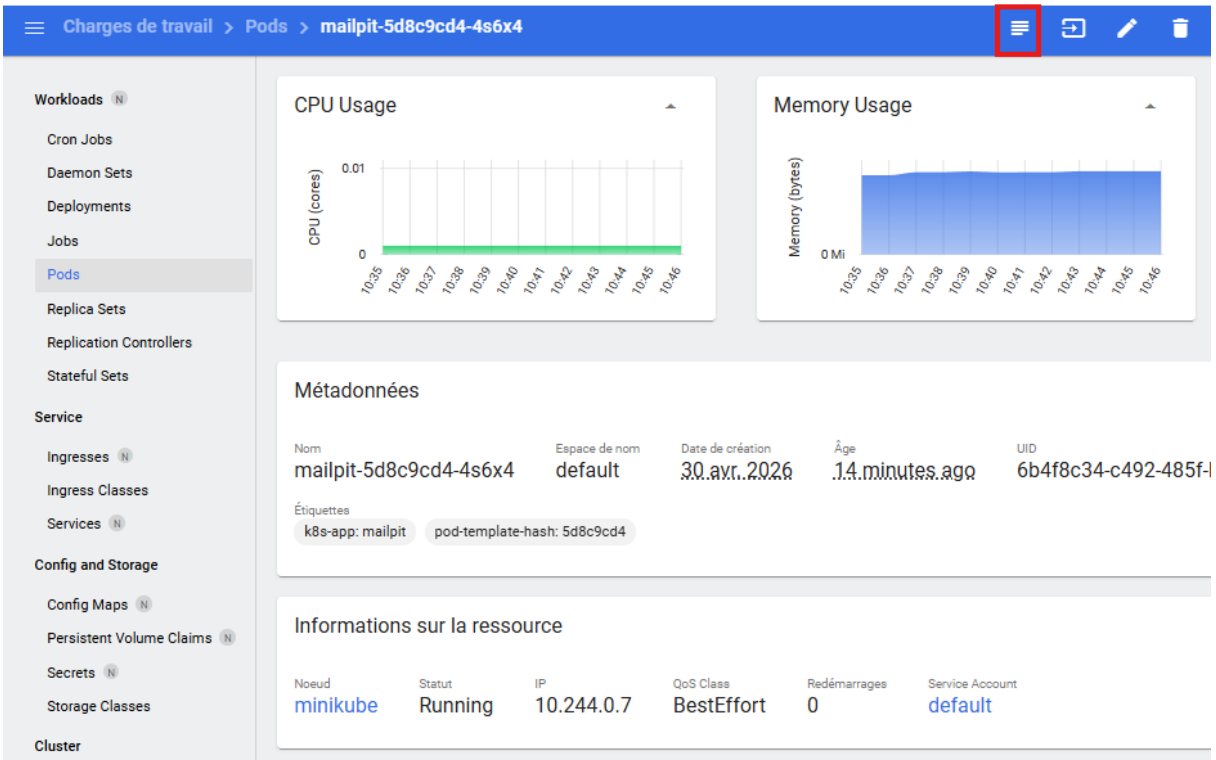


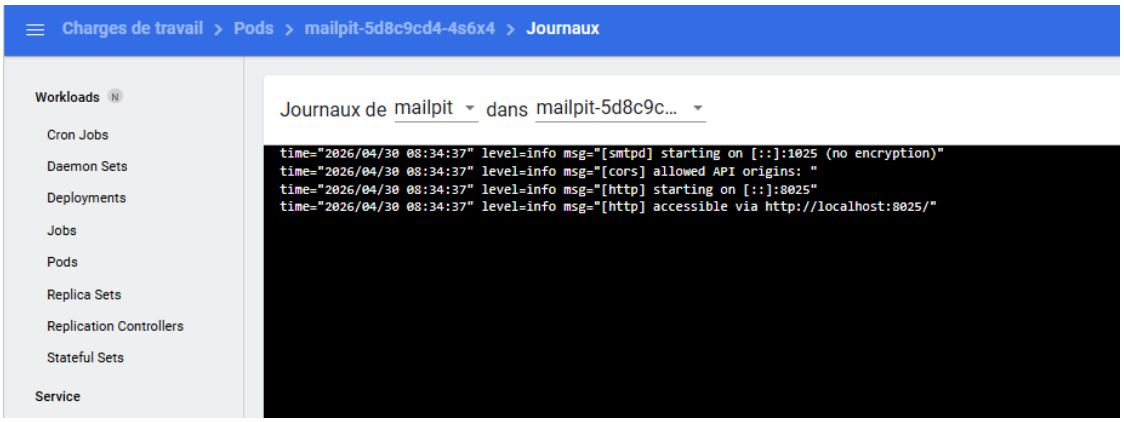




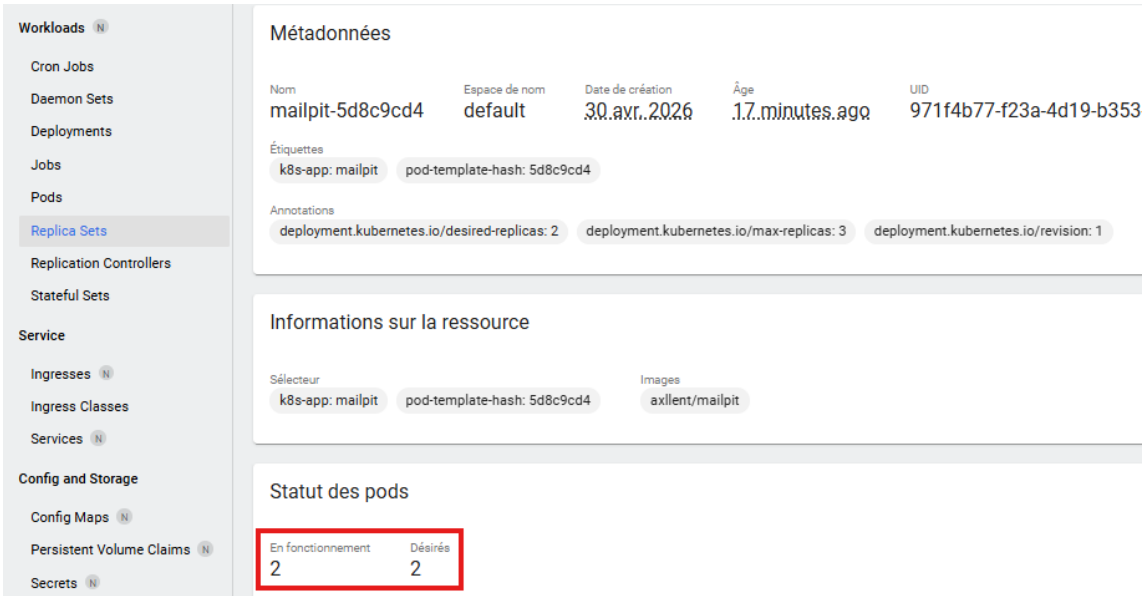


Nous consultons les journaux.

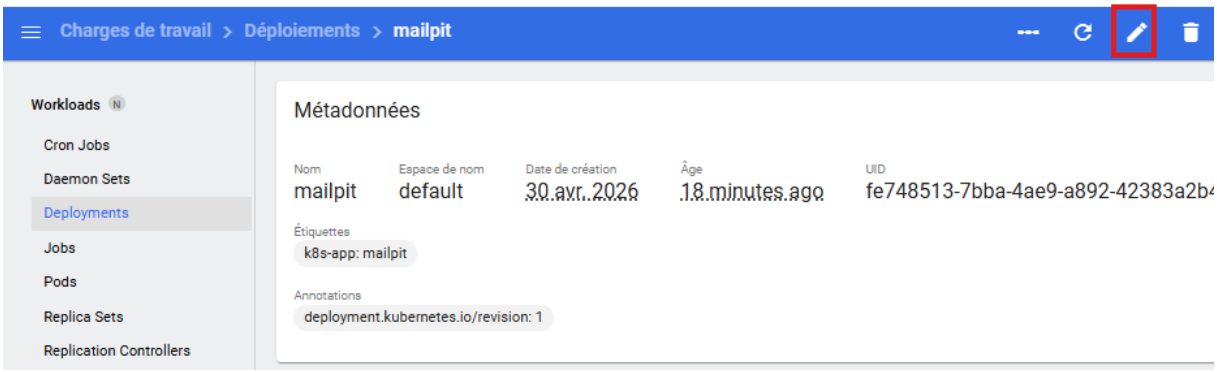




Nous modifions le nombre de déploiement.



Nous modifions le mailpit.



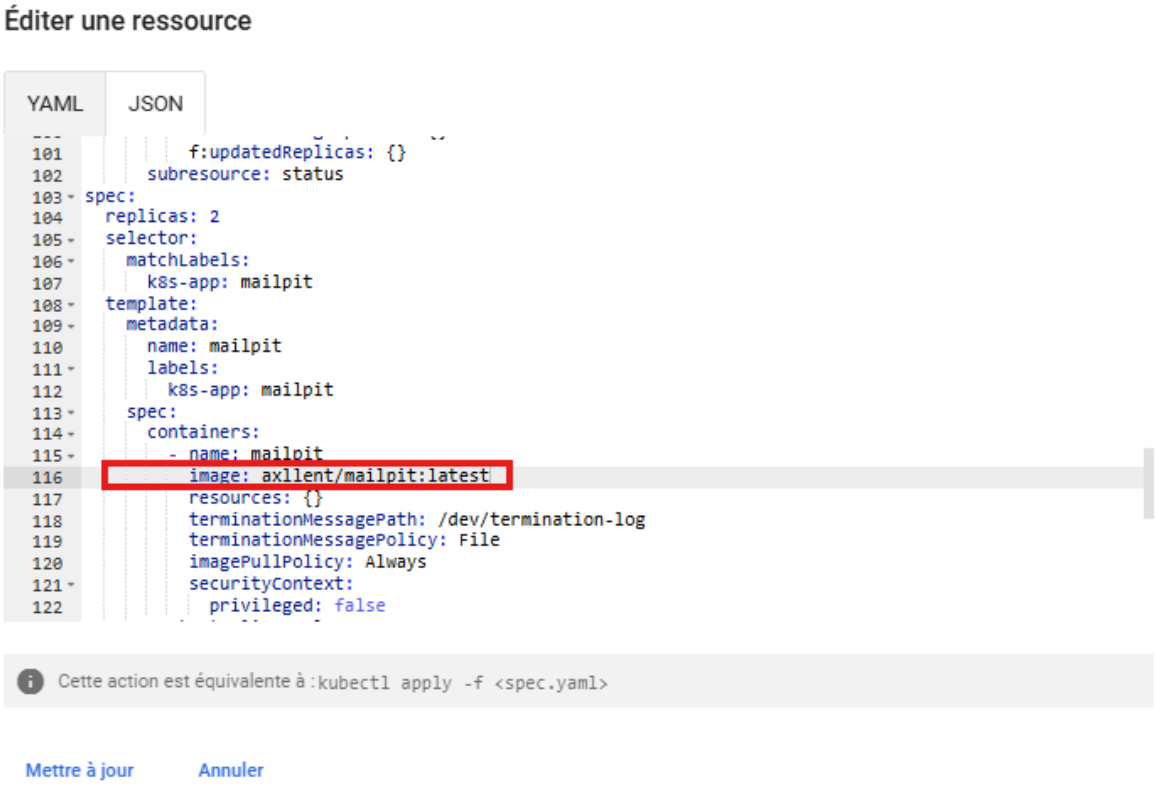
Éditer une ressource

YAML JSON

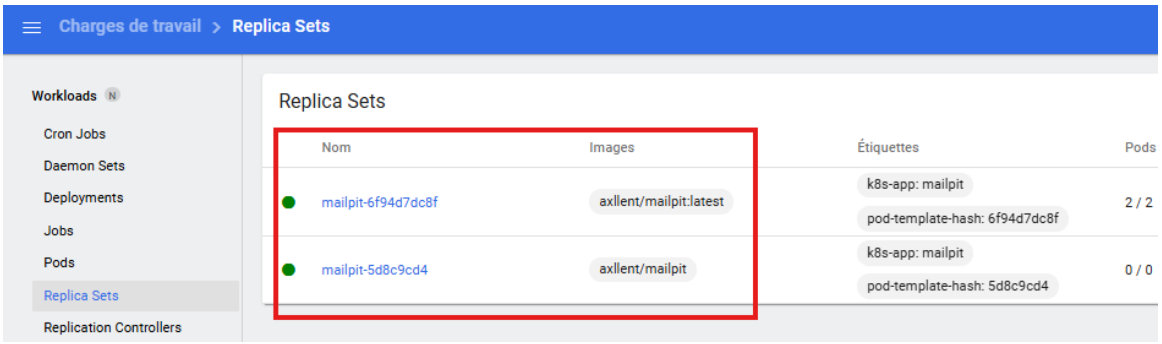
```yaml
kind: Deployment
apiVersion: apps/v1
metadata:
  name: mailpit
  namespace: default
  uid: fe748513-7bba-4ae9-a892-42383a2b4587
  resourceVersion: '3314'
  generation: 2
  creationTimestamp: '2026-04-30T08:34:32Z'
  labels:
    k8s-app: mailpit
  annotations:
    deployment.kubernetes.io/revision: '1'
  managedFields:
    - manager: dashboard
      operation: Update
      apiVersion: apps/v1
      fieldsType: FieldsV1
      fieldsV1:
        f:spec:
          f:replicas: {}
      subresource: scale
```

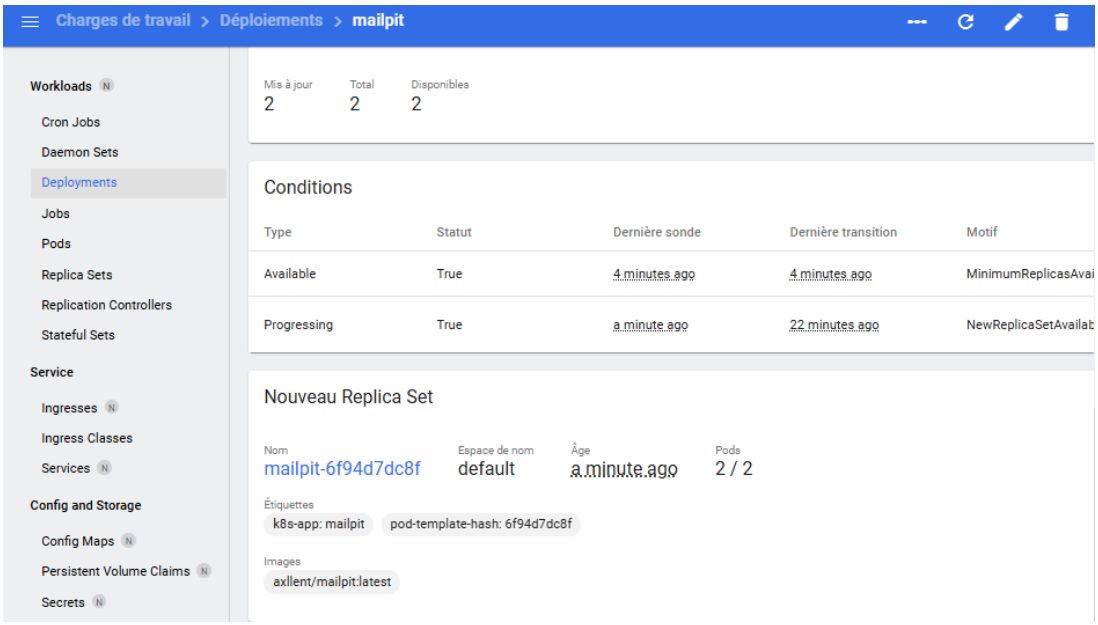
Cette action est équivalente à : `kubectl apply -f <spec.yaml>`

Mettre à jour Annuler



Une fois la ressource éditée, nous constatons qu'il crée un deuxième réplica.





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

Nous supprimons Mailpit créé précédemment.

```
PS C:\Users\nnicolau> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mailpit   2/2     2            2           24m
PS C:\Users\nnicolau> kubectl delete deployment mailpit
deployment.apps "mailpit" deleted from default namespace
PS C:\Users\nnicolau> kubectl get deployment
No resources found in default namespace.
PS C:\Users\nnicolau>
```

Nous recréons un déploiement, mais depuis le cmd.

```
PS C:\Users\nnicolau> kubectl create deployment mailpit --image=axllent/mailpit
deployment.apps/mailpit created
PS C:\Users\nnicolau> kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mailpit   1/1     1            1           10s
PS C:\Users\nnicolau> kubectl get deployment -o wide
NAME      READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES            SELECTOR
mailpit   1/1     1            1           29s   mailpit      axllent/mailpit   app=mailpit
PS C:\Users\nnicolau>
```

```
PS C:\Users\nnicolau> kubectl describe deployment mailpit
Name:                   mailpit
Namespace:              default
CreationTimestamp:      Thu, 30 Apr 2026 11:01:26 +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  75s   deployment-controller  Scaled up replica set mailpit-7b96b48c5d from
 0 to 1
PS C:\Users\nnicolau>
```

Nous affichons les réplicats.

```
PS C:\Users\nnicolau> kubectl get replicaset
NAME                 DESIRED   CURRENT   READY   AGE
mailpit-7b96b48c5d   1         1         1       2m34s
PS C:\Users\nnicolau> 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  3m18s  replicaset-controller  Created pod: mailpit-7b96b48c5d-2xj79
PS C:\Users\nnicolau>
```

Nous affichons l'état du pod.

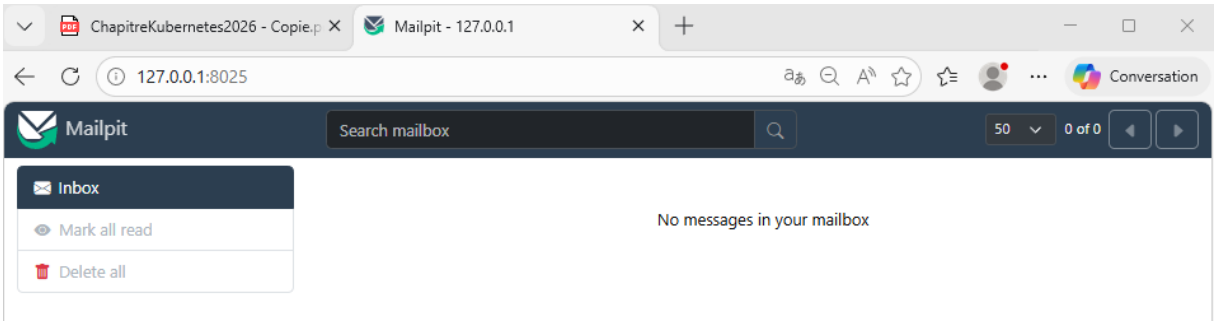
```
PS C:\Users\nnicolau> kubectl get pods
NAME                       READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-2xj79   1/1     Running   0          4m5s
PS C:\Users\nnicolau> kubectl describe pods mailpit-7b96b48c5d-2xj79
Name:             mailpit-7b96b48c5d-2xj79
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.23.63.113
Start Time:       Thu, 30 Apr 2026 11:01:26 +0200
Labels:           app=mailpit
                  pod-template-hash=7b96b48c5d
Annotations:      <none>
Status:           Running
IP:               10.244.0.11
IPs:
  IP:           10.244.0.11
Controlled By:  ReplicaSet/mailpit-7b96b48c5d
Containers:
  mailpit:
    Container ID:   docker://8e4bb07bb948591a36f4ec19c72f3700985dd8645634a81c4acc4c8d9ab07935
    Image:          axllent/mailpit
    Image ID:       docker-pullable://axllent/mailpit@sha256:757f22b56c1da03570afdb3d259effe5091018008a81bbedc8158c
ee7e16fdbc
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 30 Apr 2026 11:01:27 +0200
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-stqx4 (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-stqx4:
    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  4m24s  default-scheduler  Successfully assigned default/mailpit-7b96b48c5d-2xj79 to minikube
  Normal  Pulling    4m24s  kubelet            Pulling image "axllent/mailpit"
  Normal  Pulled     4m23s  kubelet            Successfully pulled image "axllent/mailpit" in 904ms (904ms includin
g waiting). Image size: 36229179 bytes.
  Normal  Created    4m23s  kubelet            Container created
  Normal  Started    4m23s  kubelet            Container started
PS C:\Users\nnicolau>
```

Nous affichons les logs du pod et des conteneurs.

```
PS C:\Users\nnicolau> kubectl logs mailpit-7b96b48c5d-2xj79
time="2026/04/30 09:01:27" level=info msg="[smtpd] starting on [::]:1025 (no encryption)"
time="2026/04/30 09:01:27" level=info msg="[cors] allowed API origins: "
time="2026/04/30 09:01:27" level=info msg="[http] starting on [::]:8025"
time="2026/04/30 09:01:27" level=info msg="[http] accessible via http://localhost:8025/"
PS C:\Users\nnicolau> kubectl logs mailpit-7b96b48c5d-2xj79 -c mailpit
time="2026/04/30 09:01:27" level=info msg="[smtpd] starting on [::]:1025 (no encryption)"
time="2026/04/30 09:01:27" level=info msg="[cors] allowed API origins: "
time="2026/04/30 09:01:27" level=info msg="[http] starting on [::]:8025"
time="2026/04/30 09:01:27" level=info msg="[http] accessible via http://localhost:8025/"
PS C:\Users\nnicolau>
```

Nous accédons à l'application Mailpit.

```
PS C:\Users\nnicolau> kubectl port-forward mailpit-7b96b48c5d-2xj79 8025
Forwarding from 127.0.0.1:8025 -> 8025
Forwarding from [::1]:8025 -> 8025
```




```
PS C:\Users\nnicolau> kubectl port-forward mailpit-7b96b48c5d-2xj79 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
```

Nous ajoutons un DNS dans Kubernetes.

```
PS C:\Users\nnicolau> kubectl expose deployment/mailpit --port 1025,8025
service/mailpit exposed
PS C:\Users\nnicolau>
```

Nous lançons une requête DNS depuis le pod en interactif.

```
PS C:\Users\nnicolau> kubectl exec -it mailpit-7b96b48c5d-2xj79 -- sh
/ # getent hosts mailpit
10.99.107.44      mailpit.default.svc.cluster.local  mailpit.default.svc.cluster.local mailpit
/ #
```

Nous lançons un pod de test.

```
PS C:\Users\nnicolau> 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.svc.cluster.local: NXDOMAIN


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

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

Name:   mailpit.default.svc.cluster.local
Address: 10.99.107.44

/ # 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\nnicolau>
```

Nous augmentons le niveau de résilience de la réplication.

```
PS C:\Users\nnicolau> kubectl scale deployment mailpit --replicas=2
deployment.apps/mailpit scaled
PS C:\Users\nnicolau> kubectl get deployment mailpit
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mailpit   2/2     2            2           20m
PS C:\Users\nnicolau> kubectl get pods -l app=mailpit
NAME                       READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-2xj79   1/1     Running   0          20m
mailpit-7b96b48c5d-6rs4g   1/1     Running   0          59s
PS C:\Users\nnicolau>
```

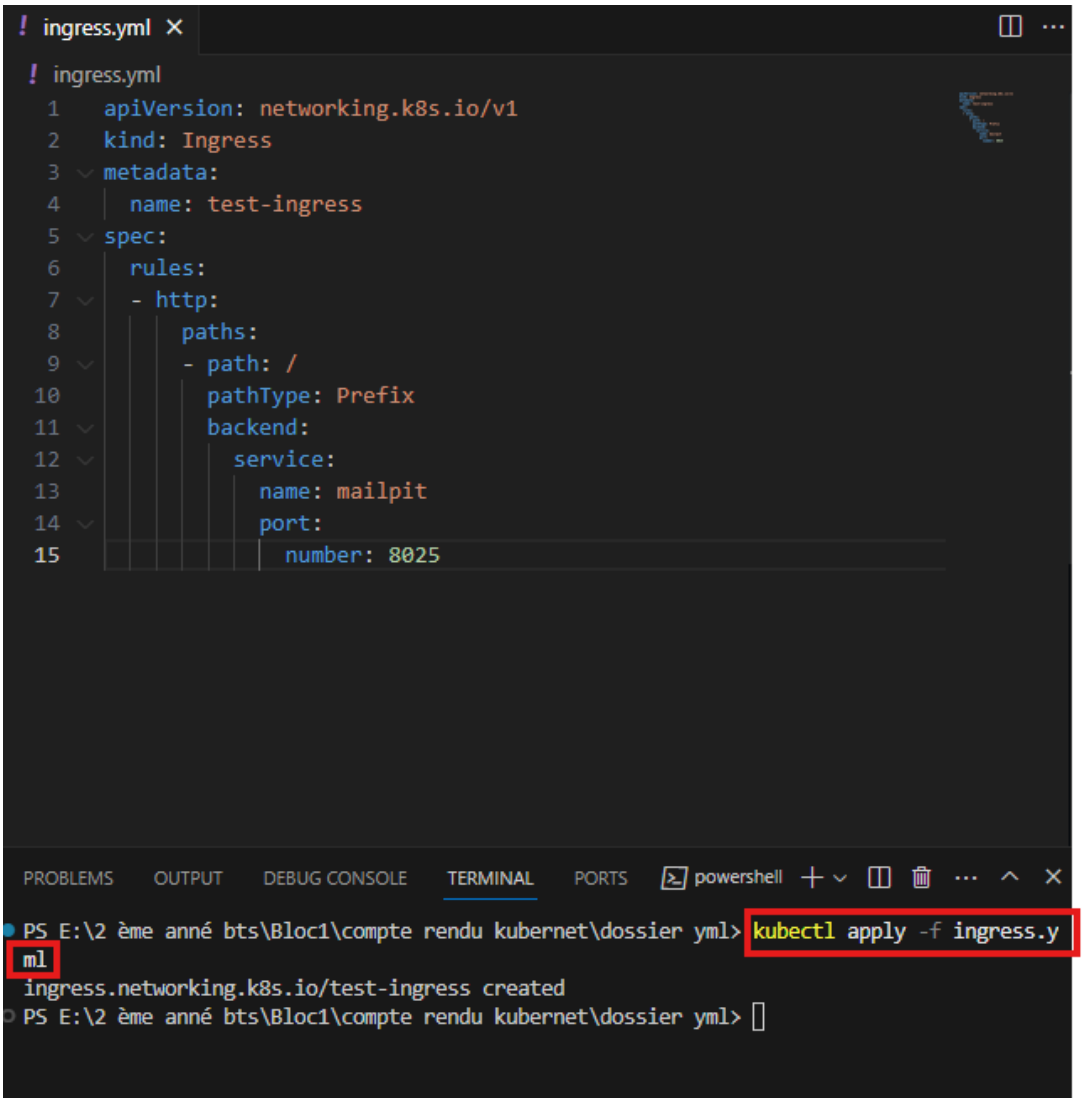
Nous installons et activons ingress dans minikube.

```
PS C:\Users\nnicolau> 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/OWNERS
    ▪ 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\nnicolau>
```

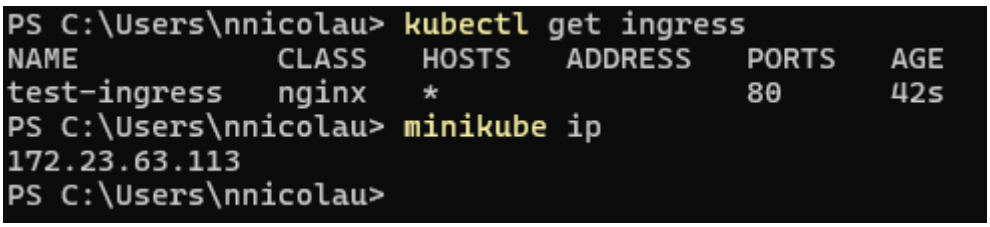
```
PS C:\Users\nnicolau> kubectl get namespace
NAME                   STATUS   AGE
default                Active   89m
ingress-nginx          Active   55s
kube-node-lease        Active   89m
kube-public            Active   89m
kube-system            Active   89m
kubernetes-dashboard   Active   57m
PS C:\Users\nnicolau>
```

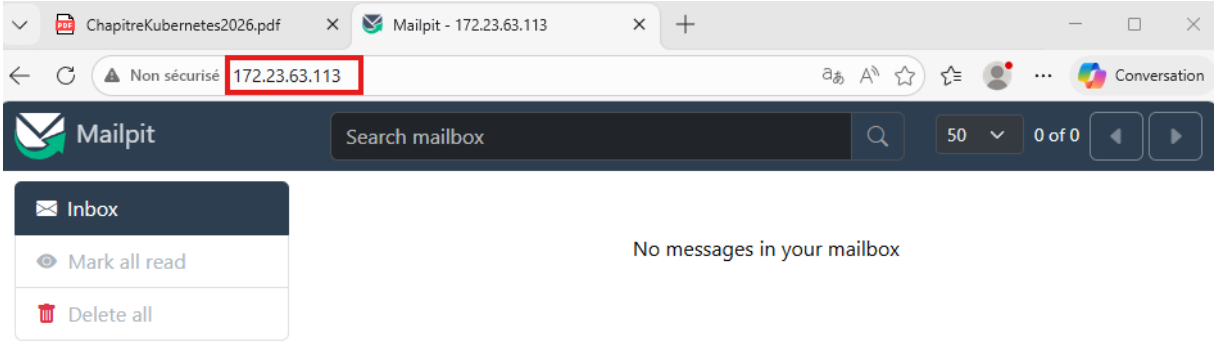
```
PS C:\Users\nnicolau> kubectl -n ingress-nginx get pods -l app.kubernetes.io/name
NAME                                        READY   STATUS      RESTARTS   AGE
ingress-nginx-admission-create-g6xt4        0/1     Completed   0          118s
ingress-nginx-admission-patch-2hbkg         0/1     Completed   0          118s
ingress-nginx-controller-596f8778bc-c7hgr   1/1     Running     0          118s
PS C:\Users\nnicolau>
```

Nous créons une règle d'impression.



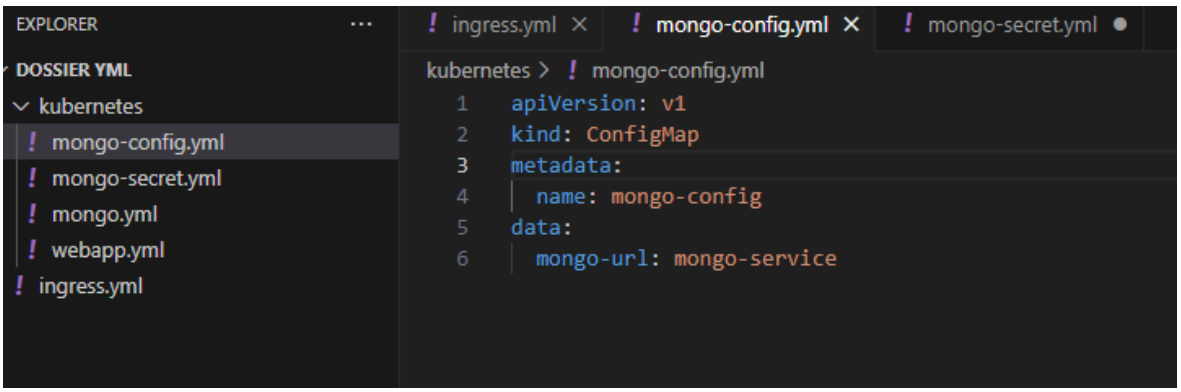
Nous constatons que la règle a bien été prise en compte.



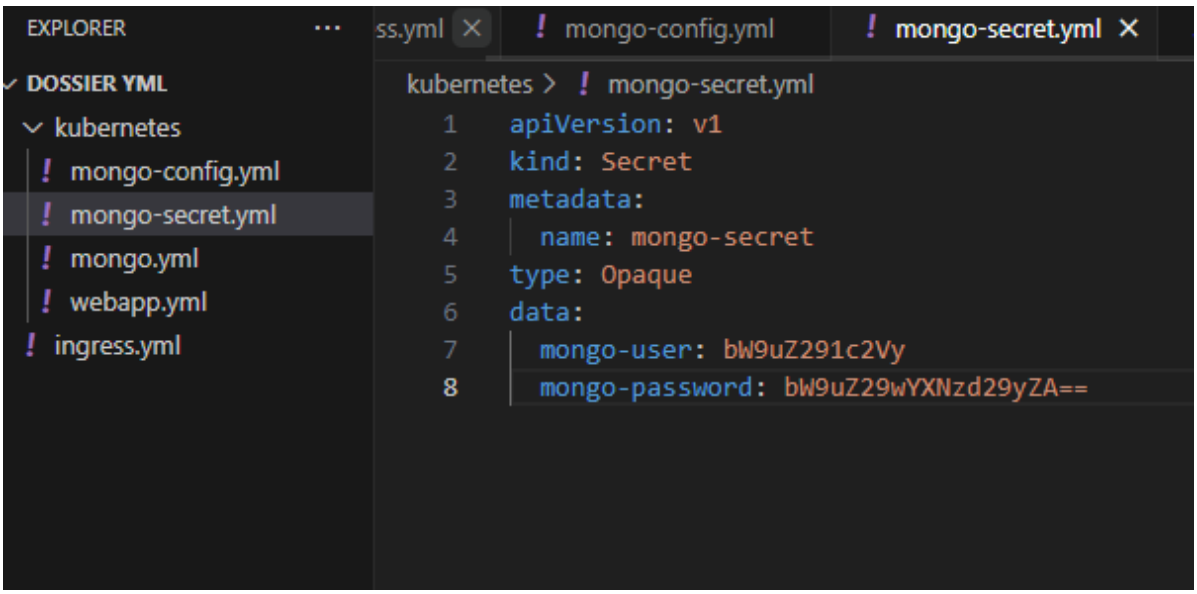


5. Automatisation de déploiement par fichier YAML.

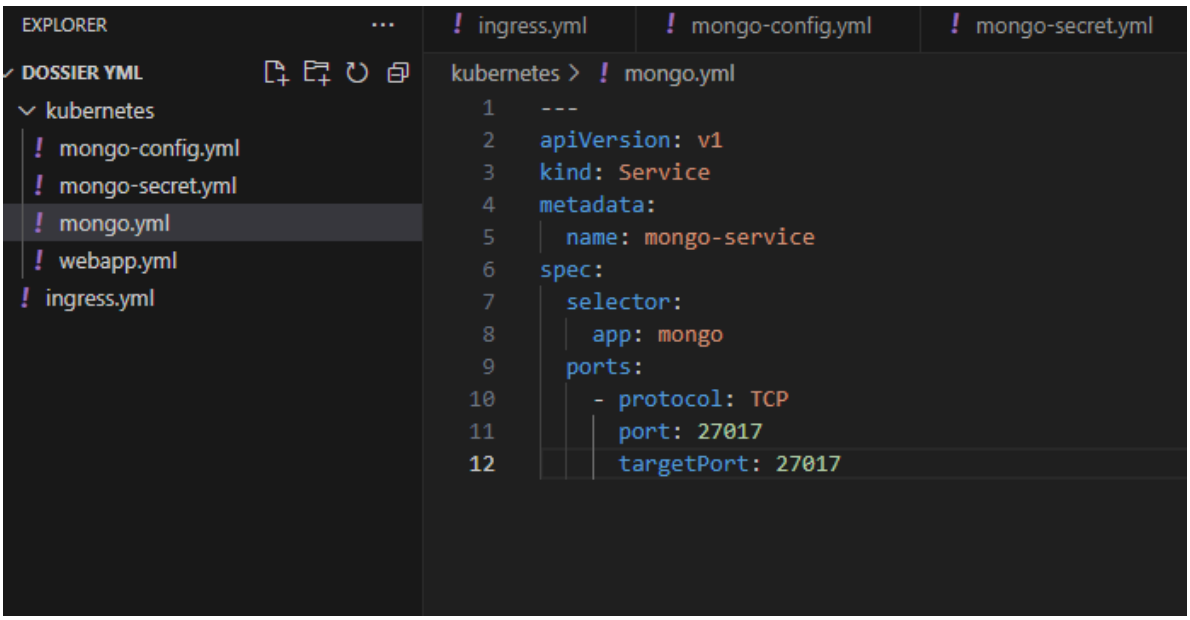
Nous créons le service configMap.



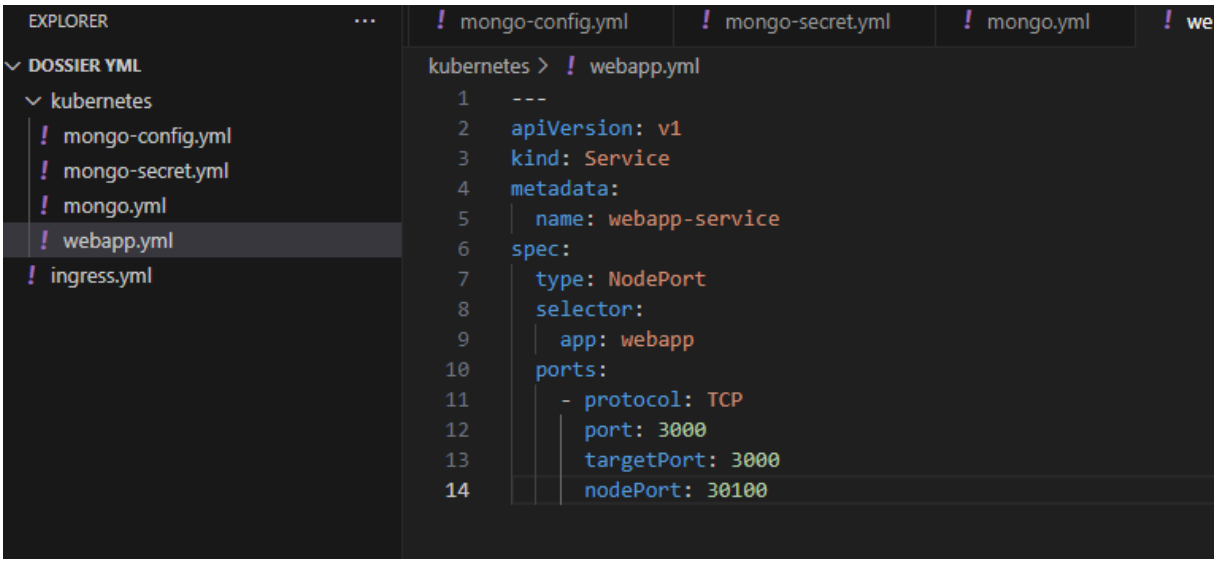
Nous créons le secret pour les MDP.



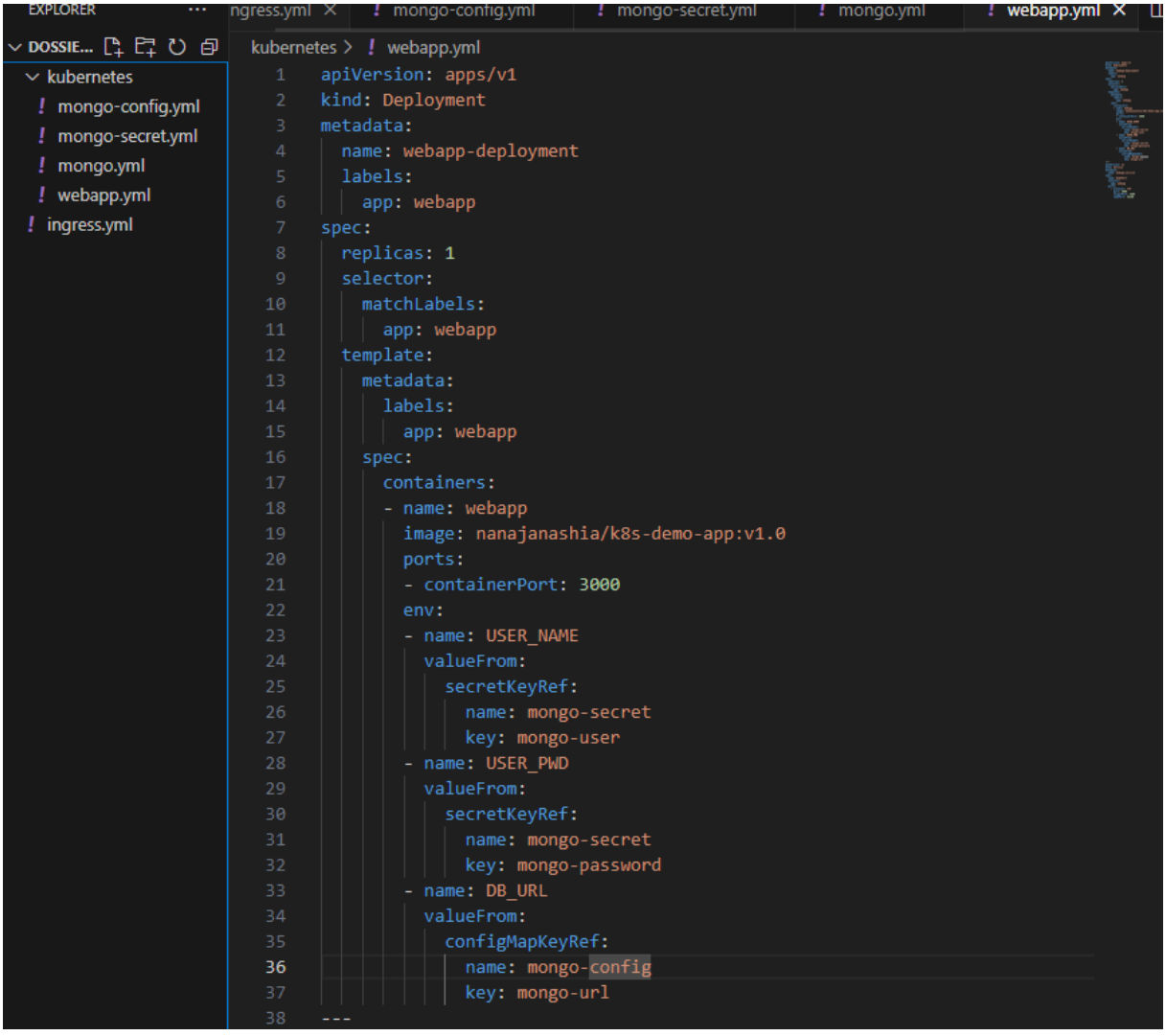
Nous créons un service pour la communication entre le pod front et le pod back end.



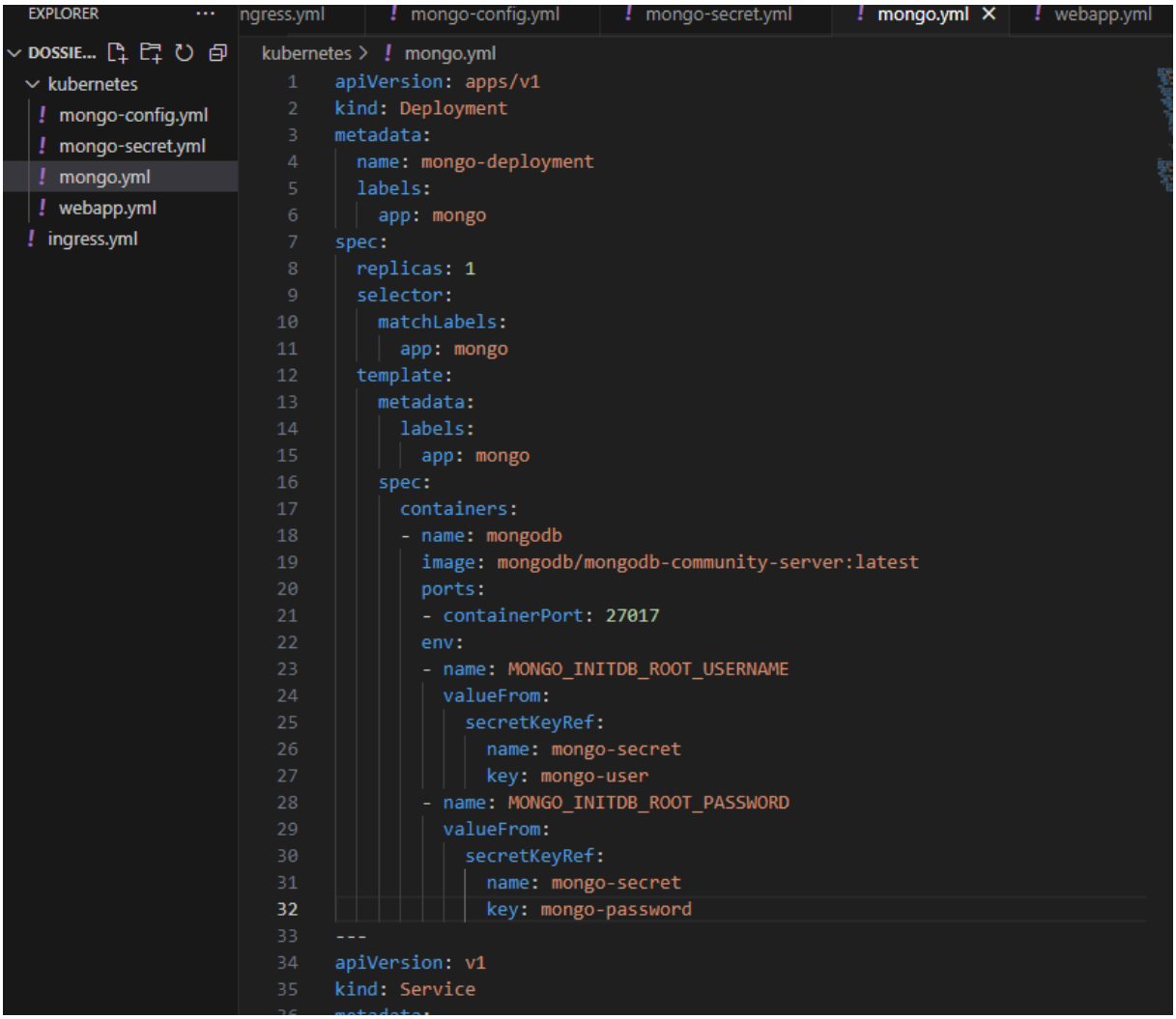
Nous créons un service externe qui permet de joindre le pod front depuis l'extérieur du cluster.



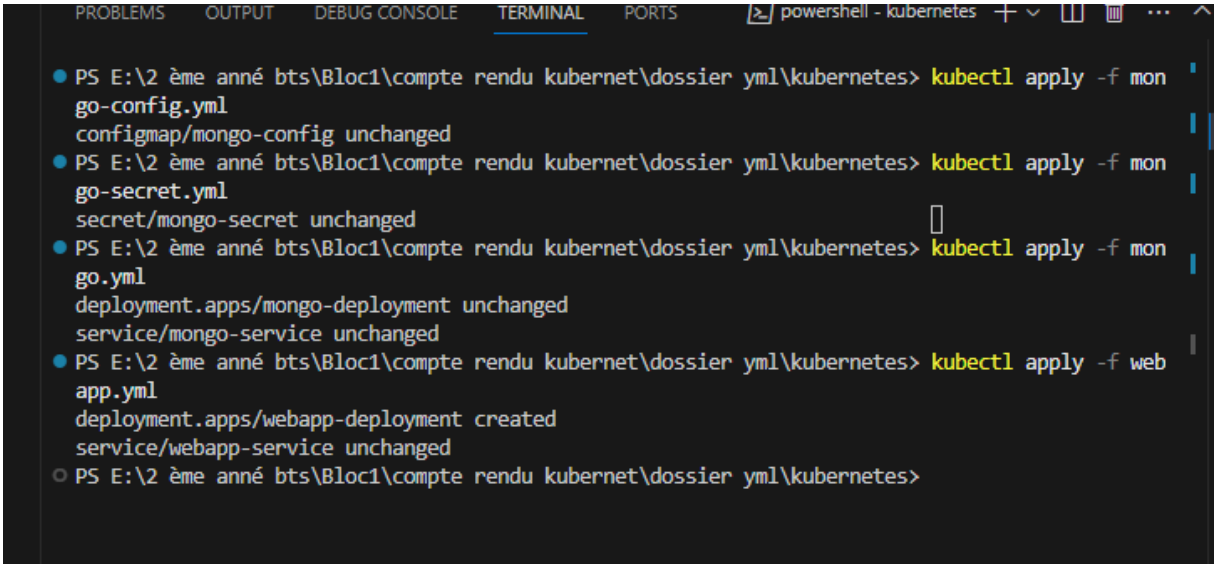
Nous créons un composant de type déploiement.



Nous créons un fichier pour déployer l'application mongo.



Nous exécutons les scripts.



Nous listons les objets crée.

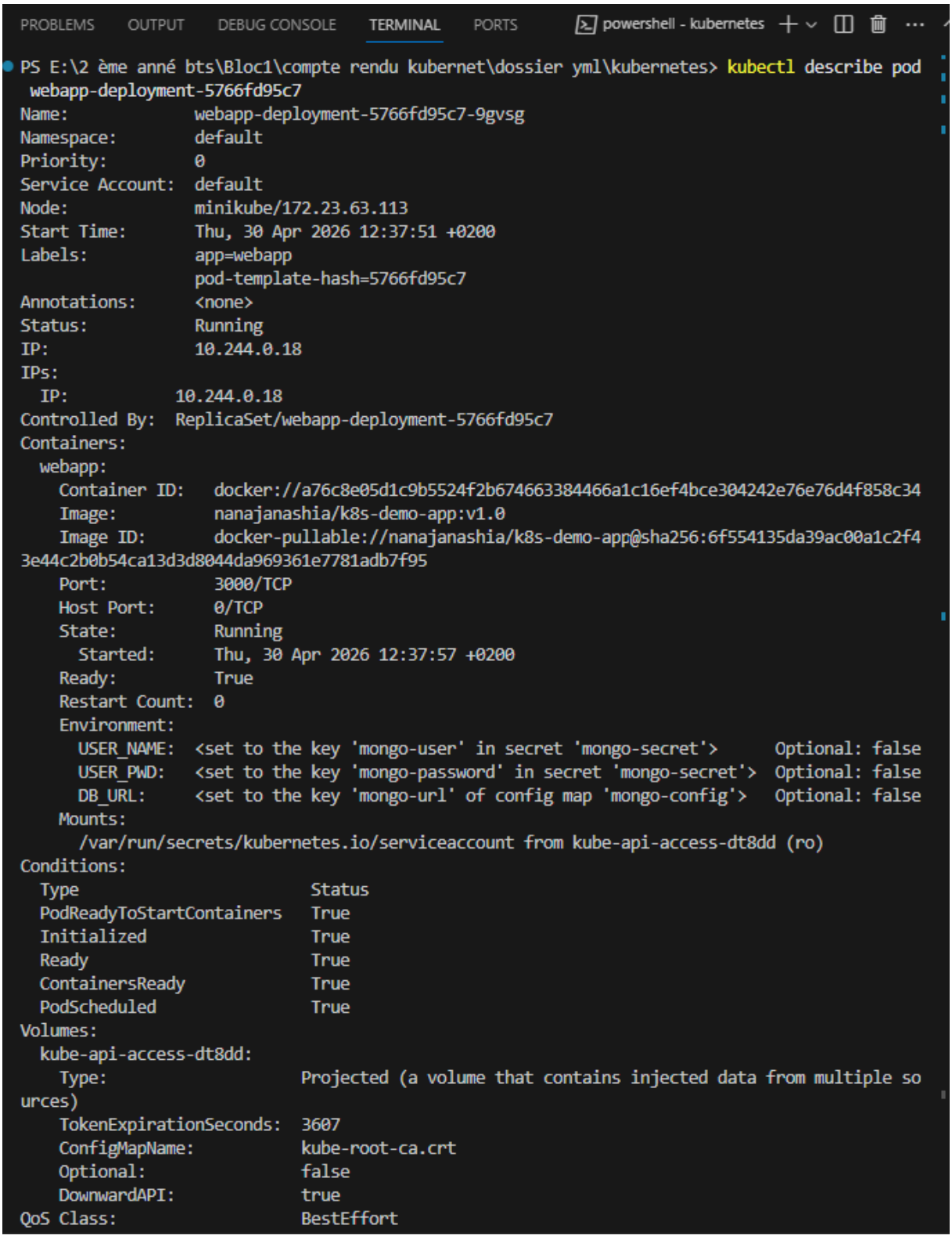
```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/mailpit-7b96b48c5d-2xj79             1/1     Running   0          98m
pod/mailpit-7b96b48c5d-6rs4g             1/1     Running   0          79m
pod/mongo-deployment-744864fdd7-kbk7c    1/1     Running   0          3m46s
pod/webapp-deployment-5766fd95c7-9gvsg   1/1     Running   0          2m13s

NAME                     TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)             AGE
service/kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP             165m
service/mailpit          ClusterIP   10.99.107.44     <none>        1025/TCP,8025/TCP   84m
service/mongo-service    ClusterIP   10.101.126.187   <none>        27017/TCP           19m
service/webapp-service   NodePort    10.106.184.88    <none>        3000:30100/TCP      3m27s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mailpit             2/2     2            2           98m
deployment.apps/mongo-deployment    1/1     1            1           3m46s
deployment.apps/webapp-deployment   1/1     1            1           2m13s

NAME                                           DESIRED   CURRENT   READY   AGE
replicaset.apps/mailpit-7b96b48c5d             2         2         2       98m
replicaset.apps/mongo-deployment-744864fdd7    1         1         1       3m46s
replicaset.apps/webapp-deployment-5766fd95c7   1         1         1       2m13s
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes>
```

Nous listons les autres objets.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl get configma
p
NAME              DATA   AGE
kube-root-ca.crt  1      167m
mongo-config      1      25m
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl get secret
NAME           TYPE     DATA   AGE
mongo-secret   Opaque   2      22m
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes>
```

Nous l'instons les services.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl describe ser
vice webapp-service
Name:                     webapp-service
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 app=webapp
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.106.184.88
IPs:                      10.106.184.88
Port:                     <unset>  3000/TCP
TargetPort:               3000/TCP
NodePort:                 <unset>  30100/TCP
Endpoints:                10.244.0.18:3000
Session Affinity:         None
External Traffic Policy:  Cluster
Internal Traffic Policy:  Cluster
Events:                   <none>
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes>
```



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  powershell - kubernetes

PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl describe pod
 webapp-deployment-5766fd95c7
Name:             webapp-deployment-5766fd95c7-9gvsg
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.23.63.113
Start Time:       Thu, 30 Apr 2026 12:37:51 +0200
Labels:           app=webapp
                  pod-template-hash=5766fd95c7
Annotations:      <none>
Status:           Running
IP:               10.244.0.18
IPs:
  IP:           10.244.0.18
Controlled By:  ReplicaSet/webapp-deployment-5766fd95c7
Containers:
  webapp:
    Container ID:   docker://a76c8e05d1c9b5524f2b674663384466a1c16ef4bce304242e76e76d4f858c34
    Image:          nanajanashia/k8s-demo-app:v1.0
    Image ID:       docker-pullable://nanajanashia/k8s-demo-app@sha256:6f554135da39ac00a1c2f4
3e44c2b0b54ca13d3d8044da969361e7781adb7f95
    Port:           3000/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Thu, 30 Apr 2026 12:37:57 +0200
    Ready:          True
    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-dt8dd (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-dt8dd:
    Type:                    Projected (a volume that contains injected data from multiple so
urces)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    DownwardAPI:             true
QoS Class:                   BestEffort
```

Nous consulter les logs des pod.

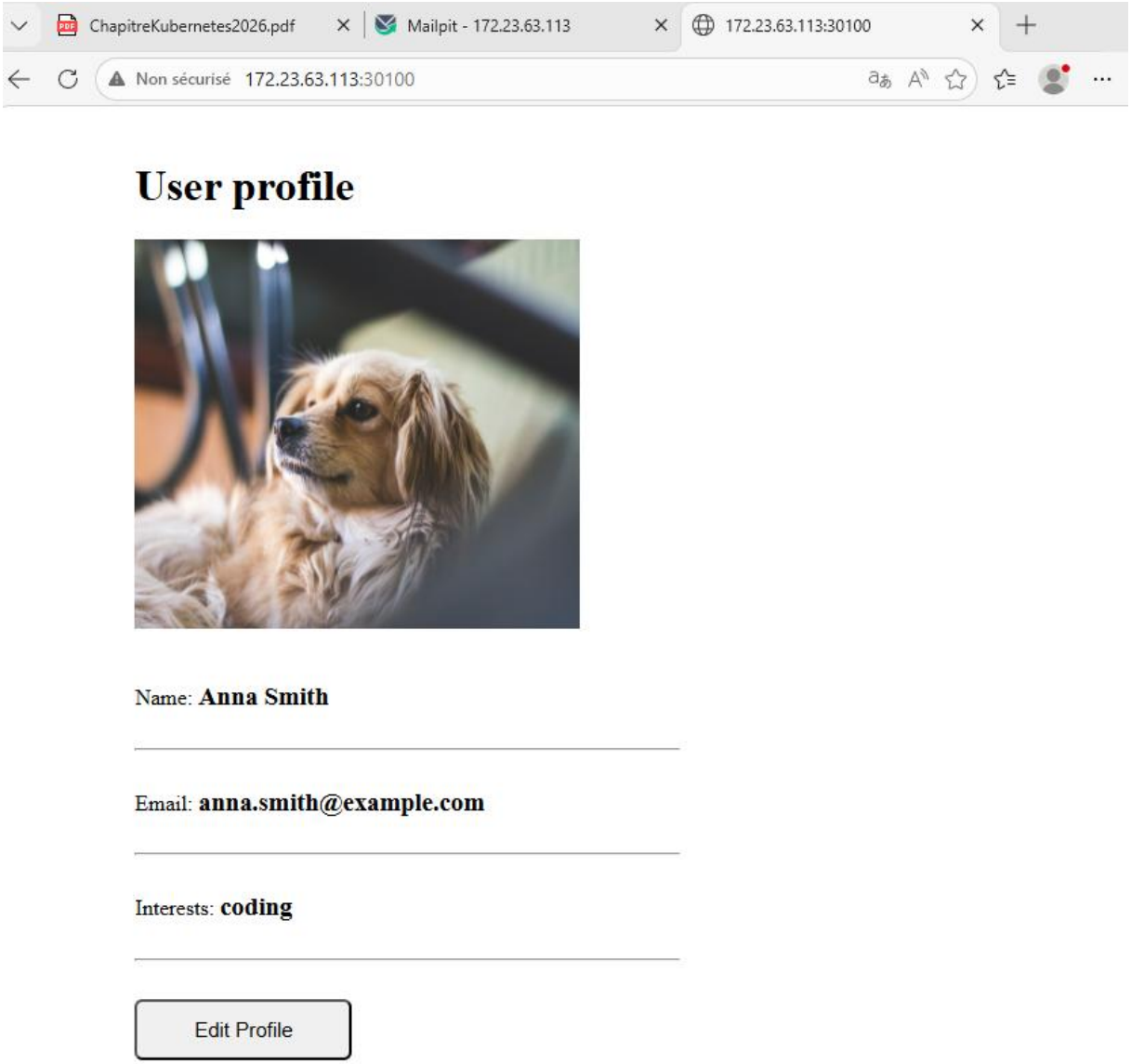
```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-2xj79            1/1     Running   0          105m
mailpit-7b96b48c5d-6rs4g            1/1     Running   0          85m
mongo-deployment-744864fdd7-kbk7c   1/1     Running   0          10m
webapp-deployment-5766fd95c7-9gvsg  1/1     Running   0          8m44s
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl logs  webapp
-deployment-5766fd95c7
error: error from server (NotFound): pods "webapp-deployment-5766fd95c7" not found in namespa
ce "default"
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl logs webapp-
deployment-5766fd95c7-9gvsg
app listening on port 3000!
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl logs webapp-
deployment-5766fd95c7-9gvsg -f
app listening on port 3000!
```

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl get service
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)             AGE
kubernetes      ClusterIP   10.96.0.1        <none>        443/TCP             4h49m
mailpit         ClusterIP   10.99.107.44     <none>        1025/TCP,8025/TCP   3h29m
mongo-service   ClusterIP   10.101.126.187   <none>        27017/TCP           144m
webapp-service  NodePort    10.106.184.88    <none>        3000:30100/TCP      127m
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes>
```

```
PS C:\Users\nnicolau> minikube ip
172.23.63.113
PS C:\Users\nnicolau>
```

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes> kubectl get node
NAME       STATUS   ROLES           AGE    VERSION
minikube   Ready    control-plane   4h53m   v1.35.1
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\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   4h53m   v1.35.1   172.23.63.113   <none>        Buildroot 2025.02   6.6.95           docker://28.5.2
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\kubernetes>
```

Nous essayons d’accéder au site.



Une fois sur le site, nous éditons le profil.

User profile

Name: Maximus the small

Email: maxence.le.magnifique@gm

Interests: Cybertcriminel

Update Profile

User profile

Name: **Maximus the small**

Email: **maxence.le.magnifique@gmail.com**

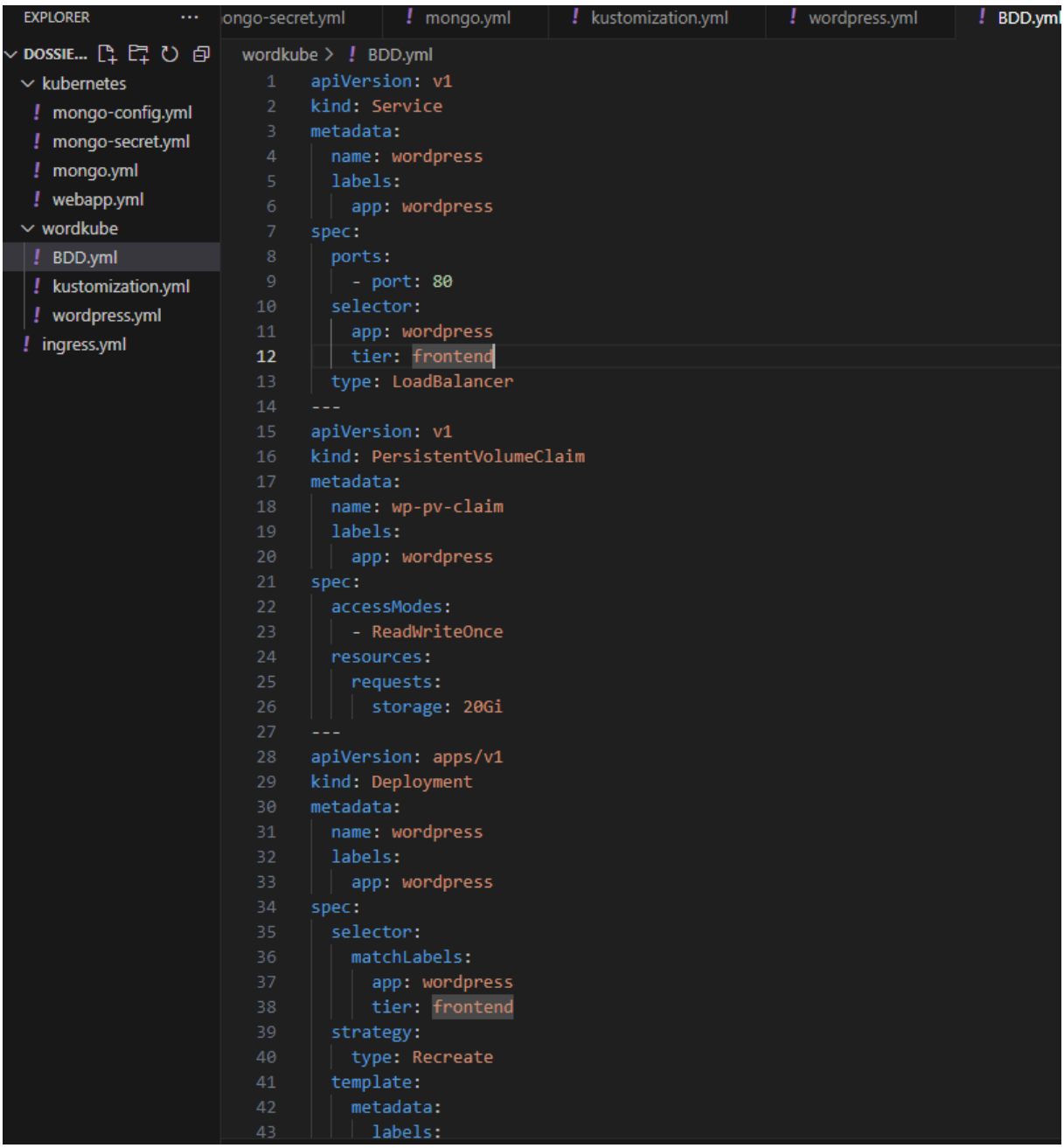
Interests: **Cybertcriminel**

Edit Profile

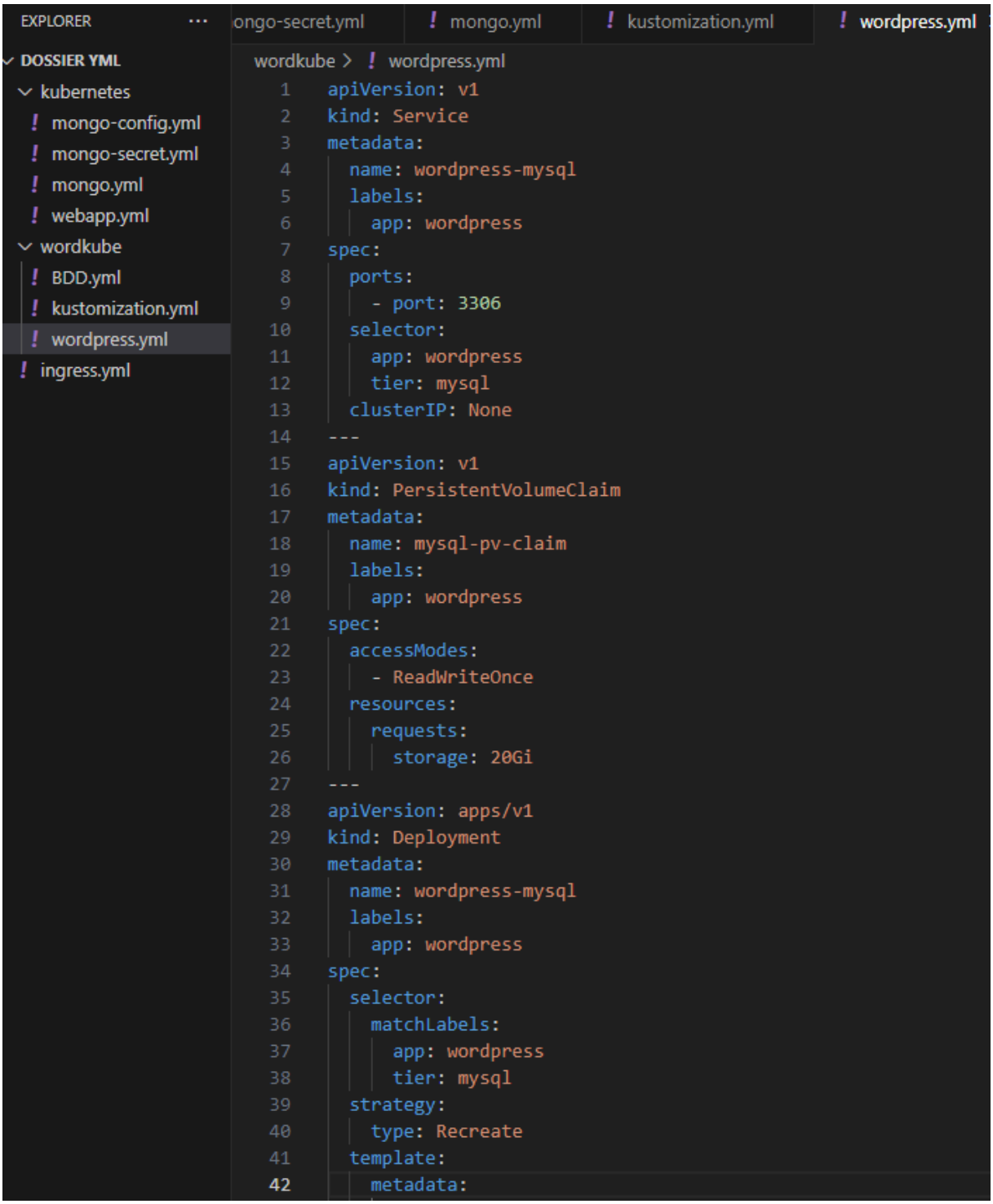
6. Application.

Pour avoir le WordPress qui tourne sur les nœuds, nous commençons par créer les fichiers de configuration pour la BDD et le WordPress.

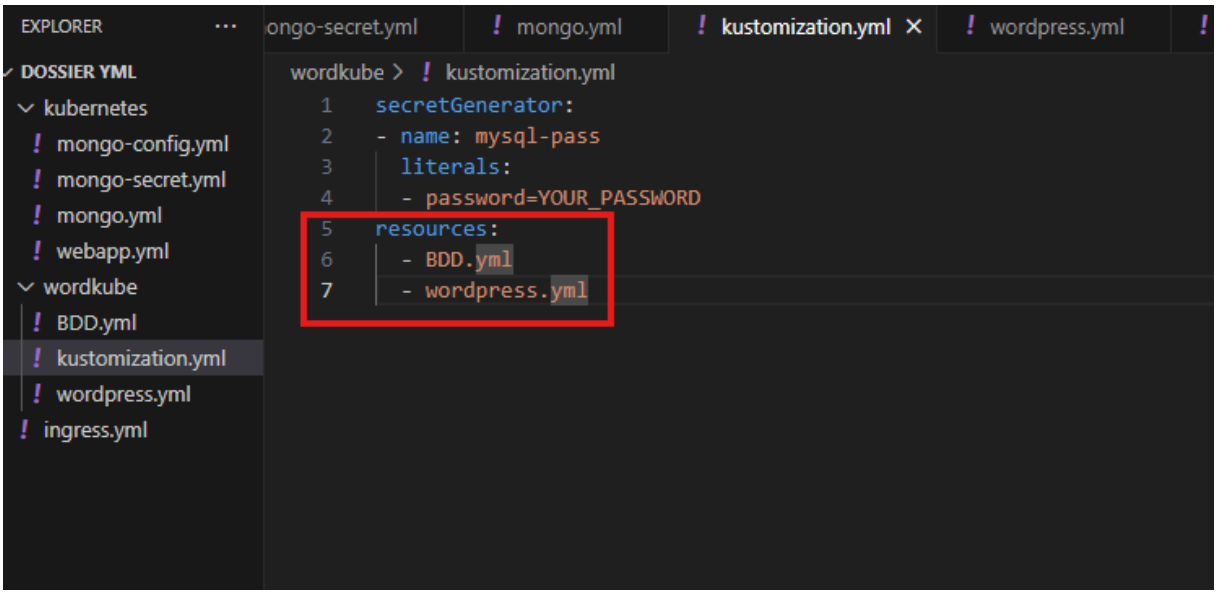
Bdd :



wordpress :


EXPLORER
DOSSIER YML
kubernetes
mongo-config.yml
mongo-secret.yml
mongo.yml
webapp.yml
wordkube
BDD.yml
kustomization.yml
wordpress.yml
ingress.yml
ongo-secret.yml
mongo.yml
kustomization.yml
wordpress.yml
wordkube > wordpress.yml
apiVersion: v1
kind: Service
metadata:
name: wordpress-mysql
labels:
app: wordpress
spec:
ports:
- port: 3306
selector:
app: wordpress
tier: mysql
clusterIP: None
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: mysql-pv-claim
labels:
app: wordpress
spec:
accessModes:
- ReadWriteOnce
resources:
requests:
storage: 20Gi
---
apiVersion: apps/v1
kind: Deployment
metadata:
name: wordpress-mysql
labels:
app: wordpress
spec:
selector:
matchLabels:
app: wordpress
tier: mysql
strategy:
type: Recreate
template:
metadata:

Nous créons un fichier kustomization qui fera appel au fichier créé précédemment.



Nous exécutons le répertoire qui contient les fichiers.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\wordkube> kubectl apply -k ./
secret/mysql-pass-5m26tmdb5k unchanged
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
```

Nous vérifions que le secret a bien été créé.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\wordkube> kubectl get secrets
NAME                  TYPE     DATA   AGE
mongo-secret          Opaque   2      167m
mysql-pass-5m26tmdb5k Opaque   1      57s
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\wordkube>
```

Nous vérifions la persistance des volumes ainsi que des pods.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\wordkube> kubectl get pvc
NAME            STATUS   VOLUME                                     CAPACITY   ACCESS MODES
  STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mysql-pv-claim  Bound    pvc-004e8812-988f-47cb-a6e7-4f333e1c5540   20Gi       RWO
  standard       <unset>                 7s
wp-pv-claim     Bound    pvc-79afd9c0-a5bd-4f10-85ea-21f3c3160a51   20Gi       RWO
  standard       <unset>                 7s
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\wordkube> kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-2xj79            1/1     Running   0          4h16m
mailpit-7b96b48c5d-6rs4g            1/1     Running   0          3h57m
mongo-deployment-744864fdd7-kbk7c   1/1     Running   0          161m
webapp-deployment-5766fd95c7-9gvsg  1/1     Running   0          160m
wordpress-867cdf74df-k8zlx          1/1     Running   0          2m28s
wordpress-mysql-7ddc855c-94kmq      1/1     Running   0          2m28s
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\wordkube>
```

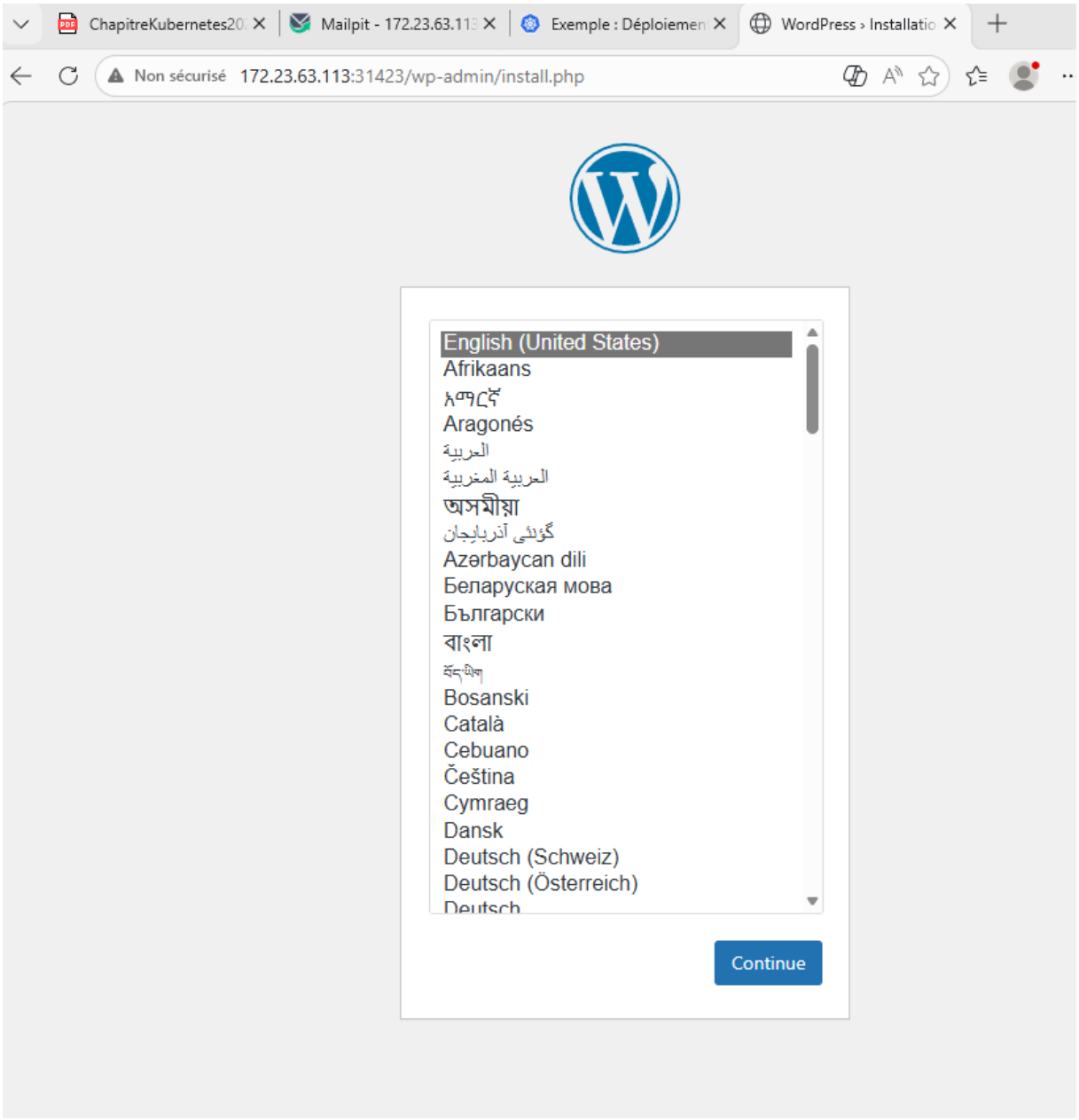
Nous vérifions que le service soit reconnu avec la commande ci-dessous.

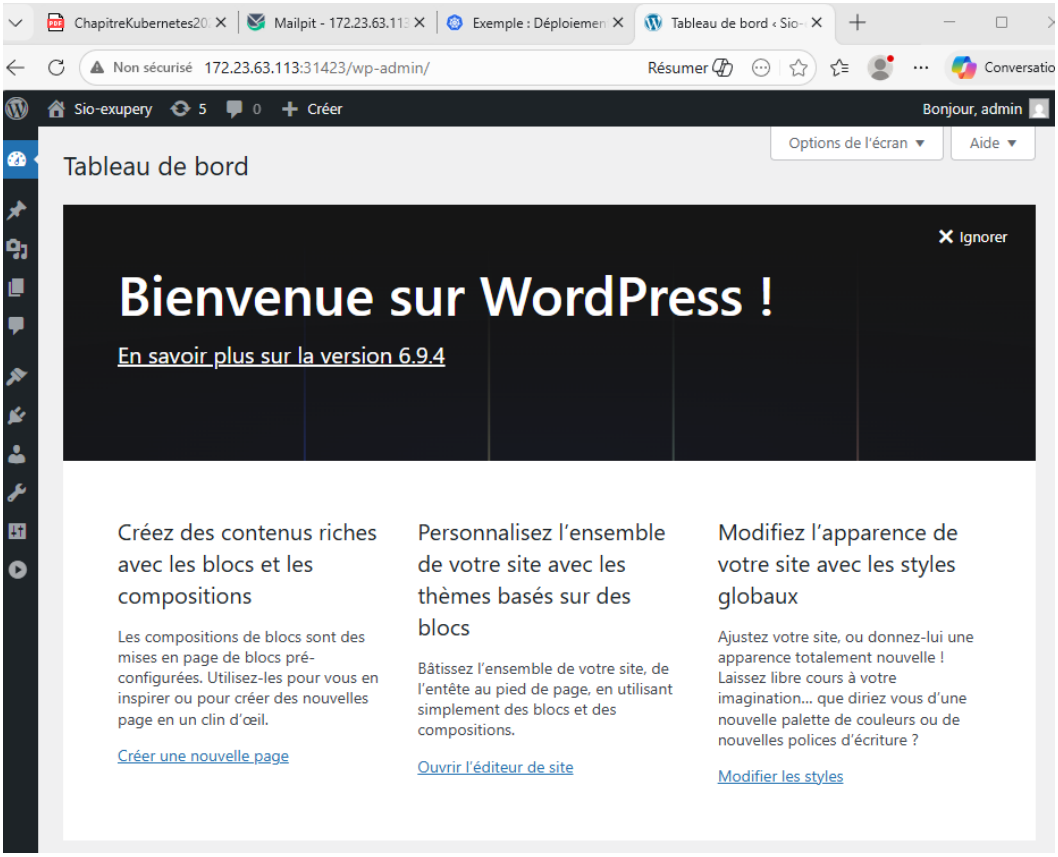
```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\wordkube> kubectl get services w
ordpress
NAME        TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
wordpress   LoadBalancer   10.107.73.189   <pending>     80:31423/TCP   3m9s
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\wordkube>
```

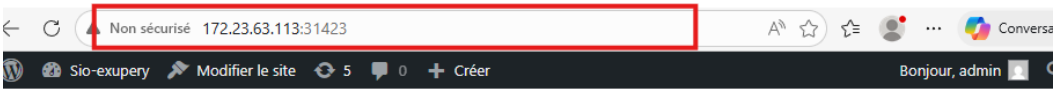
Pour connaître l'URL, nous exécutons la commande ci-dessous.

```
PS C:\Users\nnicolau> minikube service wordpress --url
http://172.23.63.113:31423
PS C:\Users\nnicolau>
```

Nous voilà maintenant sur WordPress.







Sio-exupery Page d'exemple

Mindblown: a blog about philosophy.

Site de noah.

Bonjour tout le monde !

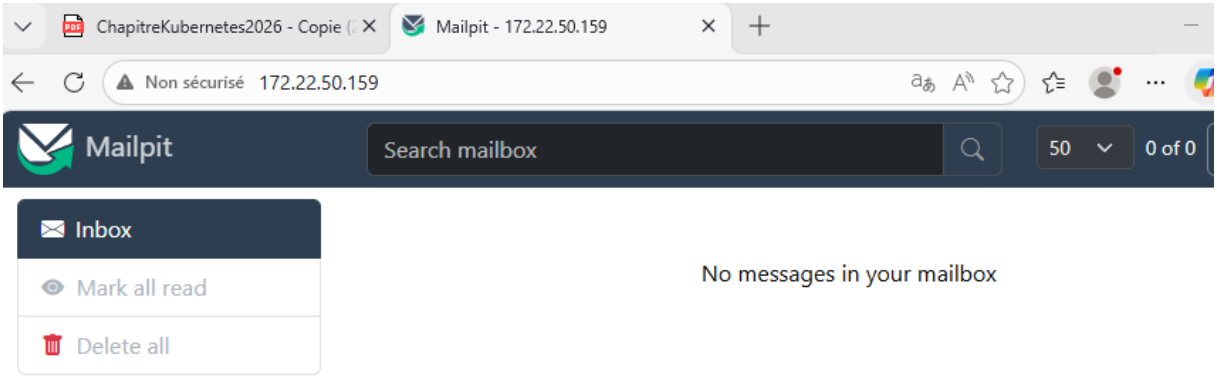
Bienvenue sur WordPress. Ceci est votre premier article. Modifiez-le ou supprimez-le, puis commencez à écrire !

30 avril 2026

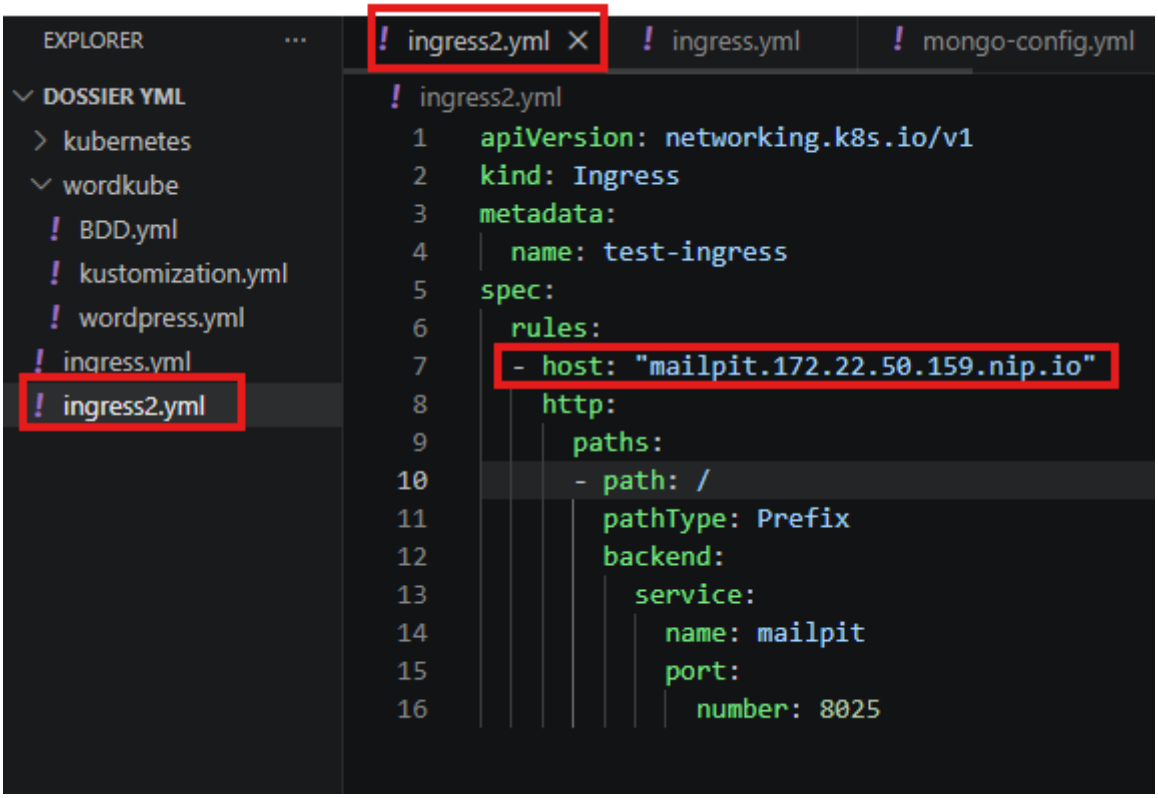
7. Suite Ingress (Mailpit) : hôtes virtuels et nom de domaine nip.io

Nous récupérons les informations sur l'objet ingress (mailpit).

```
PS C:\Users\nnicolau> kubectl get ingress
NAME           CLASS   HOSTS   ADDRESS   PORTS   AGE
test-ingress   nginx   *                 80      6d2h
PS C:\Users\nnicolau> minikube ip
172.22.50.159
PS C:\Users\nnicolau>
```

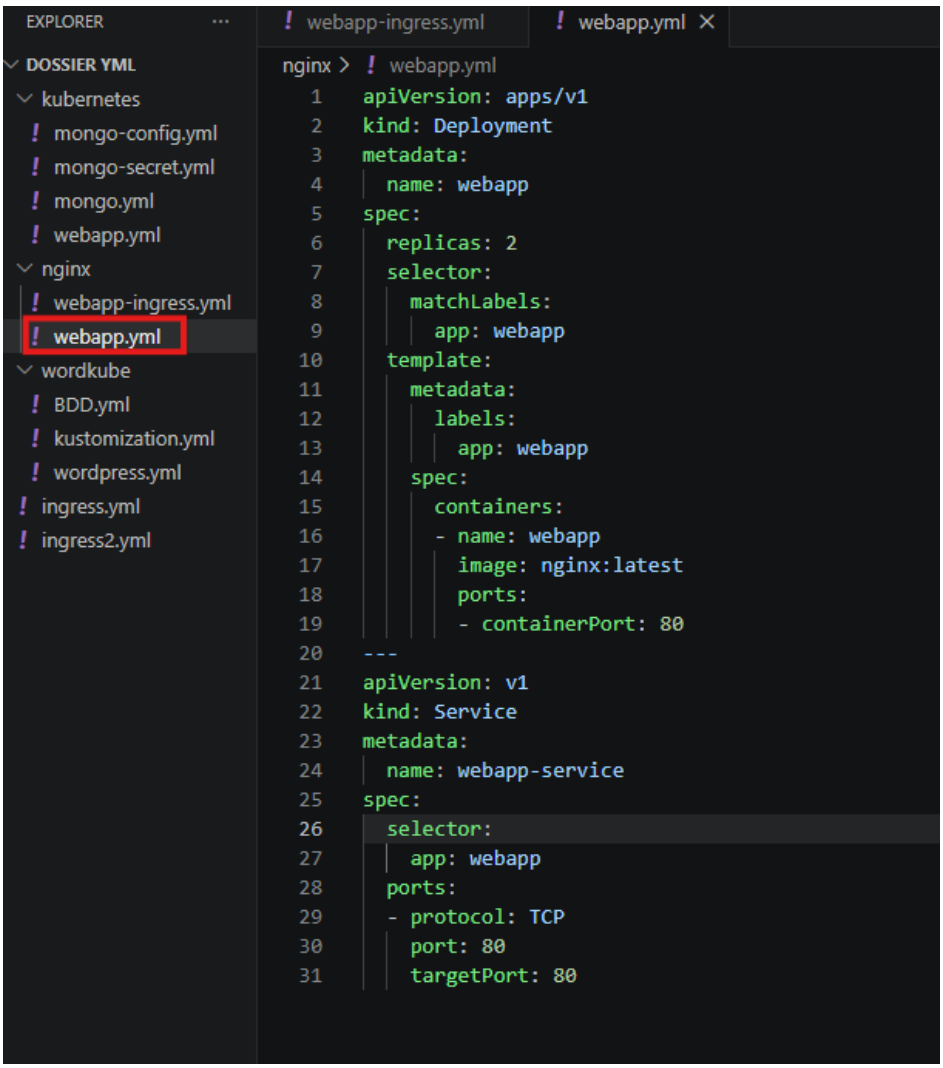


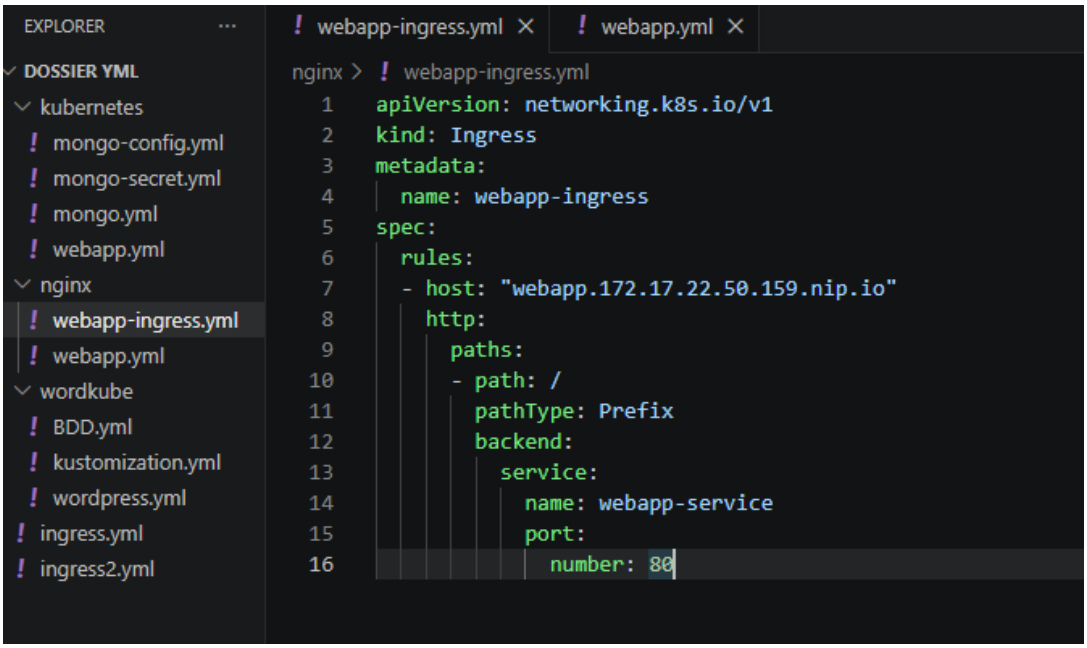
Nous créons un hôte virtuel pour accéder à mailpit.



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

Nous créons un fichier Web avec nginx.





9. Cycle de vie d'un conteneur dans Kubernetes

Nous consultons les pods.

```
PS C:\Users\nnicolau> kubectl get pods -l app=mailpit
NAME                       READY   STATUS    RESTARTS   AGE
mailpit-7b96b48c5d-6j8xs   1/1     Running   0          42m
PS C:\Users\nnicolau>
```

Nous listons les processus et créons un répertoire et le tuons.

```
PS C:\Users\nnicolau> kubectl exec -it deployment/mailpit -- sh
/ # ps -ef
PID   USER     TIME  COMMAND
    1 root      0:00 /mailpit
   14 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 13:46 /tmp/test
/ # kill 1
/ # command terminated with exit code 137
PS C:\Users\nnicolau>
```

Nous consultons l'état des pods.

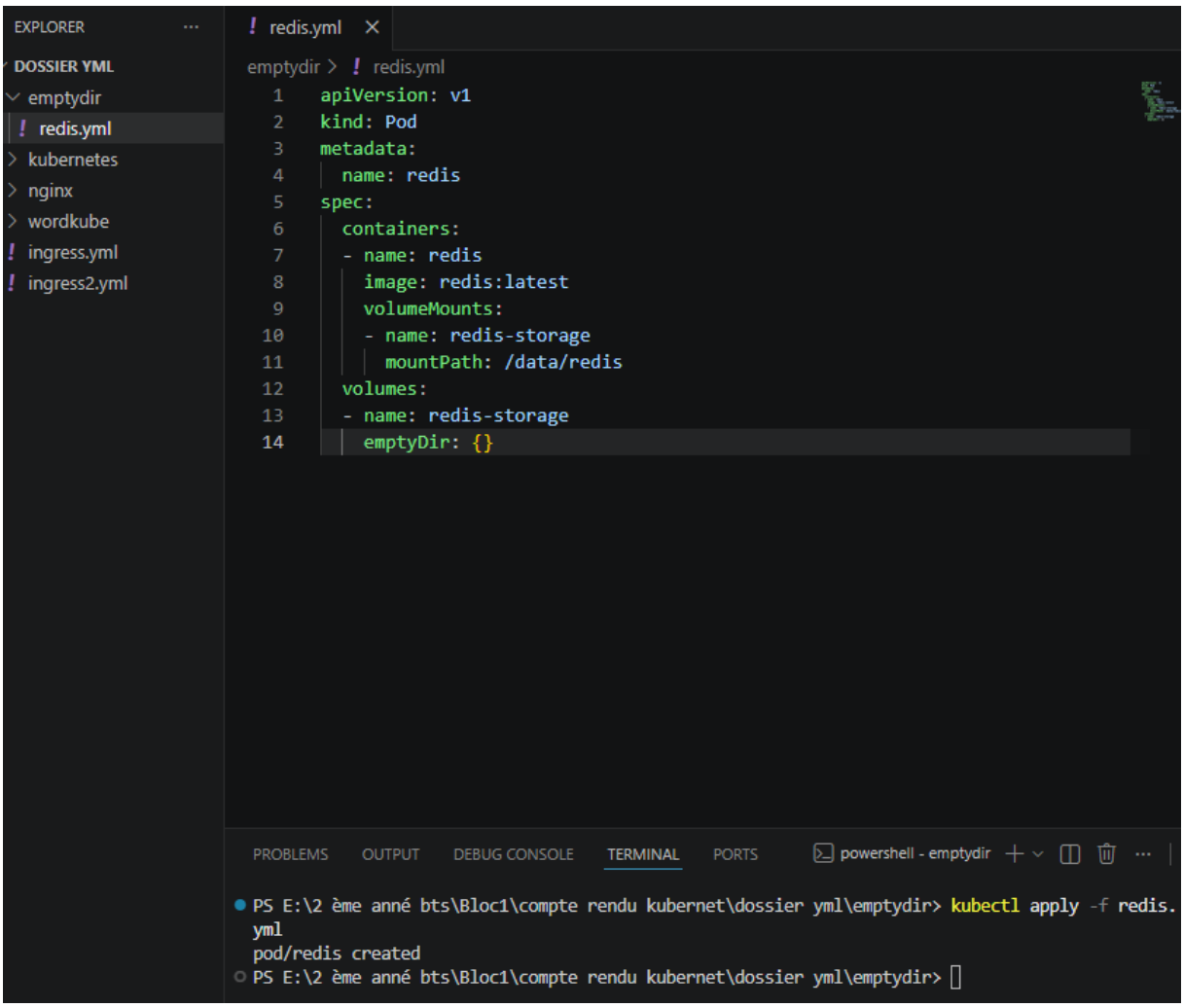
```
PS C:\WINDOWS\system32> kubectl get pods -l app=mailpit
NAME                       READY   STATUS    RESTARTS      AGE
mailpit-7b96b48c5d-6j8xs   1/1     Running   2 (28s ago)   19h
PS C:\WINDOWS\system32>
```

Nous vérifions si le fichier créé précédemment est toujours présent ou non.

```
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>
```

10. Persistance des données

Nous créons le fichier du pod et nous le nommons redis.



Nous vérifions que le pod soit en fonction.

```
PS C:\WINDOWS\system32> kubectl get pods redis --watch
NAME    READY   STATUS    RESTARTS   AGE
redis   1/1     Running   0          2m29s
```

Nous créons un fichier dans le répertoire /data/redis dans un autre terminal.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\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#
```

Nous mettons à jour le pod puis nous installons le procps et listons les processus en cours.

```
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 (9462 kB/s)
Reading package lists... Done
root@redis:/data/redis#
```

```
root@redis:/data/redis# apt-get install procps
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 procps 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
```

```
root@redis:/data/redis# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
redis          1  0.0  0.3 160376 23168 ?        Ssl  08:27   0:00 redis-server *:6379
root          29  0.0  0.0   4320  3584 pts/0    Ss   08:32   0:00 /bin/bash
root          68  0.0  0.0      0     0 pts/0    Z    08:36   0:00 [dpkg-preconfigu] <defunct>
root         157  0.0  0.0   6388  3456 pts/0    R+   08:36   0:00 ps aux
root@redis:/data/redis#
```

Une fois les processus consultés, nous arrêtons le processus. Redis.

```
root@redis:/data/redis# kill 1
root@redis:/data/redis# command terminated with exit code 137
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\emptydir>
```

Nous vérifions les changements apportés au pod.

```
PS C:\WINDOWS\system32> kubectl get pods redis --watch
NAME    READY   STATUS      RESTARTS      AGE
redis   1/1     Running     0             2m29s
redis   0/1     Completed   0             9m46s
redis   1/1     Running     1 (2s ago)    9m48s
```

Nous vérifions que le fichier test-file soit toujours présent dans le répertoire.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\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 E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\emptydir>
```

Une fois vérifié, nous supprimons le pod.

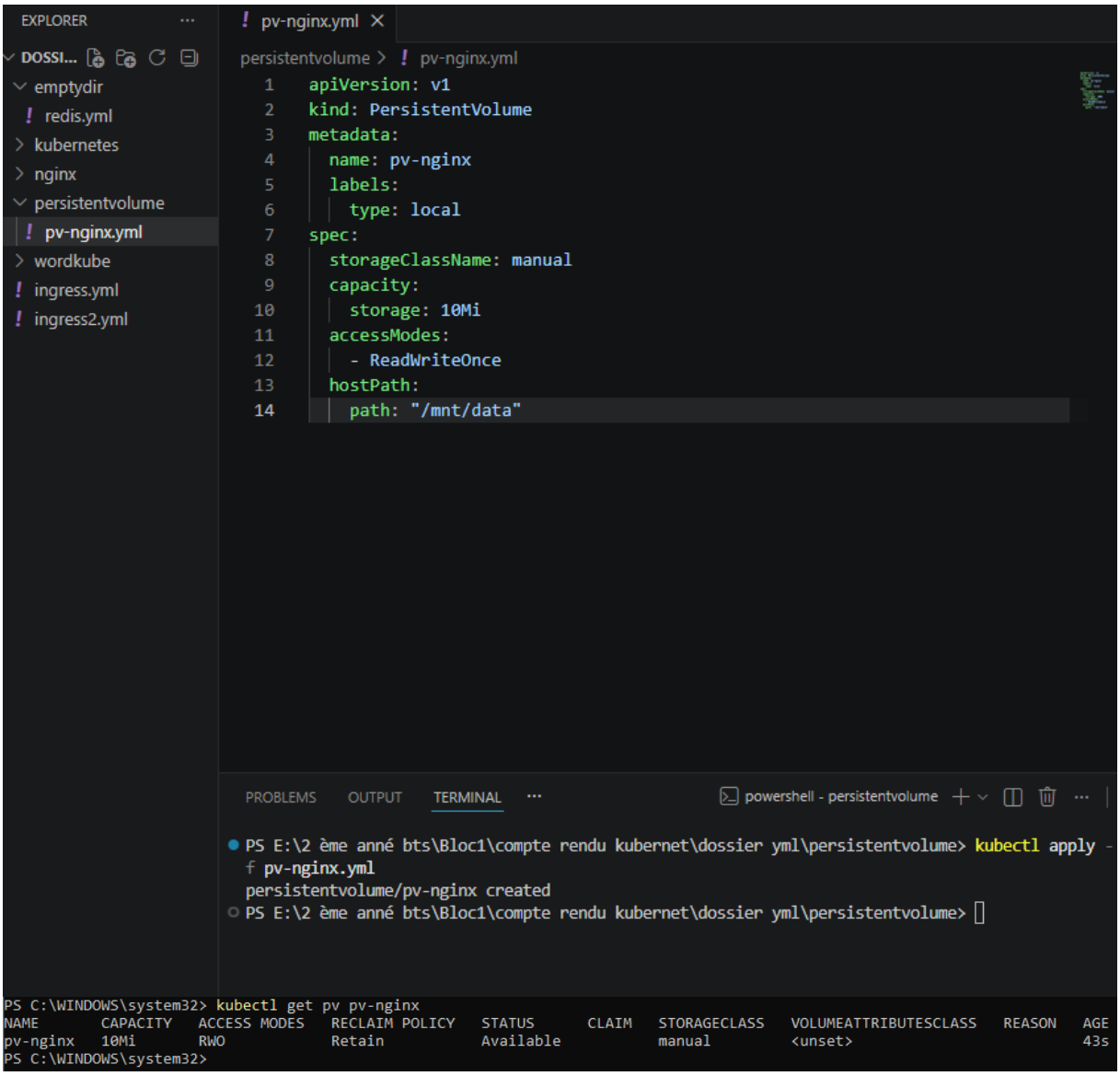
```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\emptydir> kubectl delete pod redi
s
pod "redis" deleted from default namespace
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\emptydir>
```

```
PS C:\WINDOWS\system32> kubectl get pods redis --watch
NAME    READY   STATUS    RESTARTS   AGE
redis   1/1     Running   0          2m29s
redis   0/1     Completed   0          9m46s
redis   1/1     Running     1 (2s ago)   9m48s
redis   1/1     Terminating   1 (6m24s ago)   16m
redis   1/1     Terminating   1 (6m24s ago)   16m
redis   0/1     Completed     1 (6m24s ago)   16m
redis   0/1     Completed     1 (6m25s ago)   16m
redis   0/1     Completed     1 (6m25s ago)   16m
```

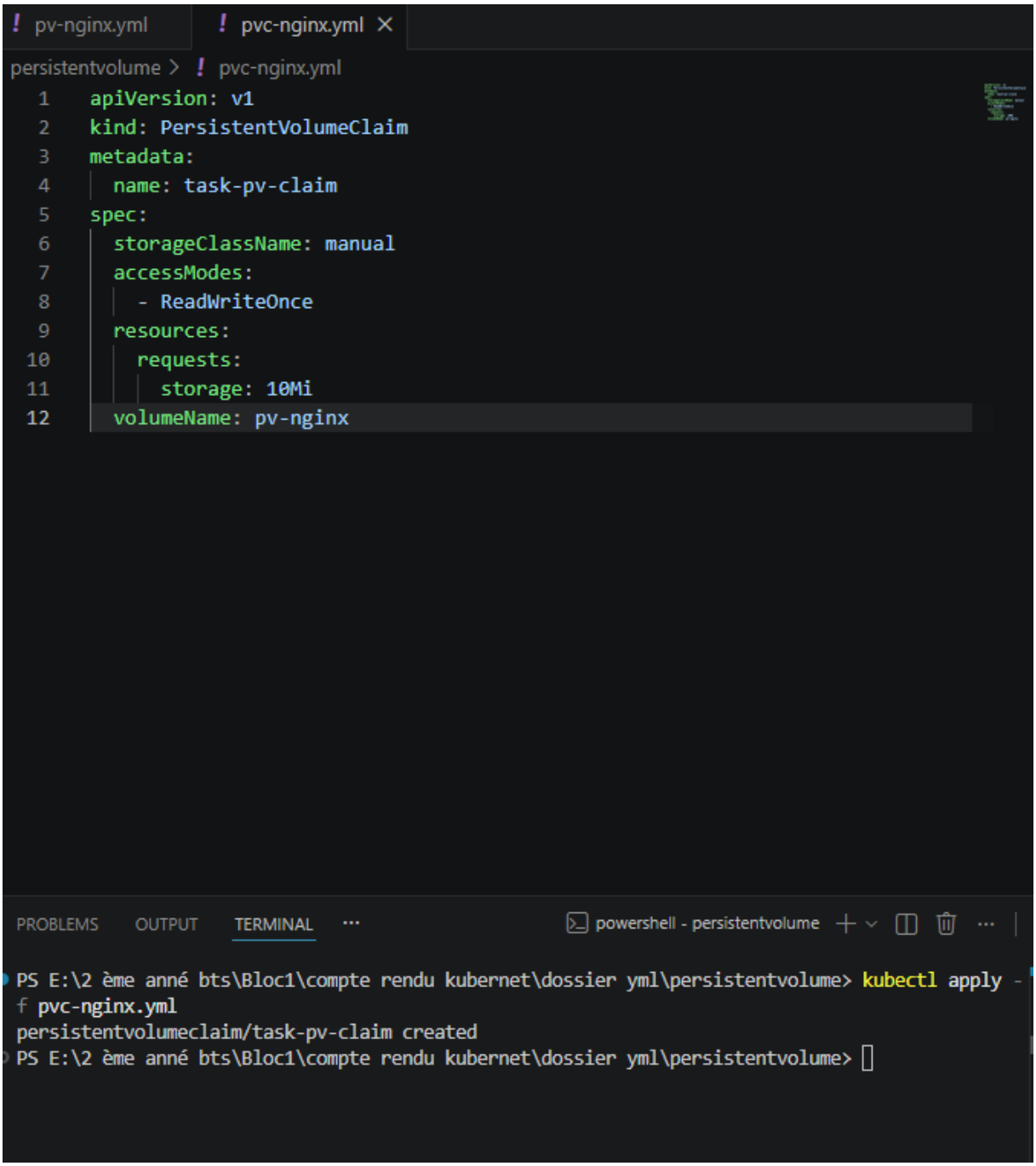
Nous accédons à minikube pour y créer un fichier qui contiendra les données persistantes.

```
PS C:\WINDOWS\system32> minikube ssh
                         _                 _
            _         _ ( )           ( )
  ___ ___  (_)  ___  (_)| |/')  _   _ | |_      __
/' _ ` _ `\| |/' _ `\| || , <  ( ) ( )| '_`\  /'__`\
| ( ) ( ) || || ( ) || || |\`\ | (_) || |_) )(  ___/
(_) (_) (_)(_)(_) (_)(_)(_) (_)`\___/'(_,__/'`\____)

$ sudo mkdir /mnt/data
mkdir: cannot create directory '/mnt/data': File exists
$ sudo sh -c "echo 'PV et PVC' > /mnt/data/index.html"
$ cat /mnt/data/index.html
PV et PVC
$ exit
logout
PS C:\WINDOWS\system32>
```

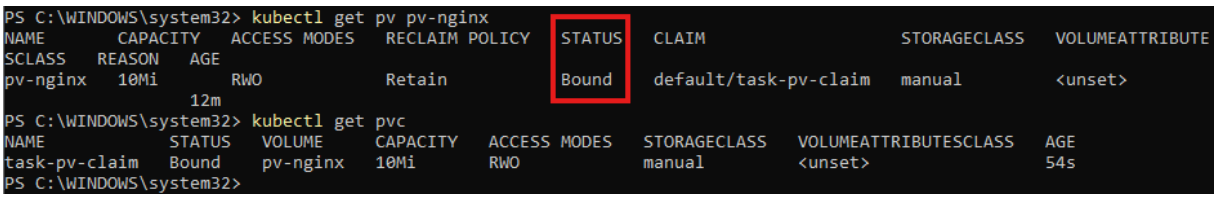
Nous créons un volume persistant du type hospath.



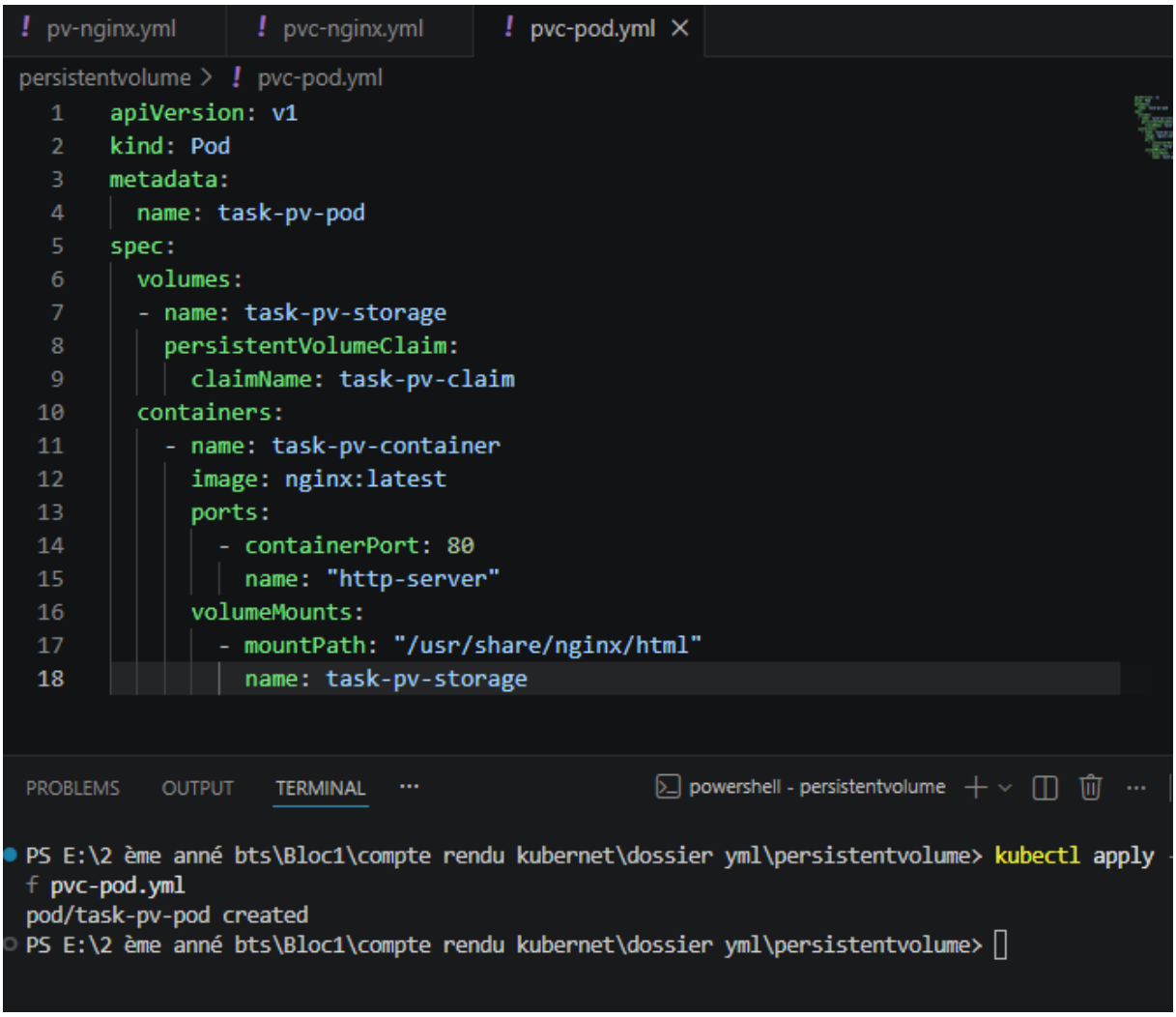
Une fois le PV créé, nous créons un PV Claim qui fera appel au PV.



Nous constatons que le PV est maintenant relié au PVC avec le statut bound.

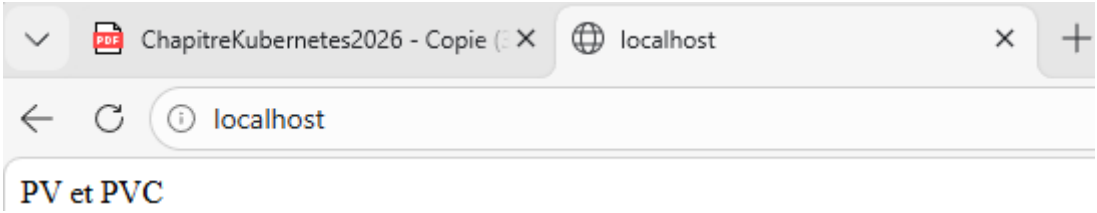


Nous créons un pod qui utilise le PVC comme stockage.




```
PS C:\WINDOWS\system32> kubectl get pod task-pv-pod
NAME          READY   STATUS    RESTARTS   AGE
task-pv-pod   1/1     Running   0          44s
PS C:\WINDOWS\system32> kubectl describe pod task-pv-pod
Name:             task-pv-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.22.50.159
Start Time:       Thu, 07 May 2026 11:16:40 +0200
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.0.34
```

Nous ajoutons le port forward au pod, pour y accéder à la page.



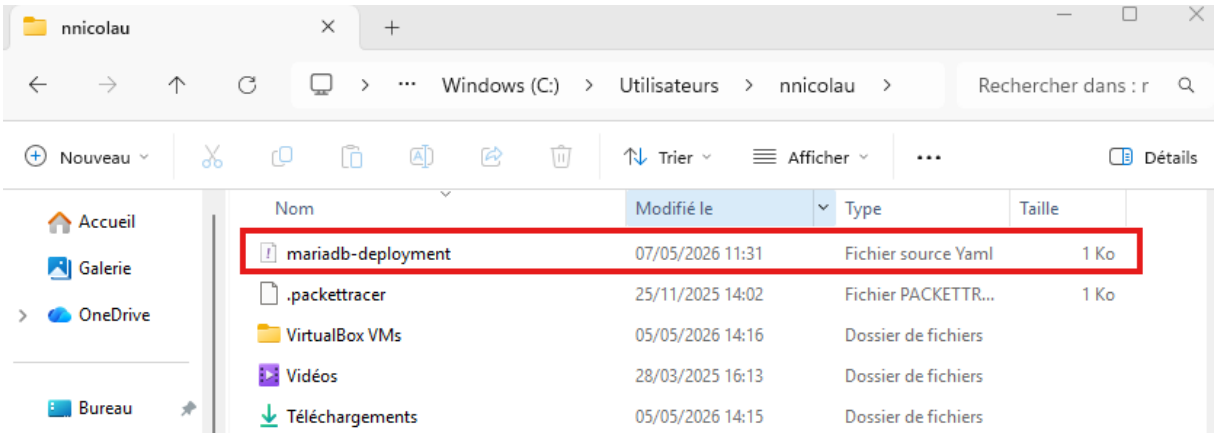
Nous mettons à jour le pod et affichons le contenu du site web.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\persistentvolume> kubectl exec -i
t 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 (9496 kB/s)
Reading package lists... Done
root@task-pv-pod:/# curl http://localhost
PV et PVC
root@task-pv-pod:/#
```

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

Nous créons un fichier de déploiement pour MariaDB.

```
PS C:\WINDOWS\system32> kubectl create deployment mariadb --image=mariadb:latest --dry-run=client --output yaml > mar
iadb-deployment.yml
PS C:\WINDOWS\system32>
```



Nous supprimons le replica, la strategy et les ressources du fichier de MariaDB.

mariadb-deployment.yml MARIADB

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: mariadb
  name: mariadb
spec:
  selector:
    matchLabels:
      app: mariadb
  template:
    metadata:
      labels:
        app: mariadb
    spec:
      containers:
      - image: mariadb:latest
        name: mariadb

```

Nous créons un objet PVC

mariadb-deployment.yml MARIADB | pvc.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mariadb-data
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
```

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl apply -f pvc.yaml
persistentvolumeclaim/mariadb-data created
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

Nous vérifions le status du volume soit en Bound.

```
PS C:\WINDOWS\system32> kubectl get pvc mariadb-data
NAME           STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS
 AGE
mariadb-data   Bound    pvc-77090b61-e8fe-42b1-bf68-006e215e9b72   100Mi      RWO            standard       <unset>
 77s
PS C:\WINDOWS\system32>
```

Nous indiquons au conteneur d'utiliser le volume persistant.

mariadb-deployment.yml MARIADB | pvc.yaml | pvc-ng

MARIADB > mariadb-deployment.yml

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

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl apply -f mari
adb-deployment.yml
deployment.apps/mariadb created
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

Nous constatons que le conteneur est en erreur, nous affichons les logs pour connaître la raison.

```
PS C:\WINDOWS\system32> kubectl get pod -l app=mariadb
NAME                       READY   STATUS    RESTARTS      AGE
mariadb-5c476b5c79-vp9v4   0/1     Error     3 (65s ago)   92s
PS C:\WINDOWS\system32>
```

Nous comprenons que le problème vient du fait qu'il n'y a pas le MDP, ce qui le bloque.

```
PS C:\WINDOWS\system32> kubectl logs -l app=mariadb
2026-05-07 09:51:20+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2+maria~ubu2404 started.
2026-05-07 09:51:21+00:00 [Warn] [Entrypoint]: /sys/fs/cgroup///memory.pressure not writable, functionality unavailab
le to MariaDB
2026-05-07 09:51:21+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2026-05-07 09:51:21+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2+maria~ubu2404 started.
2026-05-07 09:51:21+00:00 [ERROR] [Entrypoint]: Database is uninitialized and password option is not specified
        You need to specify one of MARIADB_ROOT_PASSWORD, MARIADB_ROOT_PASSWORD_HASH, MARIADB_ALLOW_EMPTY_ROOT_PASSWO
RD and MARIADB_RANDOM_ROOT_PASSWORD
PS C:\WINDOWS\system32>
```

Nous ajoutons le mot de passe pour ne plus avoir l'erreur.

mariadb-deployment.yml MARIADB | pvc.yaml | pvc-r

MARIADB > mariadb-deployment.yml

```yaml
3   metadata:
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
26          env:
27              - name: MARIADB_ROOT_PASSWORD
28                value: mot-de-passe-root
29
```

Nous relançons le fichier et constatons que le pod à bien été crée.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl apply -f mari
adb-deployment.yml
deployment.apps/mariadb configured
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

```
PS C:\WINDOWS\system32> kubectl get pod -l app=mariadb
NAME                       READY   STATUS    RESTARTS   AGE
mariadb-5bbb9c9b6d-qzkmq   1/1     Running   0          53s
PS C:\WINDOWS\system32>
```

Nous nous connectons au conteneur pour tester la connectivité avec la bdd.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl exec -it deployment/mariadb -- bash
root@mariadb-5bbb9c9b6d-qzkmq:/# mariadb-admin status -p$MARIADB_ROOT_PASSWORD
Uptime: 161  Threads: 1  Questions: 1  Slow queries: 0  Opens: 17  Open tables: 10  Queries per second avg: 0.006
root@mariadb-5bbb9c9b6d-qzkmq:/# exit
exit
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

Nous créons le service mariadb.

service.yml | mariadb-deployment.yml MARIADB | pvc.yaml | pvc-nginx.yml

MARIADB > service.yml

```yaml
apiVersion: v1
kind: Service
metadata:
  labels:
    app: mariadb
  name: mariadb
spec:
  ports:
  - name: mysql
    port: 3306
    protocol: TCP
    targetPort: 3306
  selector:
    app: mariadb
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... powershell - MARIADB

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl apply -f service.yml
service/mariadb created
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

Nous ajoutons des lignes pour surveiller la base de données.

Malgré le changement, le pod ne parvient toujours pas à se lancer.

```
PS C:\WINDOWS\system32> kubectl get pod -l app=mariadb
NAME                       READY   STATUS    RESTARTS      AGE
mariadb-5bbb9c9b6d-qzkmq   1/1     Running   0             14m
mariadb-6574478f7f-tt2zr   0/1     Running   1 (26s ago)   58s
PS C:\WINDOWS\system32>
```

Nous vérifions pourquoi il n'arrive pas à se lancer avec une description détaillée.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl describe pods
 mariadb-6574478f7f-tt2zr
Name:             mariadb-6574478f7f-tt2zr
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.22.50.159
Start Time:       Thu, 07 May 2026 12:08:24 +0200
Labels:           app=mariadb
                  pod-template-hash=6574478f7f
Annotations:      <none>
Status:           Running
IP:               10.244.0.37
IPs:
  IP:           10.244.0.37
Controlled By:  ReplicaSet/mariadb-6574478f7f
Containers:
  mariadb:
    Container ID:   docker://5a4e5935b2495eeff3aaa20885faeef0c94ebda6667957734dc9e35486d785
6e
    Image:          mariadb:latest
    Image ID:       docker-pullable://mariadb@sha256:e0236fc6386e7eacd9359e59d0a078bd7aa0d1
8280d36d13061121bedeaee903
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 07 May 2026 12:10:32 +0200
    Last State:     Terminated
      Reason:       Error
      Exit Code:    1
      Started:      Thu, 07 May 2026 12:10:01 +0200
      Finished:     Thu, 07 May 2026 12:10:32 +0200
    Ready:          False
    Restart Count:  4
    Liveness:       exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s tim
eout=1s period=10s #success=1 #failure=3
    Readiness:      exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s tim
eout=1s period=10s #success=1 #failure=3
    Startup:        exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s tim
eout=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-bvgx8 (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
amespace)
    ClaimName:  mariadb-data
    ReadOnly:   false
  kube-api-access-bvgx8:
    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
```

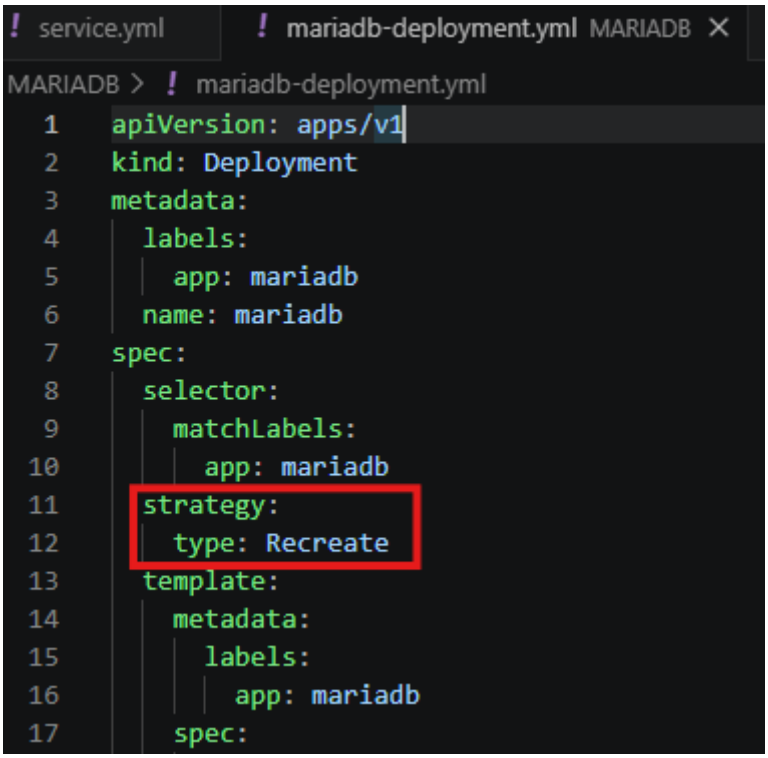
```
Events:
  Type     Reason     Age                From               Message
  ----     ------     ----               ----               -------
  Normal   Scheduled  2m33s              default-scheduler  Successfully assigned default
/mariadb-6574478f7f-tt2zr to minikube
  Normal   Pulled     2m31s              kubelet            Successfully pulled image "ma
riadb:latest" in 904ms (904ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     2m                 kubelet            Successfully pulled image "ma
riadb:latest" in 884ms (884ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     88s                kubelet            Successfully pulled image "ma
riadb:latest" in 943ms (943ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     56s                kubelet            Successfully pulled image "ma
riadb:latest" in 918ms (918ms including waiting). Image size: 335568219 bytes.
  Normal   Pulling    25s (x5 over 2m32s)  kubelet          Pulling image "mariadb:latest
"
  Normal   Created    25s (x5 over 2m31s)  kubelet          Container created
  Normal   Pulled     25s                kubelet            Successfully pulled image "ma
riadb:latest" in 877ms (877ms including waiting). Image size: 335568219 bytes.
  Normal   Started    24s (x5 over 2m31s)  kubelet          Container started
  Warning  Unhealthy  2s (x15 over 2m22s)  kubelet          Startup probe failed: mariadb
-admin: connect to server at 'localhost' failed
error: 'Can't connect to local server through socket '/run/mysqld/mysqld.sock' (2)'
Check that mariadbd is running and that the socket: '/run/mysqld/mysqld.sock' exists!
  Normal   Killing  2s (x5 over 2m2s)  kubelet  Container mariadb failed startup probe, will
 be restarted
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

Nous constatons qu'il c'est mis en crash.

```
PS C:\WINDOWS\system32> kubectl get pod -l app=mariadb
NAME                       READY   STATUS             RESTARTS      AGE
mariadb-5bbb9c9b6d-qzkmq   1/1     Running            0             17m
mariadb-6574478f7f-tt2zr   0/1     CrashLoopBackOff   6 (6s ago)    4m40s
PS C:\WINDOWS\system32>
```

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl logs  mariadb
-6574478f7f-tt2zr
2026-05-07 10:12:27+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.
2+maria~ubu2404 started.
2026-05-07 10:12:27+00:00 [Warn] [Entrypoint]: /sys/fs/cgroup///memory.pressure not writabl
e, functionality unavailable to MariaDB
2026-05-07 10:12:27+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2026-05-07 10:12:27+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.
2+maria~ubu2404 started.
2026-05-07 10:12:27+00:00 [Note] [Entrypoint]: MariaDB upgrade not required
2026-05-07 10:12:27 0 [Note] Starting MariaDB 12.2.2-MariaDB-ubu2404 source revision d26a6f
44c1f2119377e79a9540886c6d8c01472f server_uid PsVF+GLIMvZCv5XYpZuYwLo8A8k= as process 1
2026-05-07 10:12:27 0 [ERROR] mariadbd: Can't lock aria control file '/var/lib/mysql/aria_l
og_control' for exclusive use, error: 11. Will retry for 30 seconds
2026-05-07 10:12:57 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/a
ria_log_control'
2026-05-07 10:12:57 0 [ERROR] Plugin 'Aria' registration as a STORAGE ENGINE failed.
2026-05-07 10:12:57 0 [Note] InnoDB: Compressed tables use zlib 1.3
2026-05-07 10:12:57 0 [Note] InnoDB: Number of transaction pools: 1
2026-05-07 10:12:57 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
2026-05-07 10:12:57 0 [Note] mariadbd: O_TMPFILE is not supported on /tmp (disabling future
 attempts)
2026-05-07 10:12:57 0 [Note] InnoDB: Using io_uring
2026-05-07 10:12:57 0 [Note] InnoDB: innodb_buffer_pool_size_max=128m, innodb_buffer_pool_s
ize=128m
2026-05-07 10:12:57 0 [Note] InnoDB: Completed initialization of buffer pool
2026-05-07 10:12:57 0 [Note] InnoDB: File system buffers for log disabled (block size=512 b
ytes)
2026-05-07 10:12:57 0 [ERROR] InnoDB: Unable to lock ./ibdata1 error: 11
2026-05-07 10:12:57 0 [Note] InnoDB: Check that you do not already have another mariadbd pr
ocess using the same InnoDB data or log files.
2026-05-07 10:12:57 0 [ERROR] InnoDB: Plugin initialization aborted with error Generic erro
r
2026-05-07 10:12:57 0 [Note] InnoDB: Starting shutdown...
2026-05-07 10:12:58 0 [ERROR] Plugin 'InnoDB' registration as a STORAGE ENGINE failed.
2026-05-07 10:12:58 0 [Note] Plugin 'FEEDBACK' is disabled.
2026-05-07 10:12:58 0 [Note] Plugin 'wsrep-provider' is disabled.
2026-05-07 10:12:58 0 [ERROR] Could not open mysql.plugin table: "Unknown storage engine 'A
ria'". Some plugins may be not loaded
2026-05-07 10:12:58 0 [ERROR] Failed to initialize plugins.
2026-05-07 10:12:58 0 [ERROR] Aborting
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

En regardant la description, nous constatons que le problème vient du fait que la machine ne peut pas accéder à la bdd car l'ancien pod est toujours connecté dessus. Avec cette ligne, l'ancien pod se supprime à la création du nouveau.



Nous relançons le script.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl apply -f mari
adb-deployment.yml
deployment.apps/mariadb configured
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

Le script a bien supprimé l'ancien pod, mais le nouveau est toujours en échec, nous le supprimons et le relançons pour qu'il soit opérationnel.

```
PS C:\WINDOWS\system32> kubectl get pod -l app=mariadb
NAME                       READY   STATUS             RESTARTS        AGE
mariadb-6574478f7f-tt2zr   0/1     CrashLoopBackOff   7 (2m29s ago)   10m
PS C:\WINDOWS\system32>
```

```
PS C:\WINDOWS\system32> kubectl delete pods mariadb-6574478f7f-tt2zr
pod "mariadb-6574478f7f-tt2zr" deleted from default namespace
PS C:\WINDOWS\system32> kubectl get pod -l app=mariadb
NAME                       READY   STATUS    RESTARTS   AGE
mariadb-6574478f7f-9j5fv   0/1     Running   0          6s
PS C:\WINDOWS\system32>
```

```
PS C:\WINDOWS\system32> kubectl get pod -l app=mariadb
NAME                       READY   STATUS             RESTARTS        AGE
mariadb-6574478f7f-tt2zr   0/1     CrashLoopBackOff   7 (2m29s ago)   10m
PS C:\WINDOWS\system32> kubectl delete pods mariadb-6574478f7f-tt2zr
pod "mariadb-6574478f7f-tt2zr" deleted from default namespace
PS C:\WINDOWS\system32> kubectl get pod -l app=mariadb
NAME                       READY   STATUS    RESTARTS   AGE
mariadb-6574478f7f-9j5fv   0/1     Running   0          6s
PS C:\WINDOWS\system32>
```

Nous augmentons le nombre de pods associés au déploiement.

```
PS C:\WINDOWS\system32> kubectl get pod -l app=mariadb --watch
NAME                       READY   STATUS    RESTARTS      AGE
mariadb-6574478f7f-9j5fv   1/1     Running   0             2m59s
mariadb-6574478f7f-htxnj   0/1     Running   0             28s
mariadb-6574478f7f-htxnj   0/1     Running   1 (1s ago)    33s
```

Nous constatons qu'il n'arrive toujours pas à démarrer.

```
PS C:\Users\nnicolau> kubectl describe pods mariadb-6574478f7f-htxnj
Name:             mariadb-6574478f7f-htxnj
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.22.50.159
Start Time:       Thu, 07 May 2026 12:23:03 +0200
Labels:           app=mariadb
                  pod-template-hash=6574478f7f
Annotations:      <none>
Status:           Running
IP:               10.244.0.39
IPs:
  IP:           10.244.0.39
Controlled By:  ReplicaSet/mariadb-6574478f7f
Containers:
  mariadb:
    Container ID:   docker://420ba740d050dc549183a36494df0f0630a61e136cb511aa07861474553d202f
    Image:          mariadb:latest
    Image ID:       docker-pullable://mariadb@sha256:e0236fc6386e7eacd9359e59d0a078bd7aa0d18280d36d130
61121bedeaee903
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 07 May 2026 12:25:11 +0200
    Last State:     Terminated
      Reason:       Error
      Exit Code:    1
      Started:      Thu, 07 May 2026 12:24:39 +0200
      Finished:     Thu, 07 May 2026 12:25:10 +0200
    Ready:          False
    Restart Count:  4
    Liveness:       exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s timeout=1s per
iod=10s #success=1 #failure=3
    Readiness:      exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s timeout=1s per
iod=10s #success=1 #failure=3
    Startup:        exec [sh -c mariadb-admin status -p$MARIADB_ROOT_PASSWORD] delay=0s timeout=1s per
iod=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-5pm5z (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-5pm5z:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason     Age                From               Message
  ----     ------     ----               ----               -------
  Normal   Scheduled  2m10s              default-scheduler  Successfully assigned default/mariadb-6574
478f7f-htxnj to minikube
  Normal   Pulled     2m9s               kubelet            Successfully pulled image "mariadb:latest"
 in 899ms (899ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     97s                kubelet            Successfully pulled image "mariadb:latest"
 in 808ms (808ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     66s                kubelet            Successfully pulled image "mariadb:latest"
 in 901ms (901ms including waiting). Image size: 335568219 bytes.
  Normal   Pulled     34s                kubelet            Successfully pulled image "mariadb:latest"
 in 846ms (846ms including waiting). Image size: 335568219 bytes.
  Warning  Unhealthy  10s (x12 over 2m)  kubelet            Startup probe failed: mariadb-admin: conne
ct to server at 'localhost' failed
error: 'Can't connect to local server through socket '/run/mysqld/mysqld.sock' (2)'
Check that mariadbd is running and that the socket: '/run/mysqld/mysqld.sock' exists!
  Normal  Killing  10s (x4 over 100s)  kubelet  Container mariadb failed startup probe, will be restar
ted
  Normal  Pulling  3s (x5 over 2m10s)  kubelet  Pulling image "mariadb:latest"
  Normal  Created  2s (x5 over 2m9s)   kubelet  Container created
  Normal  Started  2s (x5 over 2m9s)   kubelet  Container started
  Normal  Pulled   2s                  kubelet  Successfully pulled image "mariadb:latest" in 861ms (8
61ms including waiting). Image size: 335568219 bytes.
PS C:\Users\nnicolau>
```

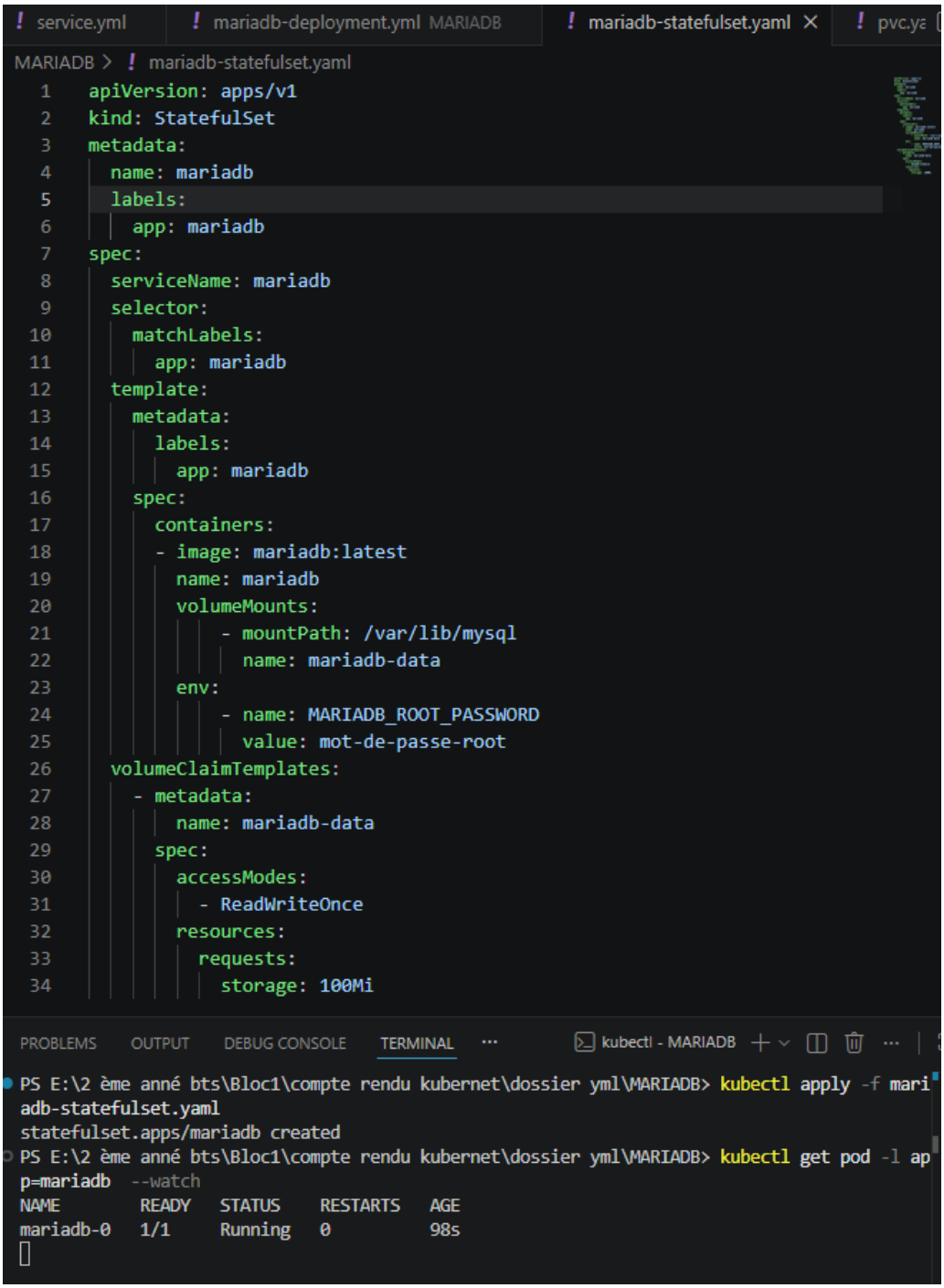
```
PS C:\WINDOWS\system32> kubectl get pod -l app=mariadb
NAME                       READY   STATUS             RESTARTS      AGE
mariadb-6574478f7f-9j5fv   1/1     Running            0             7m29s
mariadb-6574478f7f-htxnj   0/1     CrashLoopBackOff   5 (62s ago)   4m58s
PS C:\WINDOWS\system32> kubectl logs mariadb-6574478f7f-htxnj
2026-05-07 10:28:20+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2+maria~ubu2404 started.
2026-05-07 10:28:21+00:00 [Warn] [Entrypoint]: /sys/fs/cgroup///memory.pressure not writable, functionality unavailab
le to MariaDB
2026-05-07 10:28:21+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2026-05-07 10:28:21+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:12.2.2+maria~ubu2404 started.
2026-05-07 10:28:21+00:00 [Note] [Entrypoint]: MariaDB upgrade not required
2026-05-07 10:28:21 0 [Note] Starting MariaDB 12.2.2-MariaDB-ubu2404 source revision d26a6f44c1f2119377e79a9540886c6d
8c01472f server_uid FgUYoaS33r28j0dWJF+BkrZ6NUI= as process 1
2026-05-07 10:28:21 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 10:28:51 0 [ERROR] mariadbd: Got error 'Could not get an exclusive lock; file is probably in use by anothe
r process' when trying to use aria control file '/var/lib/mysql/aria_log_control'
2026-05-07 10:28:51 0 [ERROR] Plugin 'Aria' registration as a STORAGE ENGINE failed.
2026-05-07 10:28:51 0 [Note] InnoDB: Compressed tables use zlib 1.3
2026-05-07 10:28:51 0 [Note] InnoDB: Number of transaction pools: 1
2026-05-07 10:28:51 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
2026-05-07 10:28:51 0 [Note] mariadbd: O_TMPFILE is not supported on /tmp (disabling future attempts)
2026-05-07 10:28:51 0 [Note] InnoDB: Using io_uring
2026-05-07 10:28:51 0 [Note] InnoDB: innodb_buffer_pool_size_max=128m, innodb_buffer_pool_size=128m
2026-05-07 10:28:51 0 [Note] InnoDB: Completed initialization of buffer pool
2026-05-07 10:28:51 0 [Note] InnoDB: File system buffers for log disabled (block size=512 bytes)
2026-05-07 10:28:51 0 [ERROR] InnoDB: Unable to lock ./ibdata1 error: 11
2026-05-07 10:28:51 0 [Note] InnoDB: Check that you do not already have another mariadbd process using the same InnoD
B data or log files.
2026-05-07 10:28:51 0 [ERROR] InnoDB: Plugin initialization aborted with error Generic error
2026-05-07 10:28:51 0 [Note] InnoDB: Starting shutdown...
2026-05-07 10:28:51 0 [ERROR] Plugin 'InnoDB' registration as a STORAGE ENGINE failed.
2026-05-07 10:28:51 0 [Note] Plugin 'FEEDBACK' is disabled.
2026-05-07 10:28:51 0 [Note] Plugin 'wsrep-provider' is disabled.
2026-05-07 10:28:51 0 [ERROR] Could not open mysql.plugin table: "Unknown storage engine 'Aria'". Some plugins may be
 not loaded
2026-05-07 10:28:51 0 [ERROR] Failed to initialize plugins.
2026-05-07 10:28:51 0 [ERROR] Aborting
PS C:\WINDOWS\system32>
```

Le problème est que les deux pods essaient de se connecter à la même BDD, ce qui pose problème. Pour remédier à ça, nous avons dissocié leur BDD.

Nous commençons par supprimer les deux.

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

Nous créons un fichier mariadb statefulset.



Nous consultons l'état des volumes et par la suite nous consultons les volumes.

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl get pvc
NAME                    STATUS   VOLUME                                     CAPACITY   ACC
ESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mariadb-data            Bound    pvc-77090b61-e8fe-42b1-bf68-006e215e9b72   100Mi      RWO
            standard       <unset>                 60m
mariadb-data-mariadb-0  Bound    pvc-59832046-9bc9-4a21-9edb-a31dd38fae39   100Mi      RWO
            standard       <unset>                 3m1s
task-pv-claim           Bound    pv-nginx                                   10Mi       RWO
            manual         <unset>                 93m
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATU
S   CLAIM                              STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
pv-nginx                                   10Mi       RWO            Retain           Bound
    default/task-pv-claim              manual         <unset>                         106m
pvc-59832046-9bc9-4a21-9edb-a31dd38fae39   100Mi      RWO            Delete           Bound
    default/mariadb-data-mariadb-0     standard       <unset>                         3m48s
pvc-77090b61-e8fe-42b1-bf68-006e215e9b72   100Mi      RWO            Delete           Bound
    default/mariadb-data               standard       <unset>                         61m
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

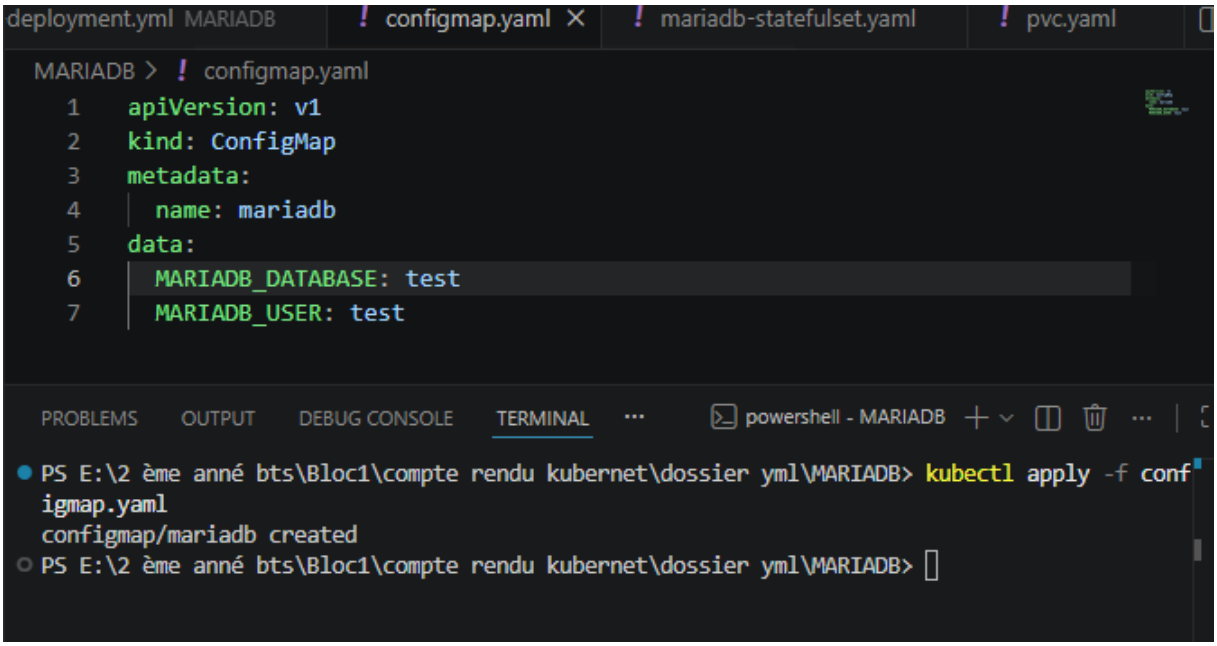
Nous supprimons le volume et constatons qu'il a gardé en mémoire les changements.

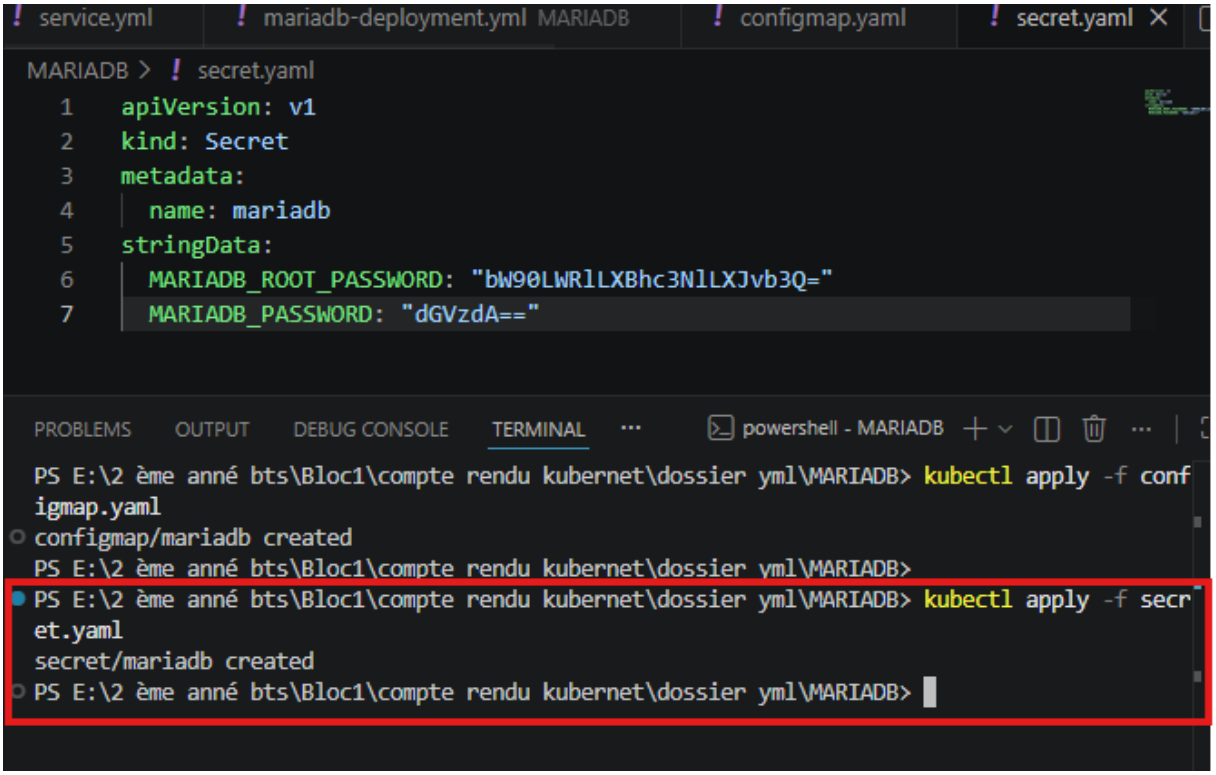
```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl delete pvc/ma
riadb-data
persistentvolumeclaim "mariadb-data" deleted from default namespace
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl scale sts mar
iadb --replicas=2
statefulset.apps/mariadb scaled
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

```
PS C:\WINDOWS\system32> kubectl get pvc -l app=mariadb
NAME                    STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
 VOLUMEATTRIBUTESCLASS   AGE
mariadb-data-mariadb-0  Bound    pvc-59832046-9bc9-4a21-9edb-a31dd38fae39   100Mi      RWO            standard
 <unset>                 6m21s
mariadb-data-mariadb-1  Bound    pvc-5044cb2f-f01f-4e6a-9ea0-a1af8aa128db   100Mi      RWO            standard
 <unset>                 43s
PS C:\WINDOWS\system32> kubectl get pods -l app=mariadb
NAME        READY   STATUS    RESTARTS   AGE
mariadb-0   1/1     Running   0          7m11s
mariadb-1   1/1     Running   0          93s
PS C:\WINDOWS\system32> kubectl get pods,pvc -l app=mariadb
NAME            READY   STATUS    RESTARTS   AGE
pod/mariadb-0   1/1     Running   0          7m25s
pod/mariadb-1   1/1     Running   0          107s

NAME                                        STATUS   VOLUME                                     CAPACITY   ACCESS
MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/mariadb-data-mariadb-0   Bound    pvc-59832046-9bc9-4a21-9edb-a31dd38fae39   100Mi      RWO
        standard       <unset>                 7m25s
persistentvolumeclaim/mariadb-data-mariadb-1   Bound    pvc-5044cb2f-f01f-4e6a-9ea0-a1af8aa128db   100Mi      RWO
        standard       <unset>                 107s
PS C:\WINDOWS\system32>
```

Nous créons une base pour vérifier s'ils sont bien stockés.





Nous le rattachons au conteneur.

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mariadb
  labels:
    app: mariadb
spec:
  serviceName: mariadb
  selector:
    matchLabels:
      app: mariadb
  template:
    metadata:
      labels:
        app: mariadb
    spec:
      containers:
      - image: mariadb:latest
        name: mariadb
        volumeMounts:
          - mountPath: /var/lib/mysql
            name: mariadb-data
        envFrom:
          - configMapRef:
              name: mariadb
          - secretRef:
              name: mariadb
        startupProbe: &probe
          exec:
            command:
              - "sh"
              - "-c"
              - "mariadb-admin status -p$MARIADB_ROOT_PASSWORD"
        livenessProbe: *probe
        readinessProbe: *probe
  volumeClaimTemplates:
    - metadata:
        name: mariadb-data
      spec:
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 100Mi
```

```
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB> kubectl apply -f mari
adb-statefulset2.yaml
statefulset.apps/mariadb configured
PS E:\2 ème anné bts\Bloc1\compte rendu kubernet\dossier yml\MARIADB>
```

La base est maintenant configurée.