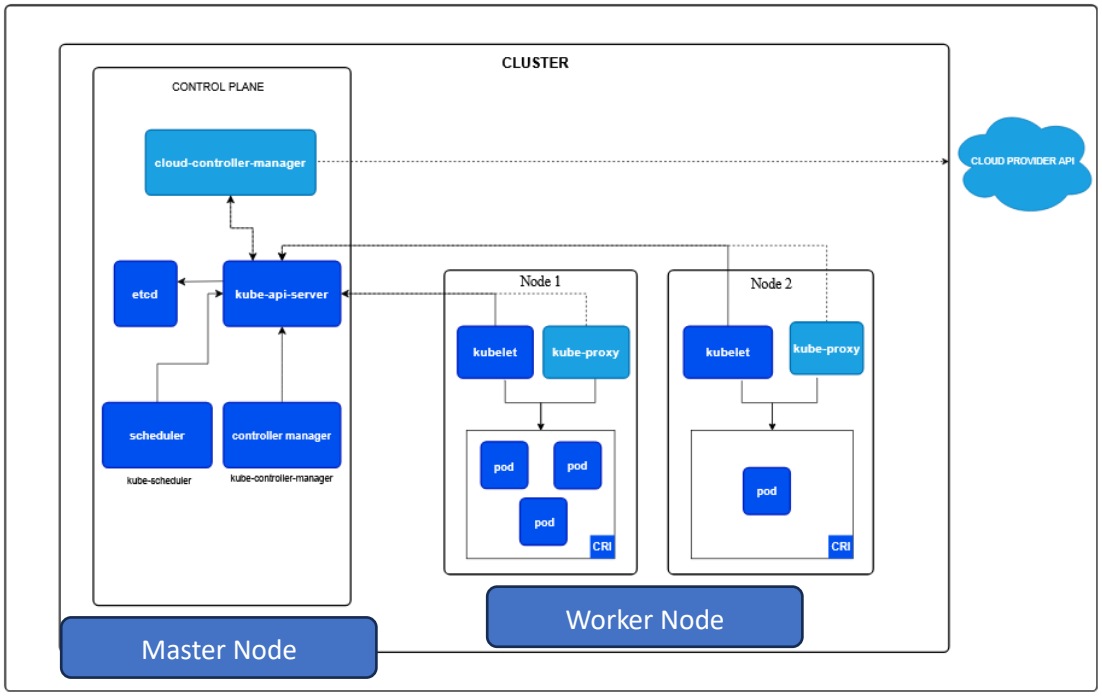
Chapitre 5 – Kubernetes : plateforme de déploiement des applications conteneurisées

Conçu par Google, Kubernetes (Timonier) ou K8s est un système d'orchestration de conteneurs et non pas un logiciel permettant le lancement de conteneurs. Il a donc besoin de docker ou d'un autre logiciel permettant de créer des conteneurs.

Architecture du cluster Kubernetes :

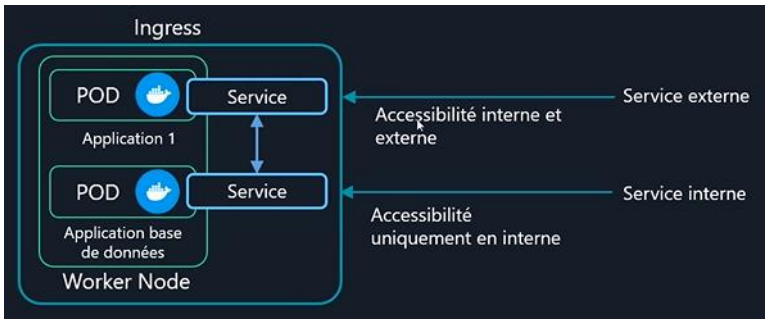
https://kubernetes.io/fr/docs/concepts/architecture/



Les composants (objets) :

Les composants services internes et externes :

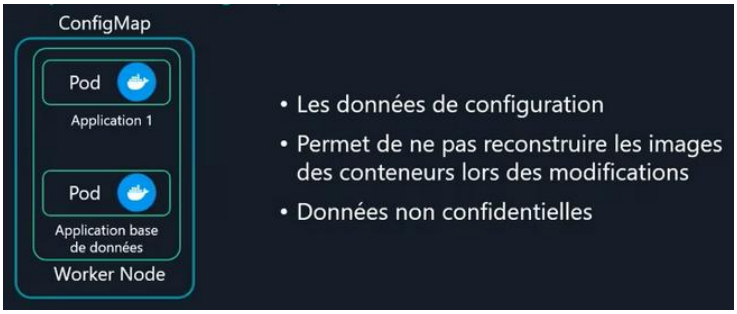
L'objet Ingress :



L'objet Ingress permet d'utiliser des url avec des noms de domaine.

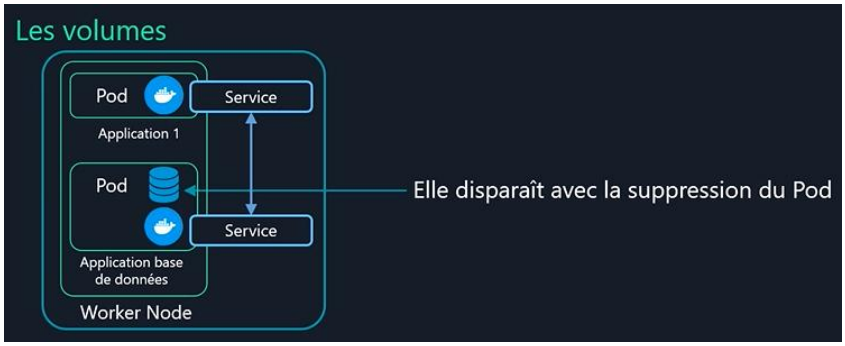
Les cycles de vie entre les pods et les services sont complètement indépendants.

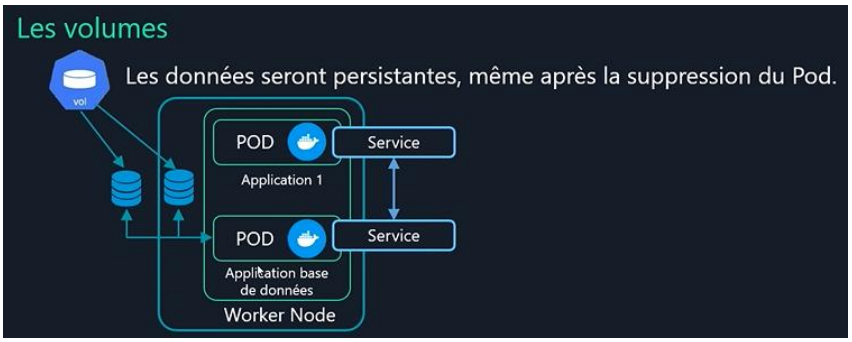
L'objet ConfigMap :



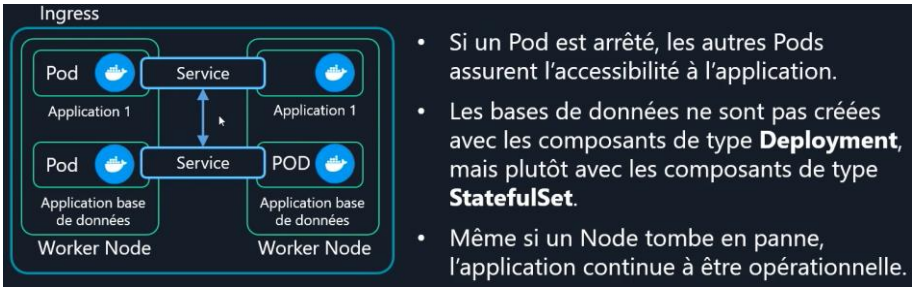
L'objet Secret :

Les volumes :





Le déploiement : manière dont on veut souhaiter déployer l'application



Créer un cluster Kubernetes Local pour l'apprentissage :

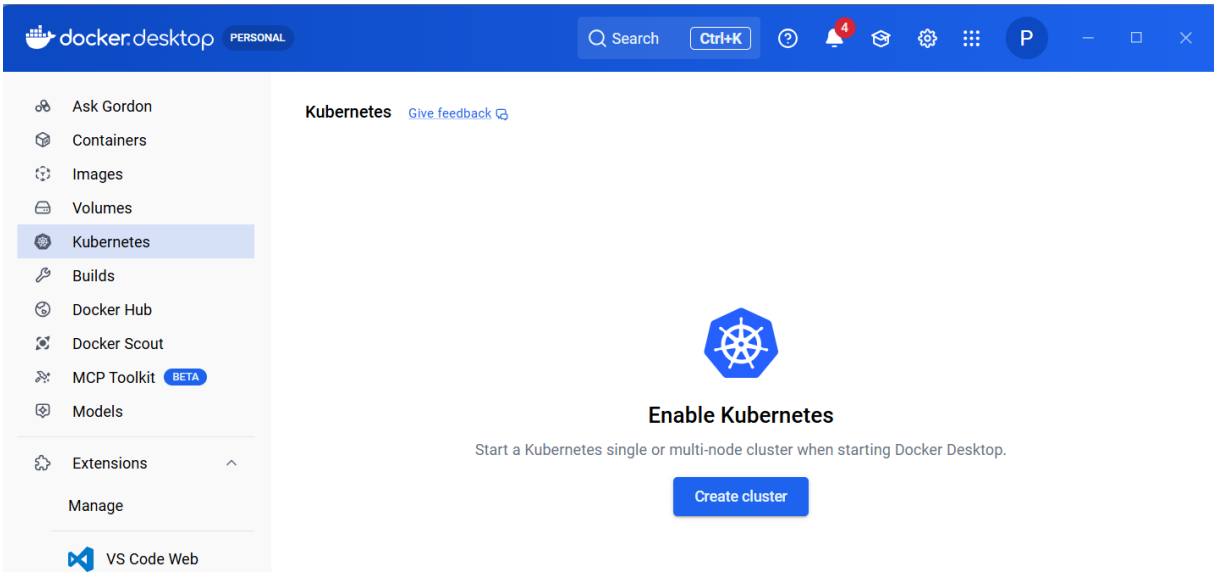
- Minikube ;
- Kind ;
- K3s.

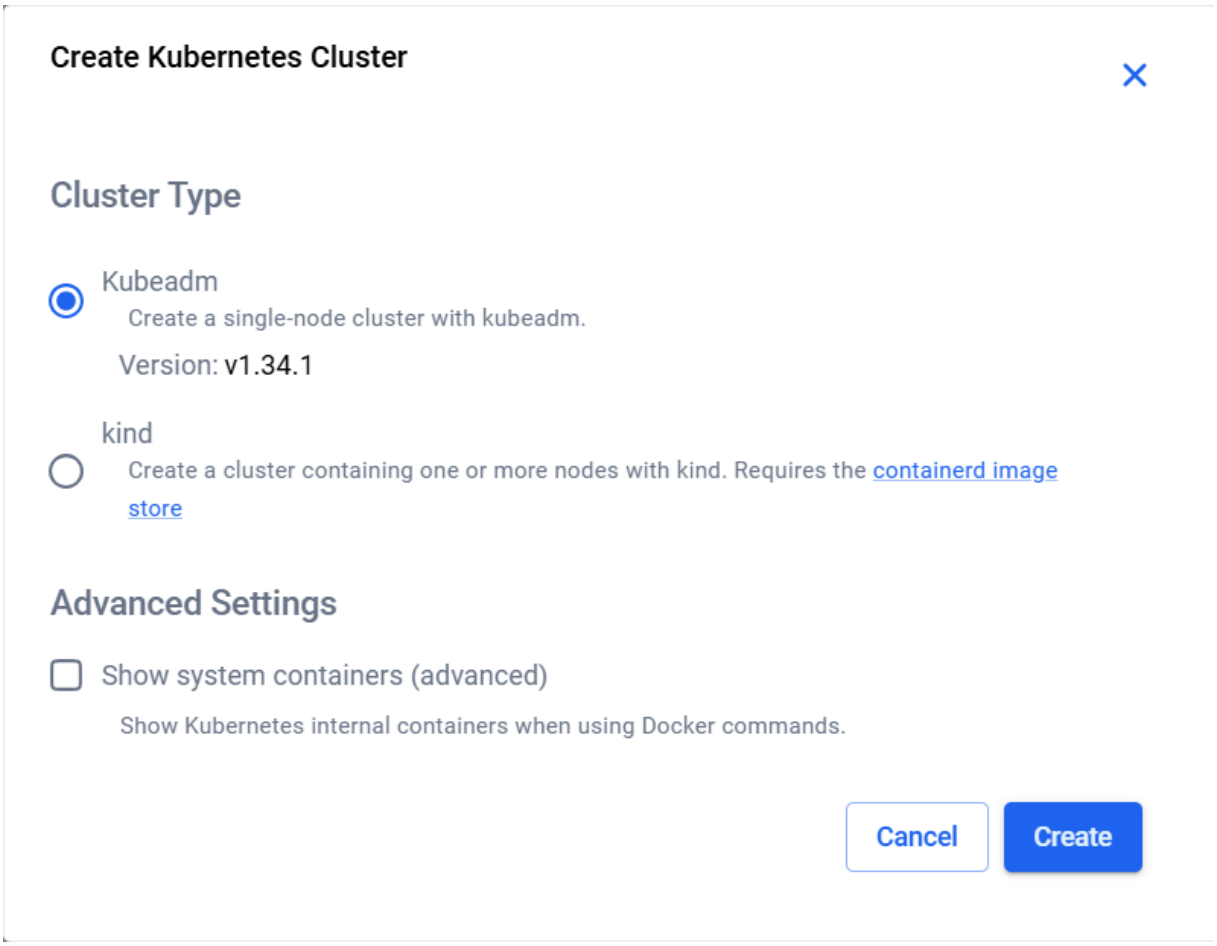
Choisir Minikube est le meilleur choix car :

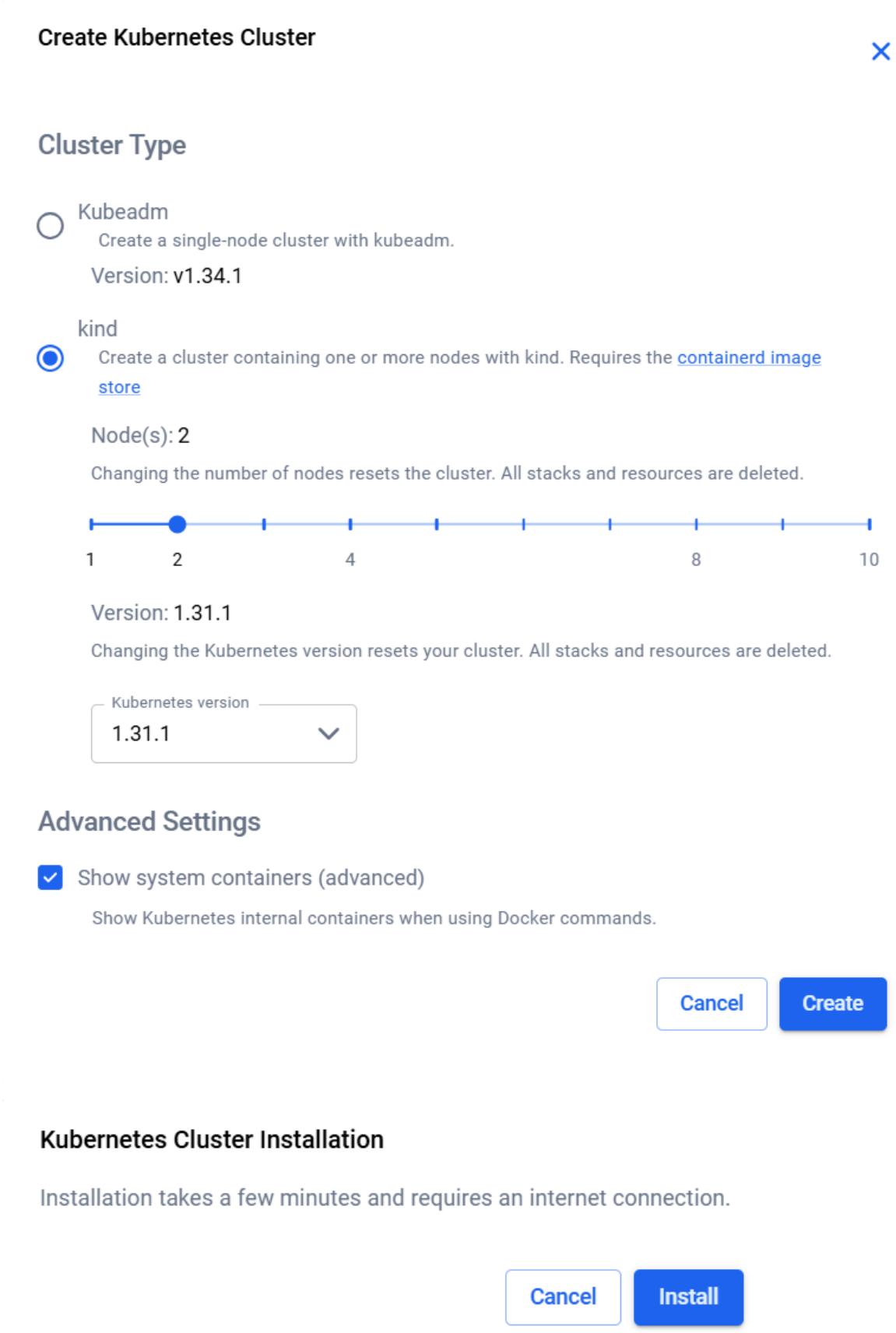
- Environnement très proche de la production ;
- Permet de tester la haute disponibilité ;
- Nombreux Add-ons disponibles tels que dashboard, metrics, ingress.

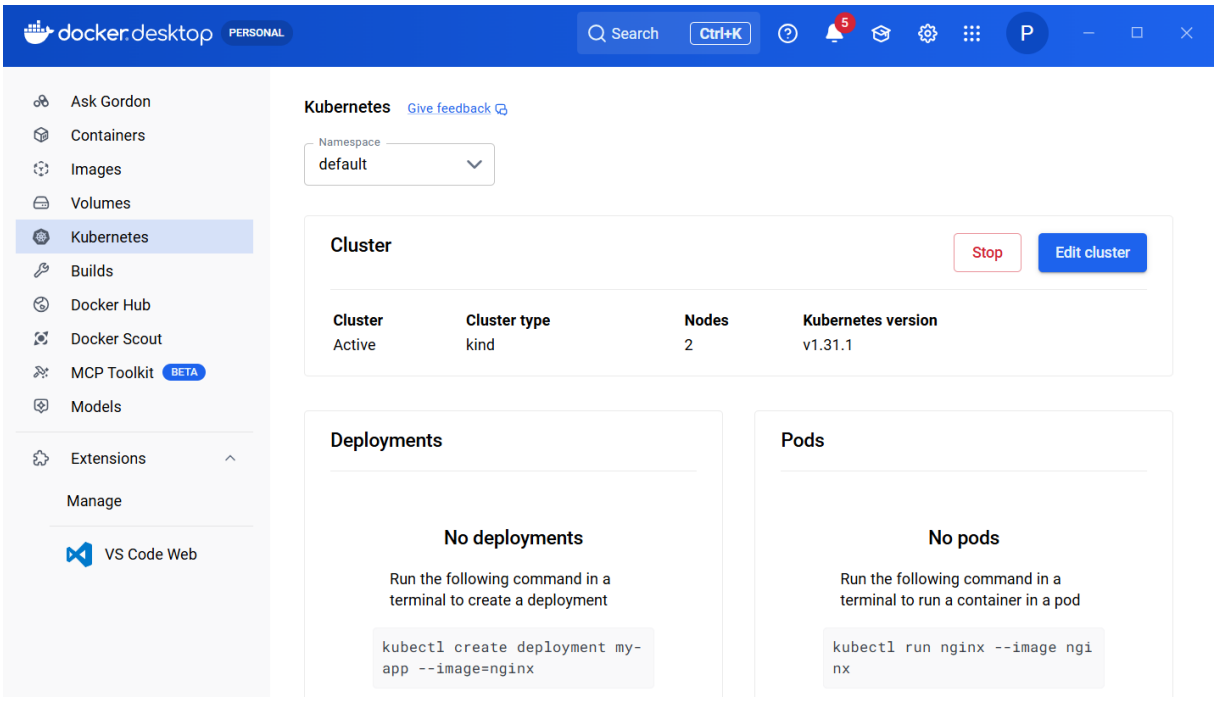
En fonction du driver choisi, chaque nœud est soit un conteneur soit une VM.

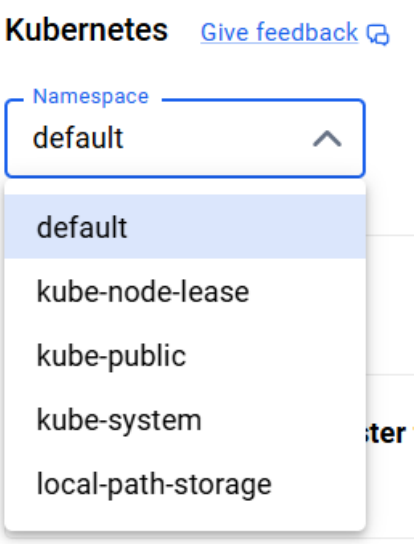
1. Kubernetes avec Docker Desktop.



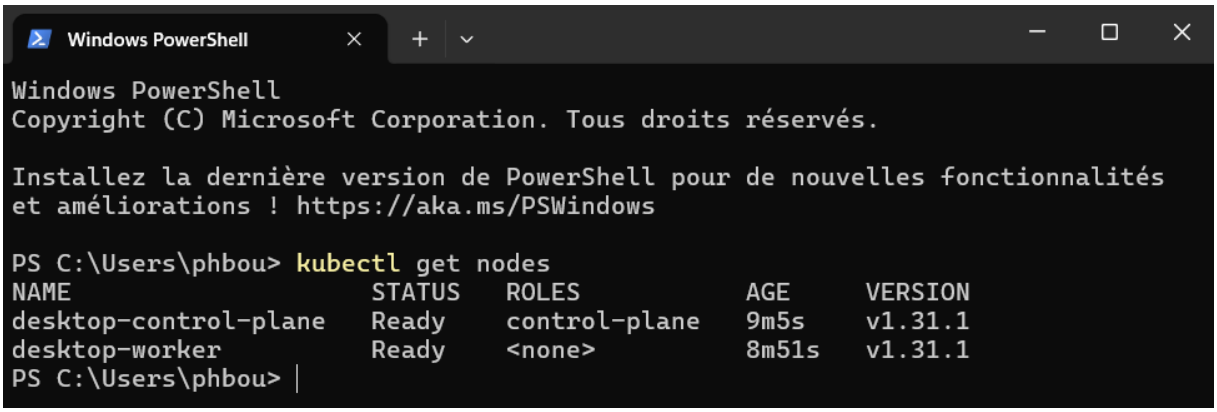

Create Kubernetes Cluster
Cluster Type
Kubeadm
Create a single-node cluster with kubeadm.
Version: v1.34.1
kind
Create a cluster containing one or more nodes with kind. Requires the containerd image store
Advanced Settings
Show system containers (advanced)
Show Kubernetes internal containers when using Docker commands.
Cancel
Create

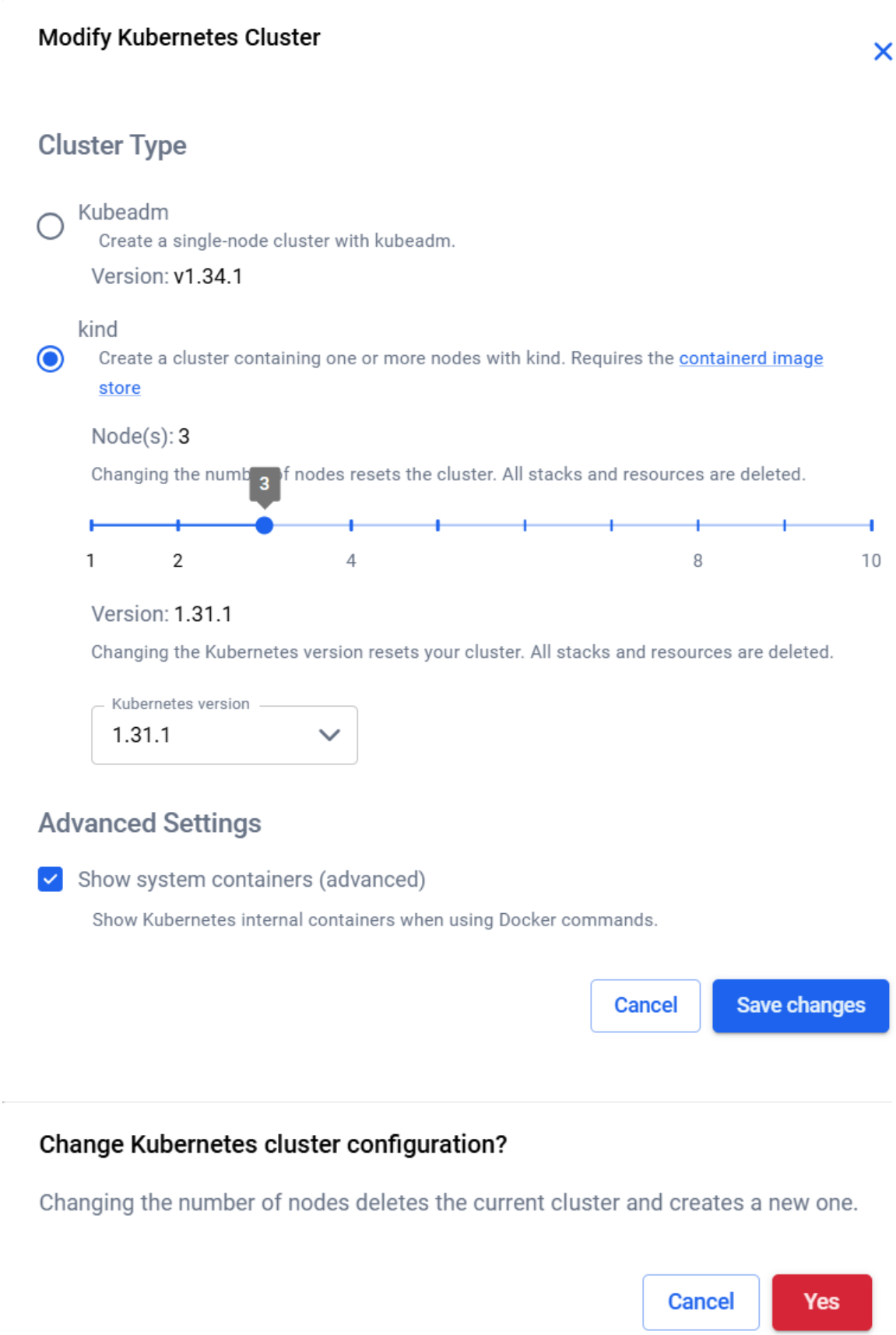

Create Kubernetes Cluster
Cluster Type
Kubeadm
Create a single-node cluster with kubeadm.
Version: v1.34.1
kind
Create a cluster containing one or more nodes with kind. Requires the containerd image store
Node(s): 2
Changing the number of nodes resets the cluster. All stacks and resources are deleted.
1
2
4
8
10
Version: 1.31.1
Changing the Kubernetes version resets your cluster. All stacks and resources are deleted.
Kubernetes version
1.31.1
Advanced Settings
Show system containers (advanced)
Show Kubernetes internal containers when using Docker commands.
Cancel
Create
Kubernetes Cluster Installation
Installation takes a few minutes and requires an internet connection.
Cancel
Install

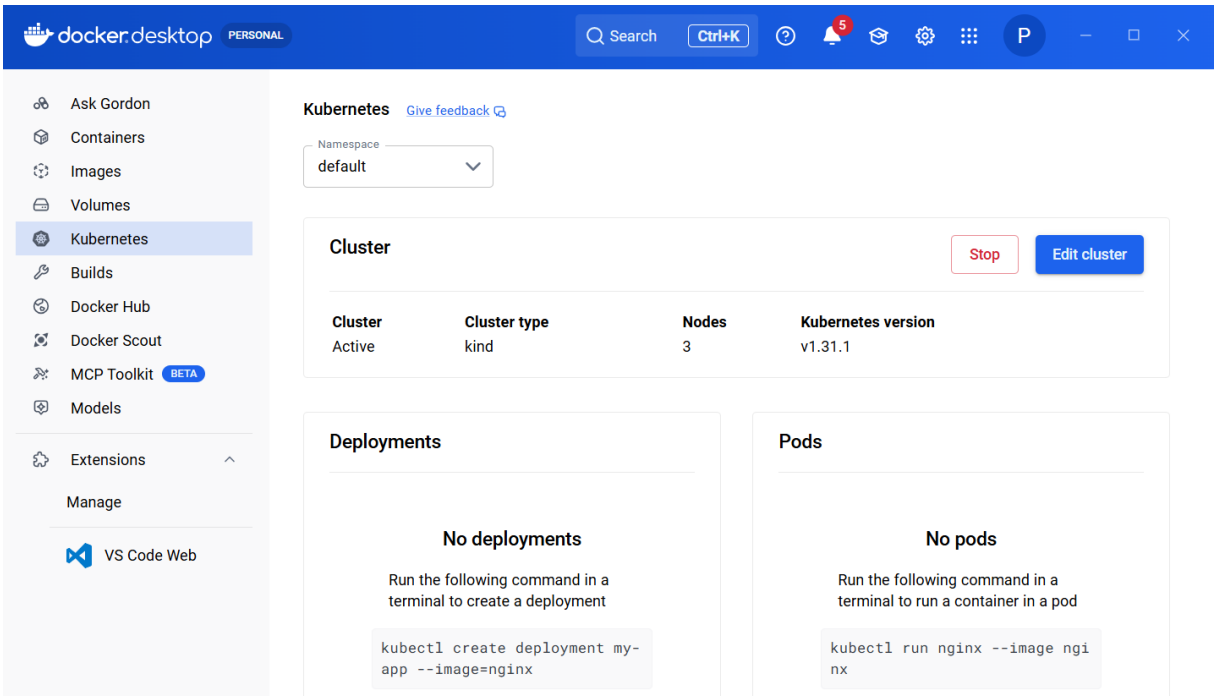




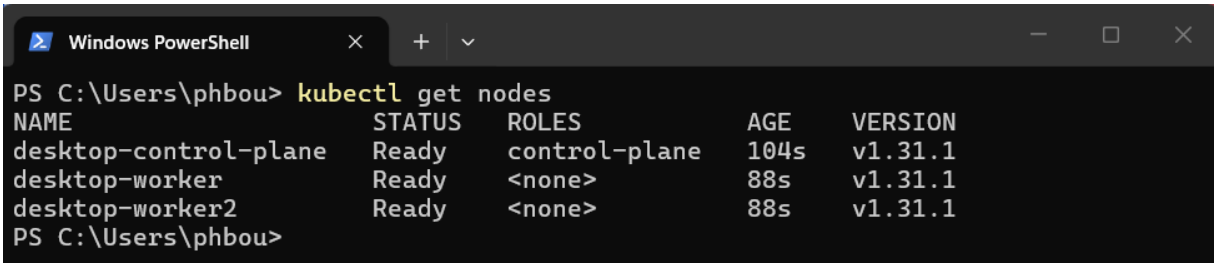
2 nœuds : un maître et un worker

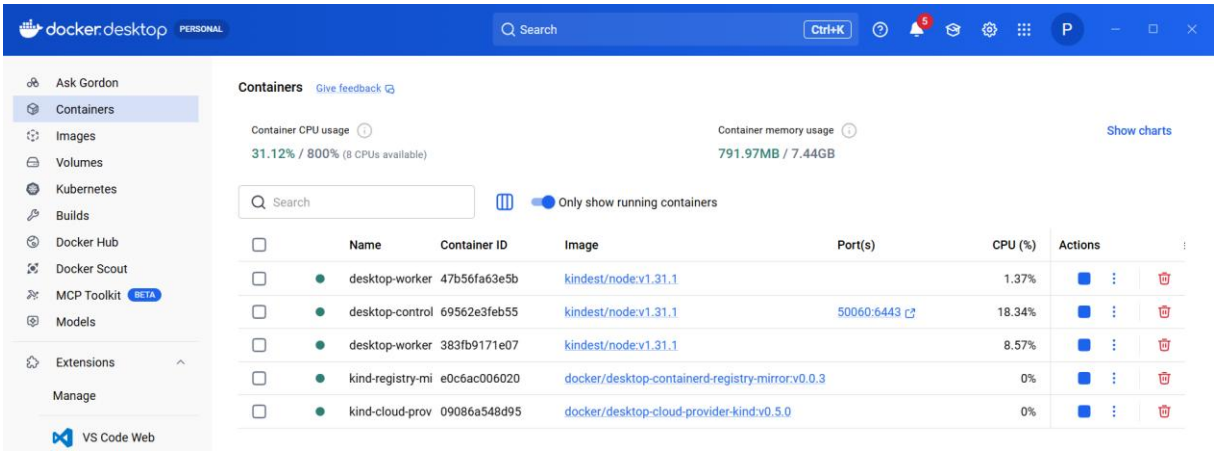



Modify Kubernetes Cluster
Cluster Type
Kubeadm
Create a single-node cluster with kubeadm.
Version: v1.34.1
kind
Create a cluster containing one or more nodes with kind. Requires the containerd image store
Node(s): 3
Changing the numb f nodes resets the cluster. All stacks and resources are deleted.
3
1
2
4
8
10
Version: 1.31.1
Changing the Kubernetes version resets your cluster. All stacks and resources are deleted.
Kubernetes version
1.31.1
Advanced Settings
Show system containers (advanced)
Show Kubernetes internal containers when using Docker commands.
Cancel
Save changes
Change Kubernetes cluster configuration?
Changing the number of nodes deletes the current cluster and creates a new one.
Cancel
Yes



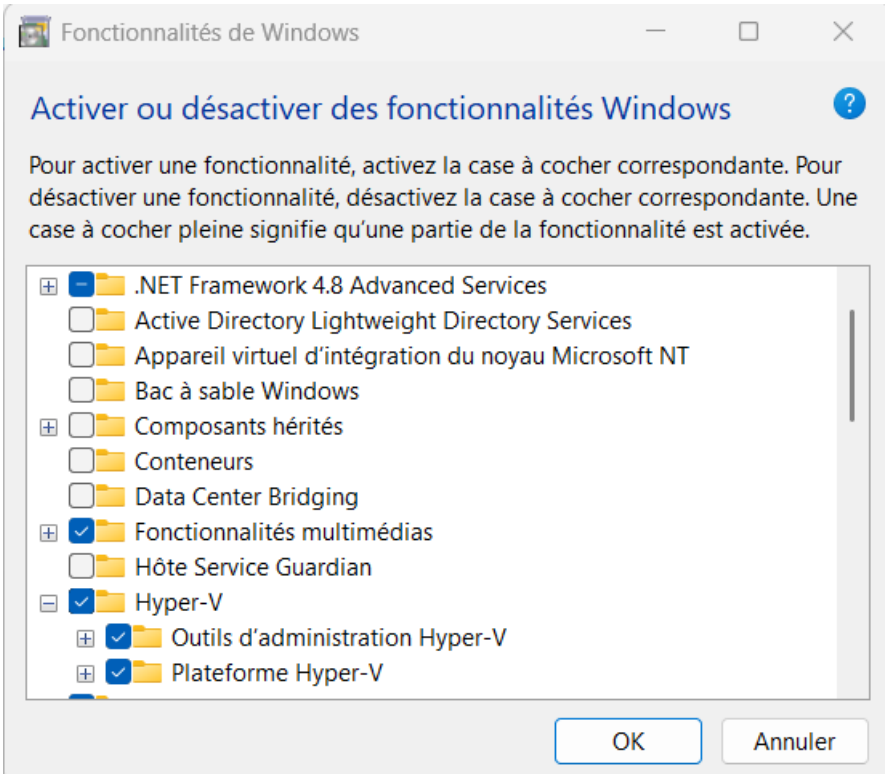
Un maître et 2 workers :





2. Installation de Minikube.

1) Activer la fonctionnalité Windows Hyper-V :



2) Installer Minikube

https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download

minikube start

minikube is local Kubernetes, focusing on making it easy to learn and develop for Kubernetes.

All you need is Docker (or similarly compatible) container or a Virtual Machine environment, and Kubernetes is a single command away: `minikube start`

What you'll need

- 2 CPUs or more
- 2GB of free memory
- 20GB of free disk space
- Internet connection
- Container or virtual machine manager, such as: Docker, QEMU, Hyperkit, Hyper-V, KVM, Parallels, Podman, VirtualBox, or VMware Fusion/Workstation

minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download

Community GitHub Try minikube in browser

1 Installation

Click on the buttons that describe your target platform. For other architectures, see the release page for a complete list of minikube binaries.

Operating system: Linux | macOS | **Windows** | Try in Browser

Architecture: **x86-64**

Release type: **Stable**

Installer type: **.exe download** | Windows Package Manager | Chocolatey

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

1. Download and run the installer for the latest release.
   Or if using `PowerShell`, use this command:

```
New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force
$ProgressPreference = 'SilentlyContinue'; Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri 'https://github.com/kuber
```

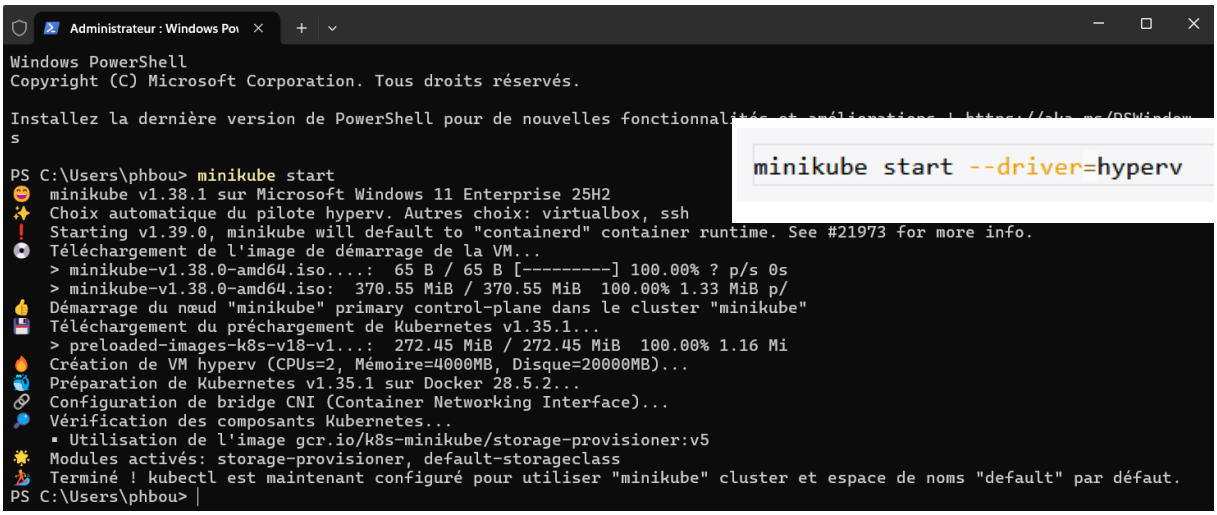
2. Add the `minikube.exe` binary to your `PATH`.
   Make sure to run PowerShell as Administrator.

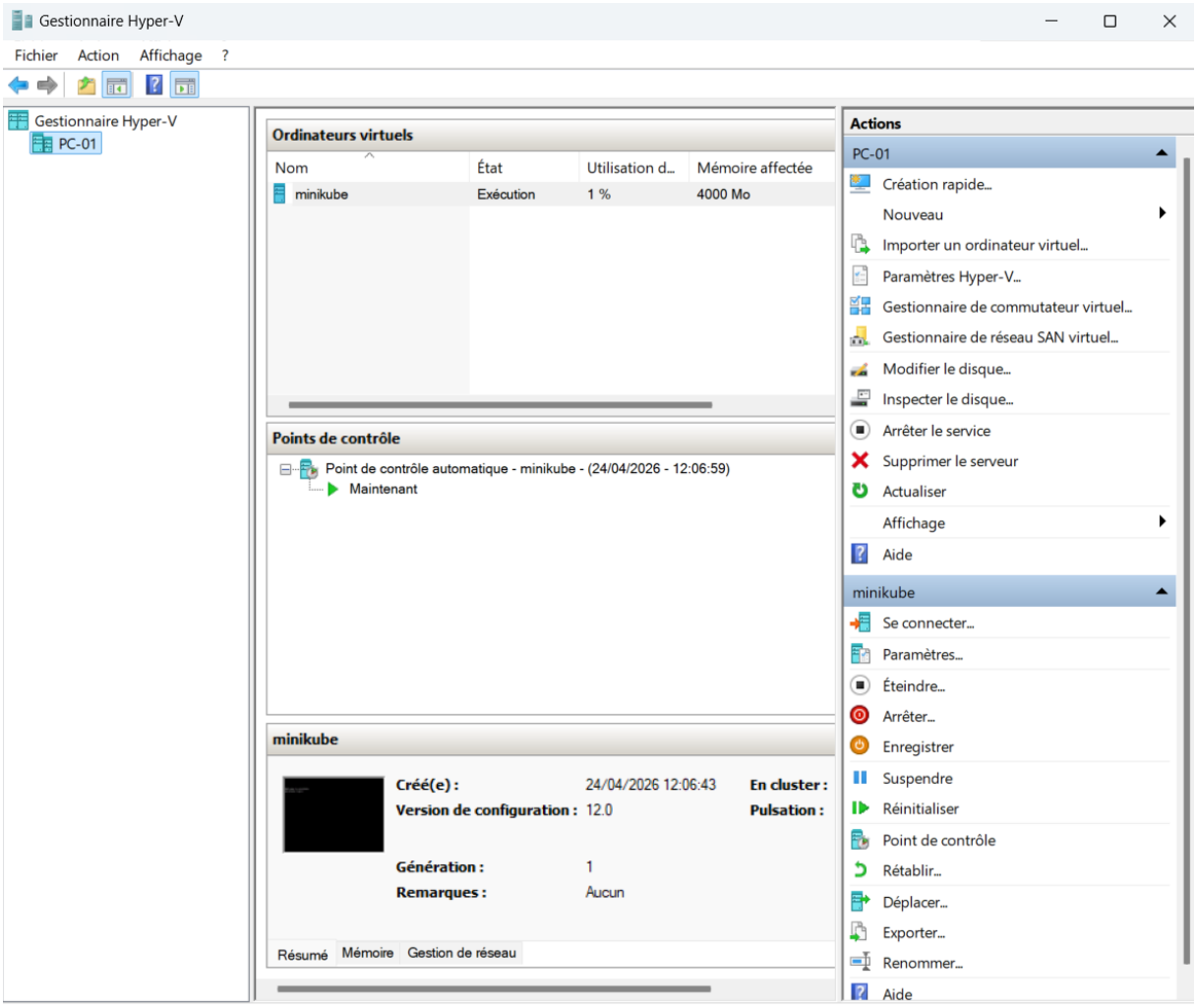
```
$oldPath = [Environment]::GetEnvironmentVariable('Path', [EnvironmentVariableTarget]::Machine)
if ($oldPath.Split(';') -inotcontains 'C:\minikube'){
```

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

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

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

PS C:\Users\phbou> New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force


    Répertoire : C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        24/04/2026     11:45                minikube


PS C:\Users\phbou> $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:\Users\phbou>
```

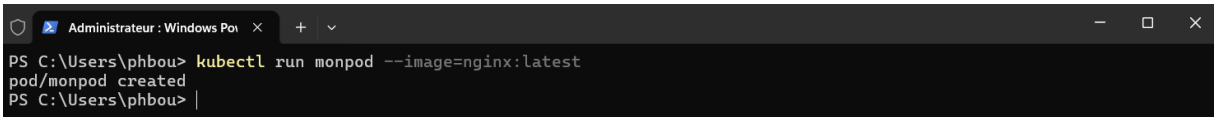
```
Administrateur : Windows Pow
PS C:\Users\phbou> $oldPath = [Environment]::GetEnvironmentVariable('Path', [EnvironmentVariableTarget]::Machine)
PS C:\Users\phbou> if ($oldPath.Split(';') -inotcontains 'C:\minikube'){
>>   [Environment]::SetEnvironmentVariable('Path', $('{0};C:\minikube' -f $oldPath), [EnvironmentVariableTarget]::Machin
e)
>> }
PS C:\Users\phbou>
```

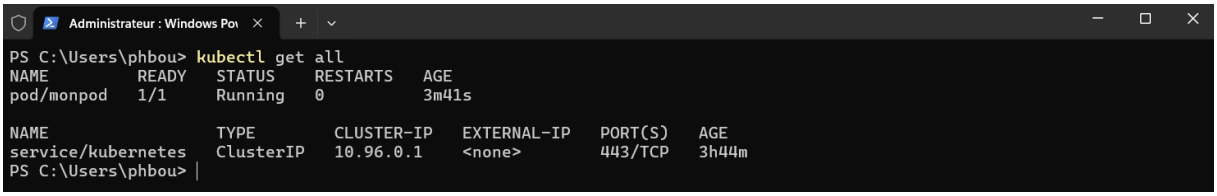




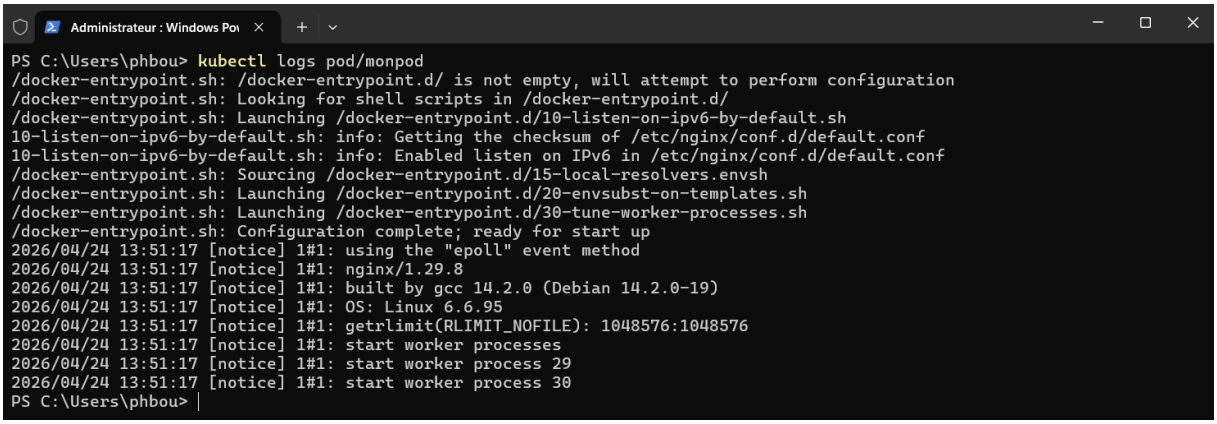
Commande pour lister les objets :

Mon premier objet (type pod) : créer un pod nginx via kubectl :





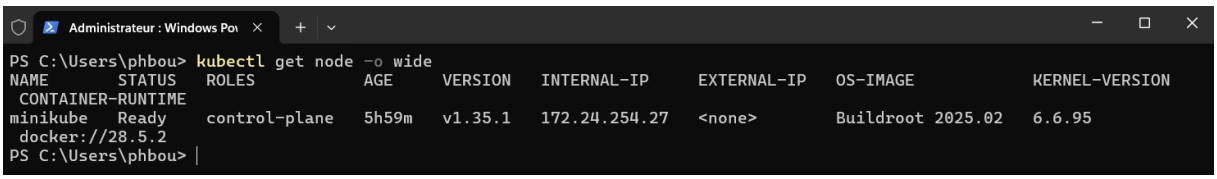
Consulter le log d'un pod :



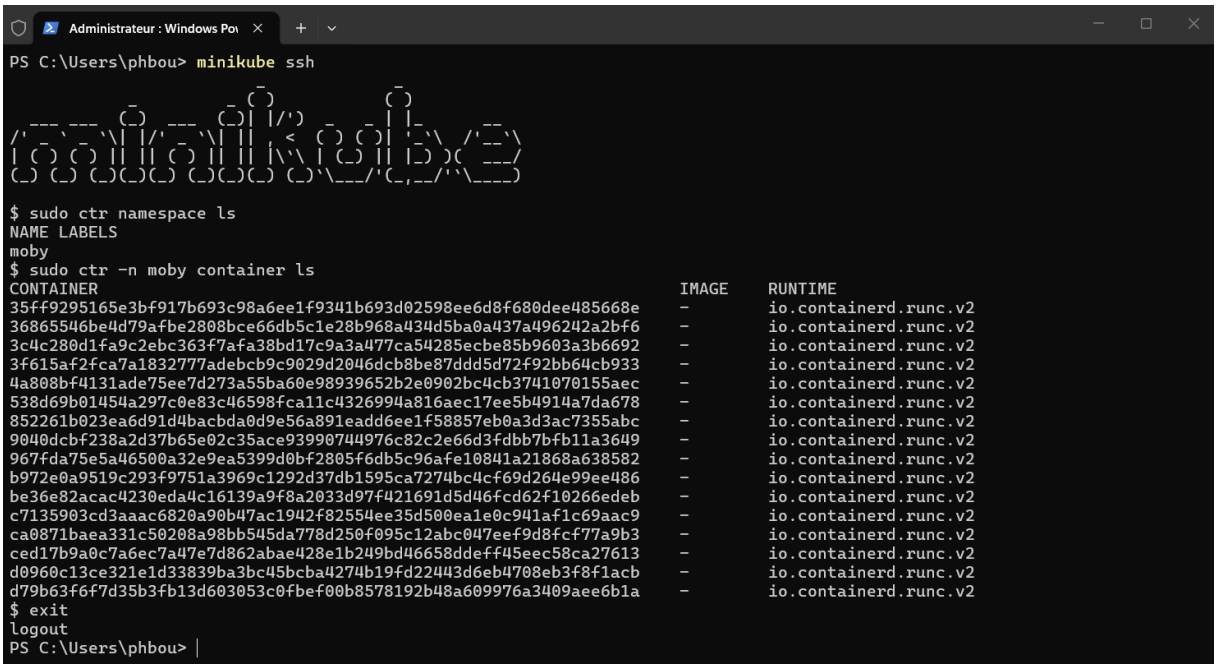
Les espaces de noms dans lesquels sont créés les objets :

Les objets : singulier ou pluriel, peu importe

Champs supplémentaires :



Accéder à Minikube et interroger le démon containerd (commande ctr précédée de sudo afin de récupérer la liste des espaces de noms). Dans le cas où Docker est utilisé comme moteur de conteneur, l'espace de noms porte la valeur moby :

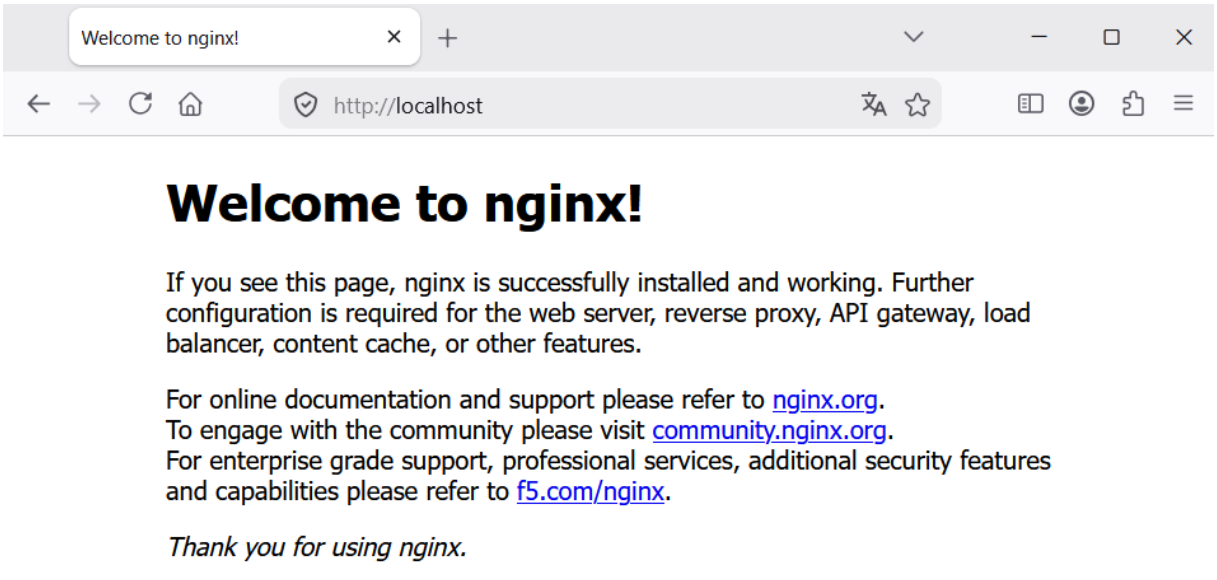


Description d'un objet (en l'occurrence un pod) :

```
Administrateur : Windows Po
PS C:\Users\phbou> kubectl describe pod/monpod
Name:             monpod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.24.254.27
Start Time:       Fri, 24 Apr 2026 15:50:25 +0200
Labels:           run=monpod
Annotations:      <none>
Status:           Running
IP:               10.244.0.3
IPs:
  IP:  10.244.0.3
Containers:
  monpod:
    Container ID:   docker://be36e82acac4230eda4c16139a9f8a2033d97f421691d5d46fcd62f10266edeb
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:6e23479198b998e5e25921dff8455837c7636a67111a04a635cf1bb363d199dc
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Fri, 24 Apr 2026 15:51:16 +0200
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-z5txl (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-z5txl:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    DownwardAPI:             true
```

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

```
Administrateur : Windows Po
PS C:\Users\phbou> kubectl port-forward monpod 80
Forwarding from 127.0.0.1:80 -> 80
Forwarding from [::1]:80 -> 80
Handling connection for 80
```



Détruire le pod nginx :

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

3. Le tableau de bord de Kubernetes.

Activer le plugin :

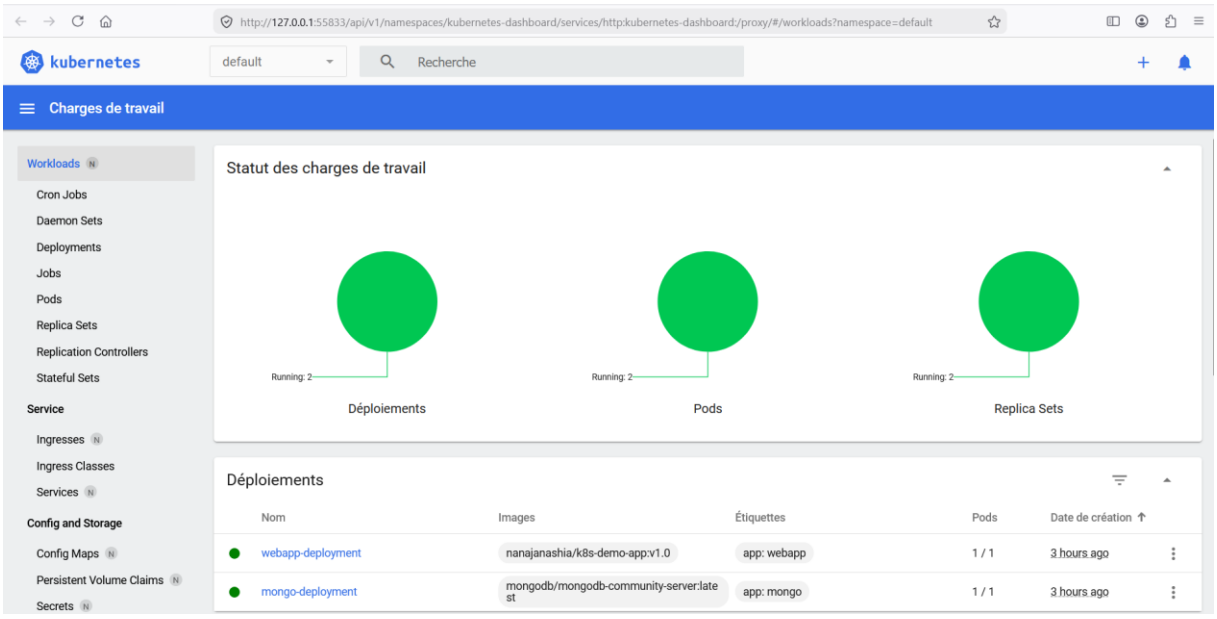
```
Administrateur : Windows Po
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

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

PS C:\Users\phbou> 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/OWNE
RS
    ▪ 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 toute
s les fonctionnalités, veuillez exécuter :

        minikube addons enable metrics-server

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

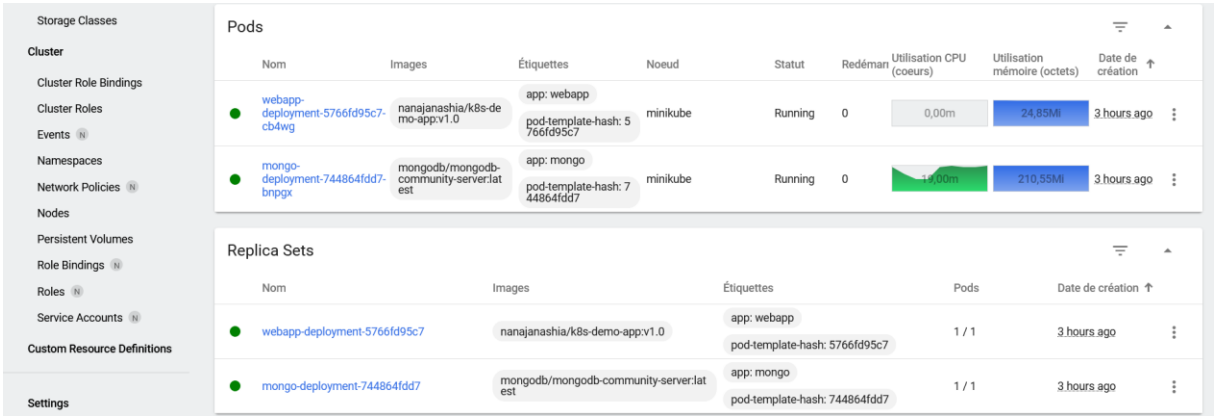
Comme indiqué ci-dessus, activer le serveur de métriques pour bénéficier de l'ensemble des fonctionnalités du tableau de bord :

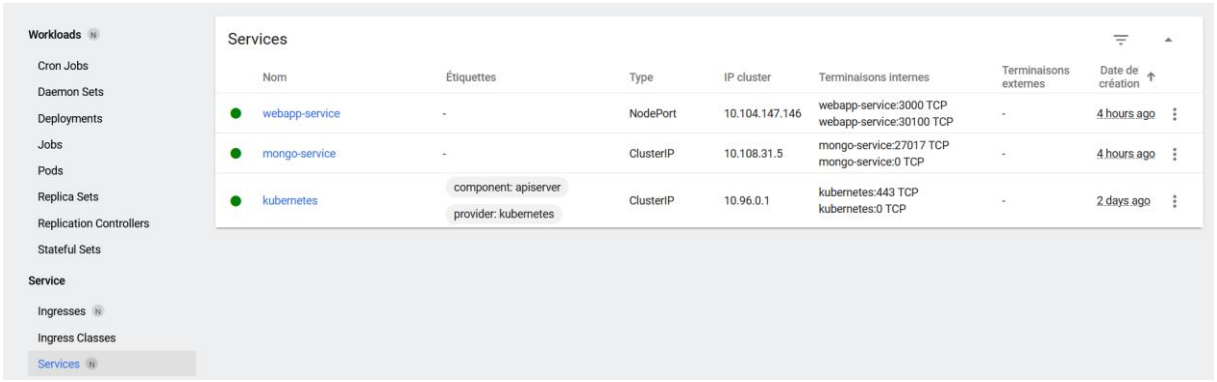
```
Administrateur : Windows Po
PS C:\Users\phbou> minikube addons enable metrics-server
    metrics-server est un addon maintenu par Kubernetes. Pour toute question, contactez minikube sur GitHub.
Vous pouvez consulter la liste des mainteneurs de minikube sur : https://github.com/kubernetes/minikube/blob/master/OWNE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

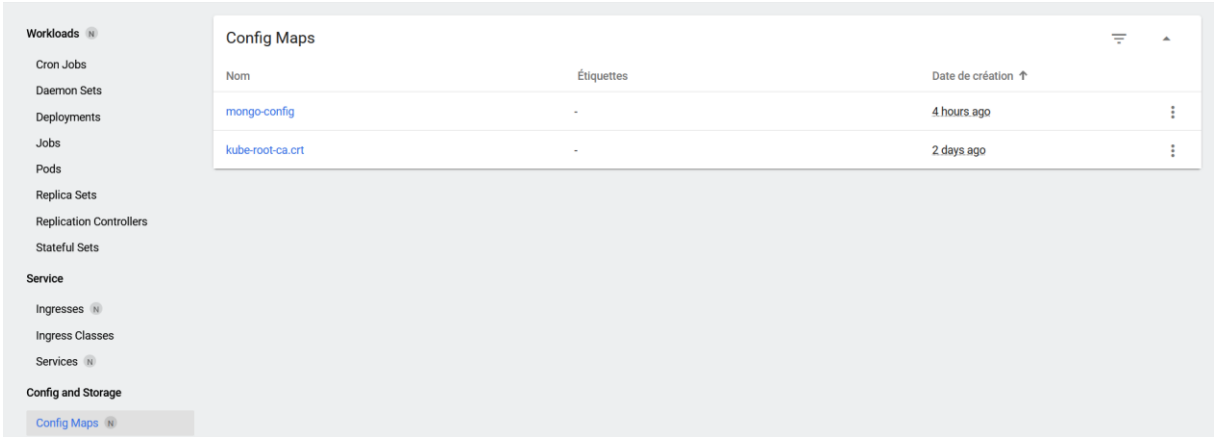
Accès au tableau de bord :

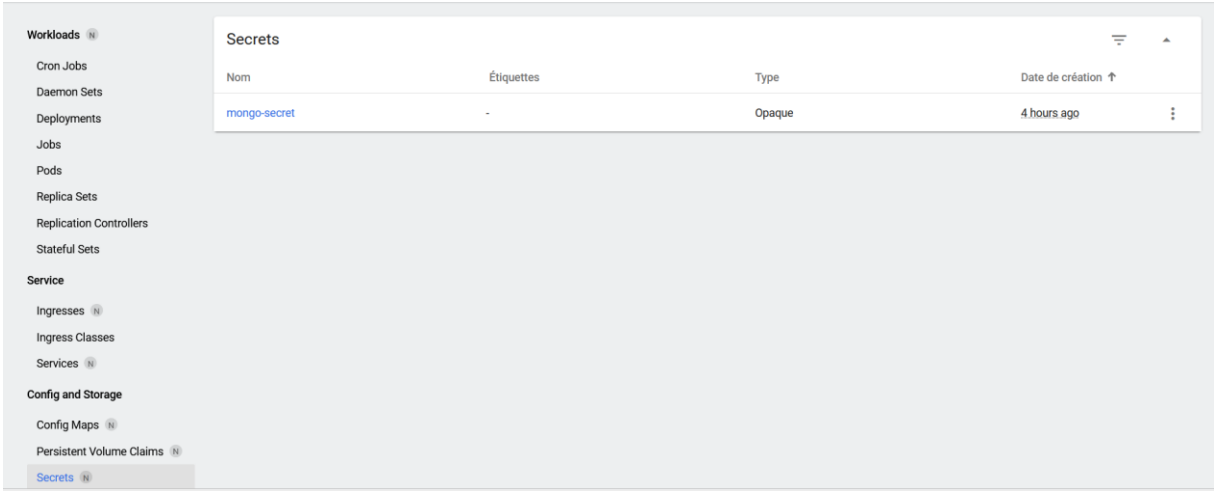
```
Administrateur : Windows Po
PS C:\Users\phbou> minikube dashboard
    Vérification de l'état du tableau de bord...
    Lancement du proxy...
    Vérification de l'état du proxy...
    Ouverture de http://127.0.0.1:55833/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy
/ dans votre navigateur par défaut...
```

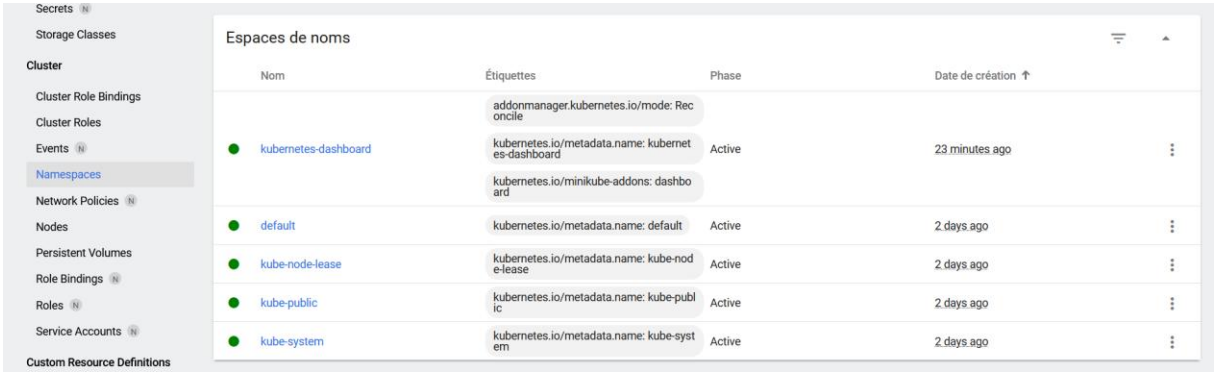


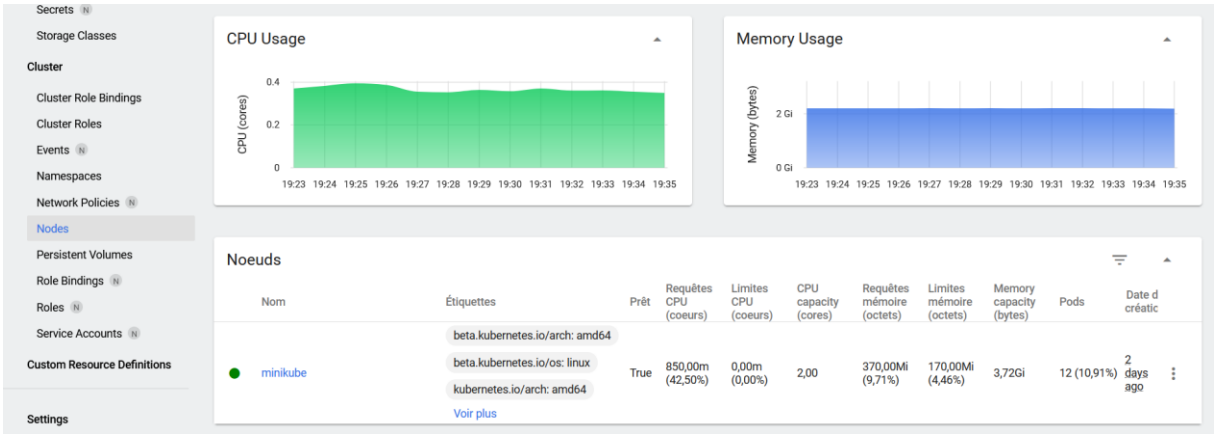

Storage Classes
Cluster
Cluster Role Bindings
Cluster Roles
Events
Namespaces
Network Policies
Nodes
Persistent Volumes
Role Bindings
Roles
Service Accounts
Custom Resource Definitions
Settings
Pods
Nom
Images
Étiquettes
Noeud
Statut
Redémarr
Utilisation CPU (coeurs)
Utilisation mémoire (octets)
Date de création
webapp-deployment-5766fd95c7-cb4wg
nanajanashia/k8s-demo-app:v1.0
app: webapp
pod-template-hash: 5766fd95c7
minikube
Running
0
0,00m
24,85Mi
3 hours ago
mongo-deployment-744864fdd7-bnpgx
mongodb/mongodb-community-server:latest
app: mongo
pod-template-hash: 744864fdd7
minikube
Running
0
19,00m
210,55Mi
3 hours ago
Replica Sets
Nom
Images
Étiquettes
Pods
Date de création
webapp-deployment-5766fd95c7
nanajanashia/k8s-demo-app:v1.0
app: webapp
pod-template-hash: 5766fd95c7
1 / 1
3 hours ago
mongo-deployment-744864fdd7
mongodb/mongodb-community-server:latest
app: mongo
pod-template-hash: 744864fdd7
1 / 1
3 hours ago


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
Terminaisons externes
Date de création
webapp-service
-
NodePort
10.104.147.146
webapp-service:3000 TCP
webapp-service:30100 TCP
-
4 hours ago
mongo-service
-
ClusterIP
10.108.31.5
mongo-service:27017 TCP
mongo-service:0 TCP
-
4 hours ago
kubernetes
component: apiserver
provider: kubernetes
ClusterIP
10.96.0.1
kubernetes:443 TCP
kubernetes:0 TCP
-
2 days ago

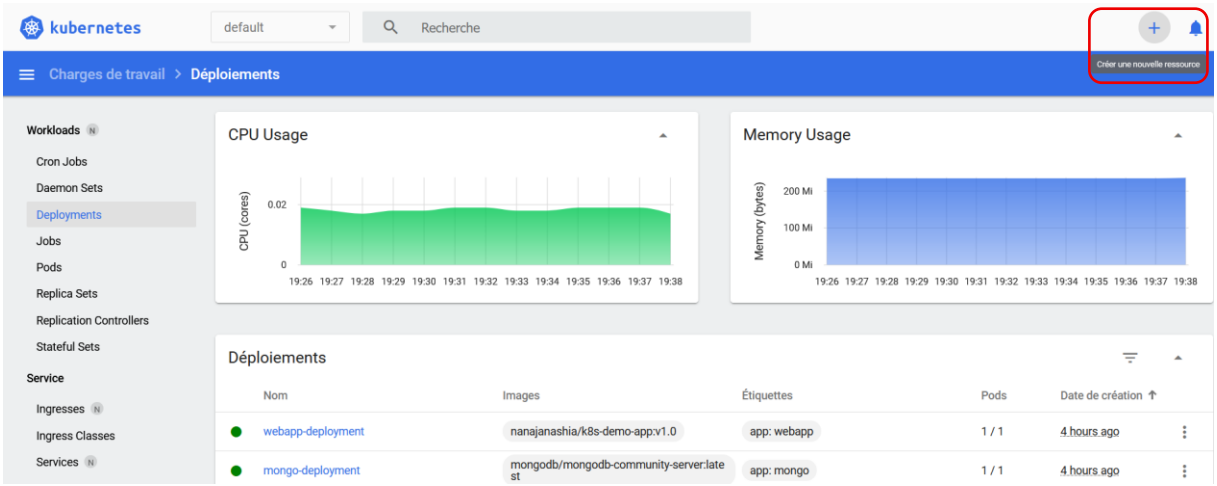

Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Config Maps
Nom
Étiquettes
Date de création
mongo-config
-
4 hours ago
kube-root-ca.crt
-
2 days ago

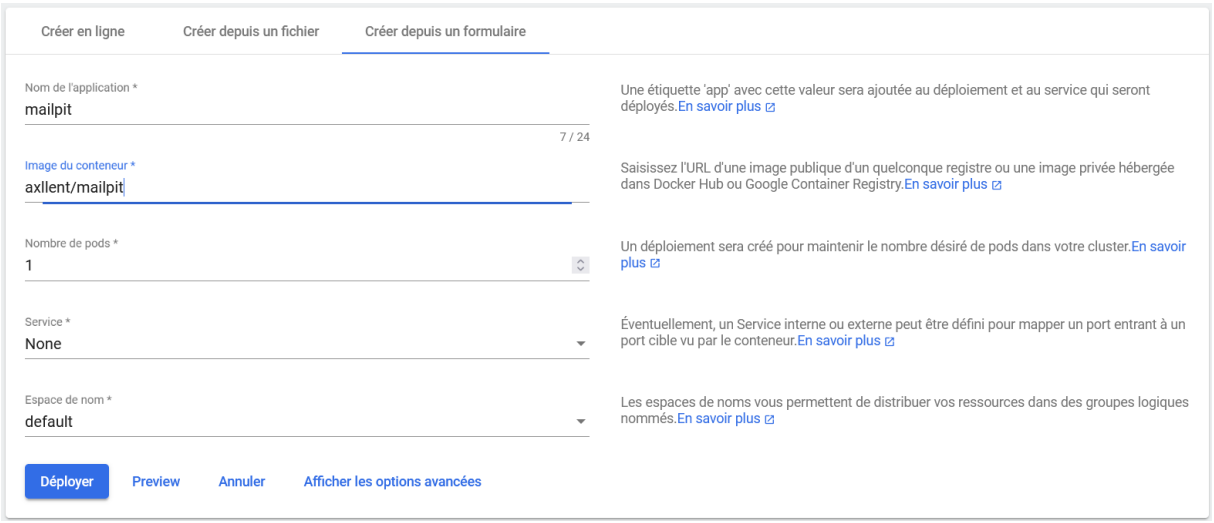

Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Secrets
Nom
Étiquettes
Type
Date de création
mongo-secret
-
Opaque
4 hours ago

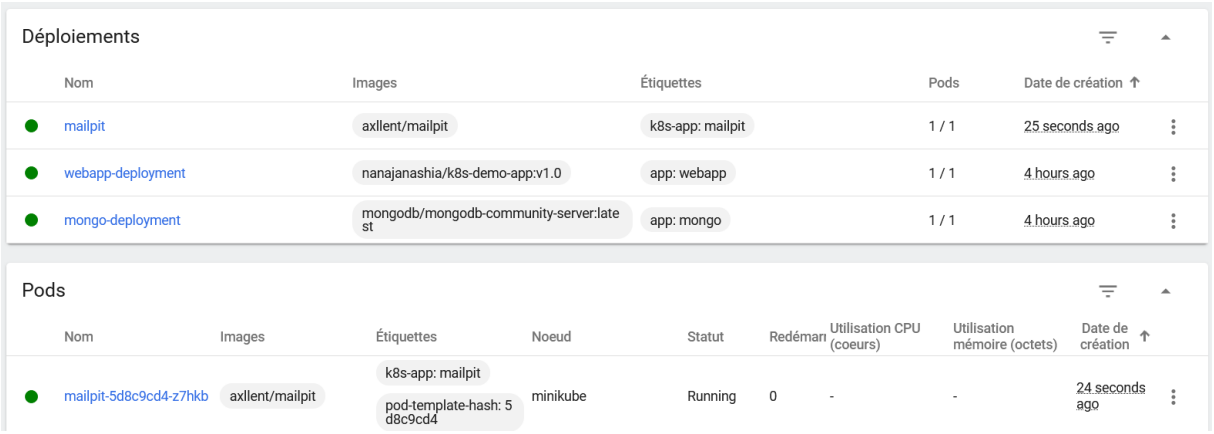


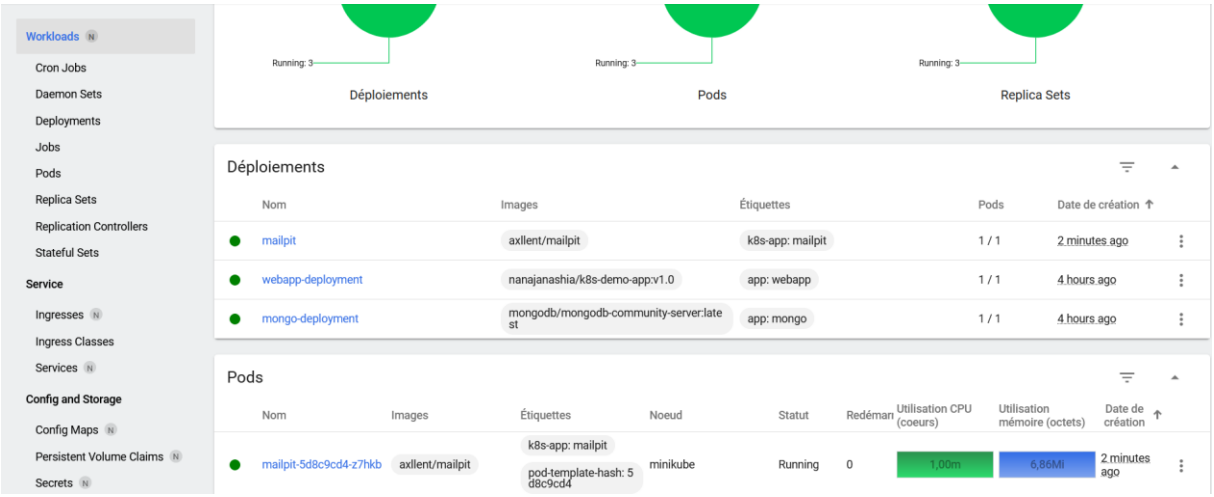


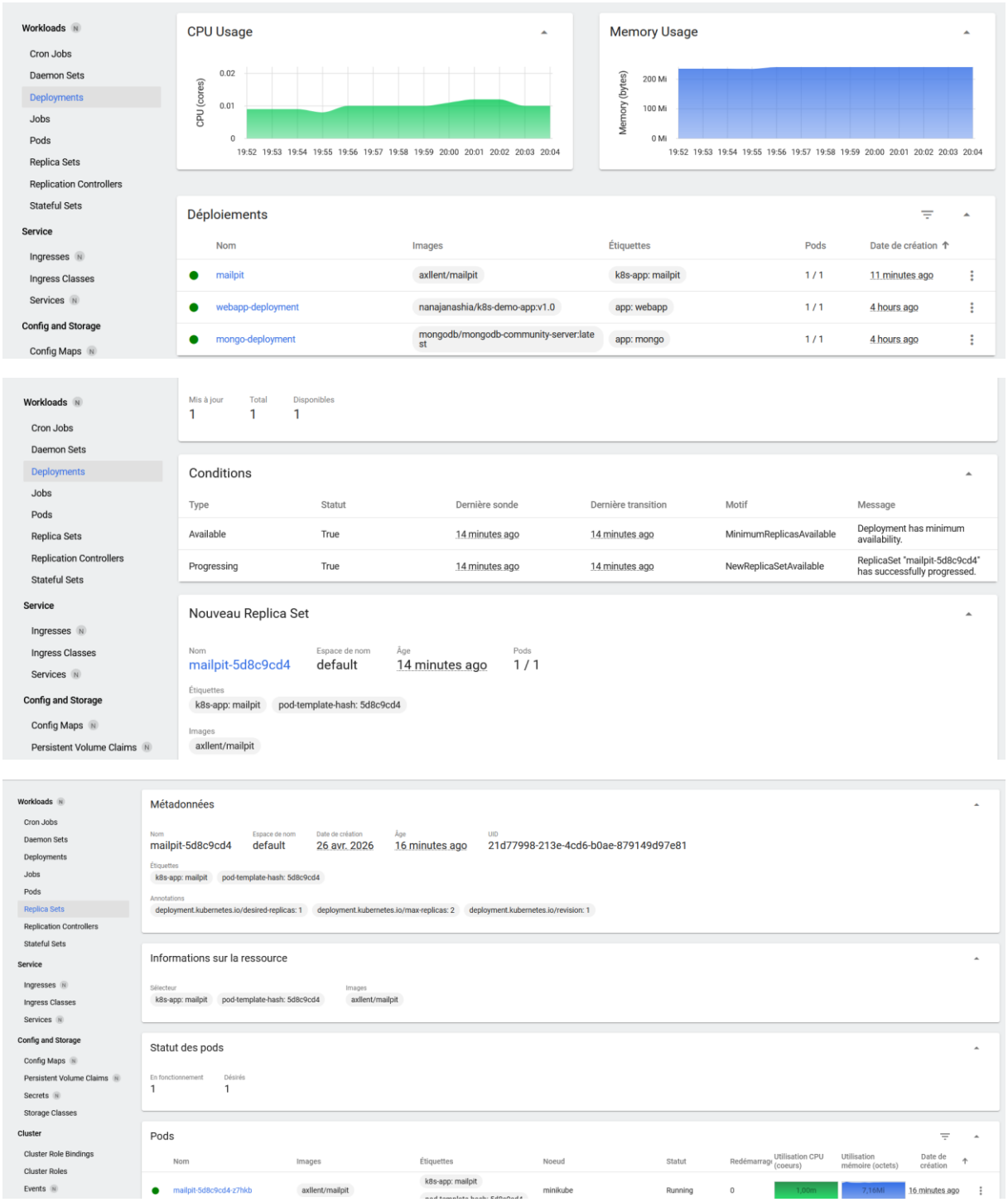
Création d'un déploiement (mailpit) :



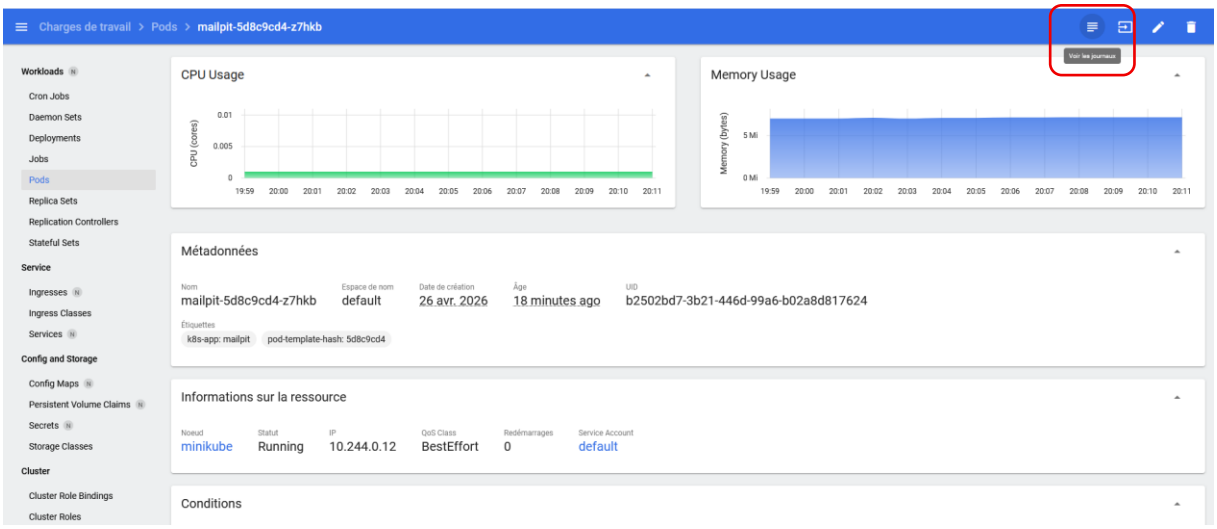

Créer en ligne
Créer depuis un fichier
Créer depuis un formulaire
Nom de l'application *
mailpit
7 / 24
Une étiquette 'app' avec cette valeur sera ajoutée au déploiement et au service qui seront déployés.En savoir plus
Image du conteneur *
axllent/mailpit
Saisissez l'URL d'une image publique d'un quelconque registre ou une image privée hébergée dans Docker Hub ou Google Container Registry.En savoir plus
Nombre de pods *
1
Un déploiement sera créé pour maintenir le nombre désiré de pods dans votre cluster.En savoir plus
Service *
None
Éventuellement, un Service interne ou externe peut être défini pour mapper un port entrant à un port cible vu par le conteneur.En savoir plus
Espace de nom *
default
Les espaces de noms vous permettent de distribuer vos ressources dans des groupes logiques nommés.En savoir plus
Déployer
Preview
Annuler
Afficher les options avancées

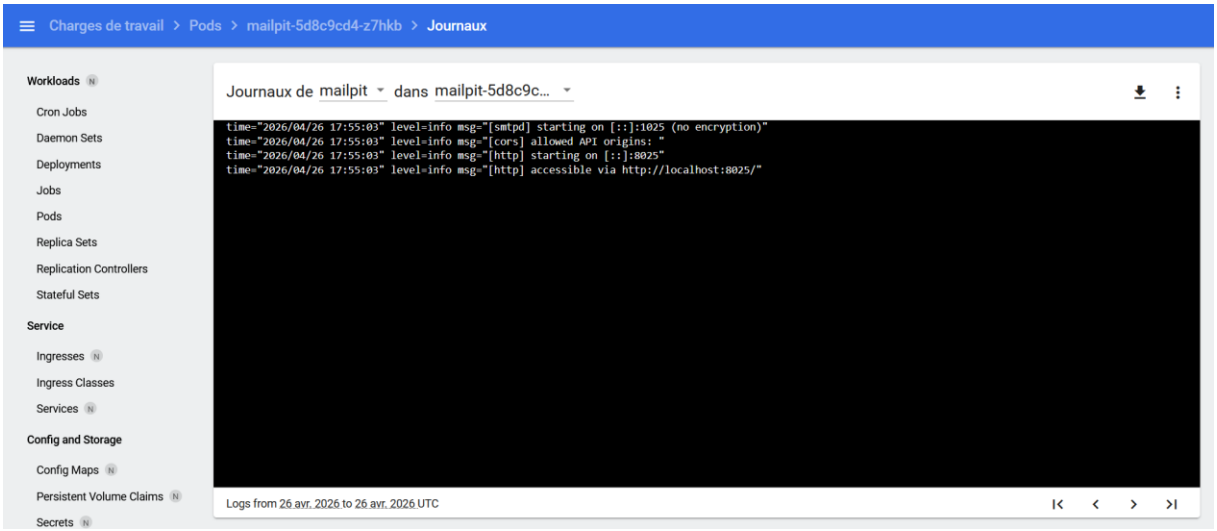

Déploiements
Nom
Images
Étiquettes
Pods
Date de création
mailpit
axllent/mailpit
k8s-app: mailpit
1 / 1
25 seconds ago
webapp-deployment
nanajanashia/k8s-demo-app:v1.0
app: webapp
1 / 1
4 hours ago
mongo-deployment
mongodb/mongodb-community-server:latest
app: mongo
1 / 1
4 hours ago
Pods
Nom
Images
Étiquettes
Noeud
Statut
Redéman
Utilisation CPU (coeurs)
Utilisation mémoire (octets)
Date de création
mailpit-5d8c9cd4-z7hkb
axllent/mailpit
k8s-app: mailpit
pod-template-hash: 5d8c9cd4
minikube
Running
0
-
-
24 seconds ago


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
Running: 3
Déploiements
Running: 3
Pods
Running: 3
Replica Sets
Déploiements
Nom
Images
Étiquettes
Pods
Date de création
mailpit
axllent/mailpit
k8s-app: mailpit
1 / 1
2 minutes ago
webapp-deployment
nanajanashia/k8s-demo-app:v1.0
app: webapp
1 / 1
4 hours ago
mongo-deployment
mongodb/mongodb-community-server:latest
app: mongo
1 / 1
4 hours ago
Pods
Nom
Images
Étiquettes
Noeud
Statut
Redéman
Utilisation CPU (coeurs)
Utilisation mémoire (octets)
Date de création
mailpit-5d8c9cd4-z7hkb
axllent/mailpit
k8s-app: mailpit
pod-template-hash: 5d8c9cd4
minikube
Running
0
1,00m
6,86Mi
2 minutes ago

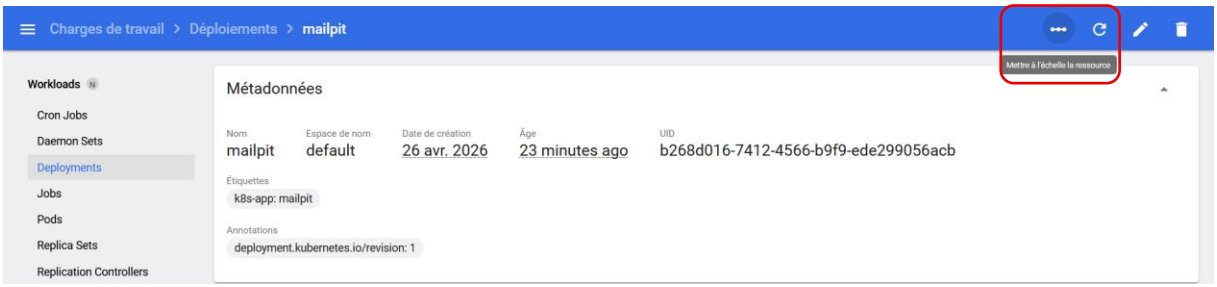


Voir les journaux :





Scalabilité :



Mettre à l'échelle une ressource

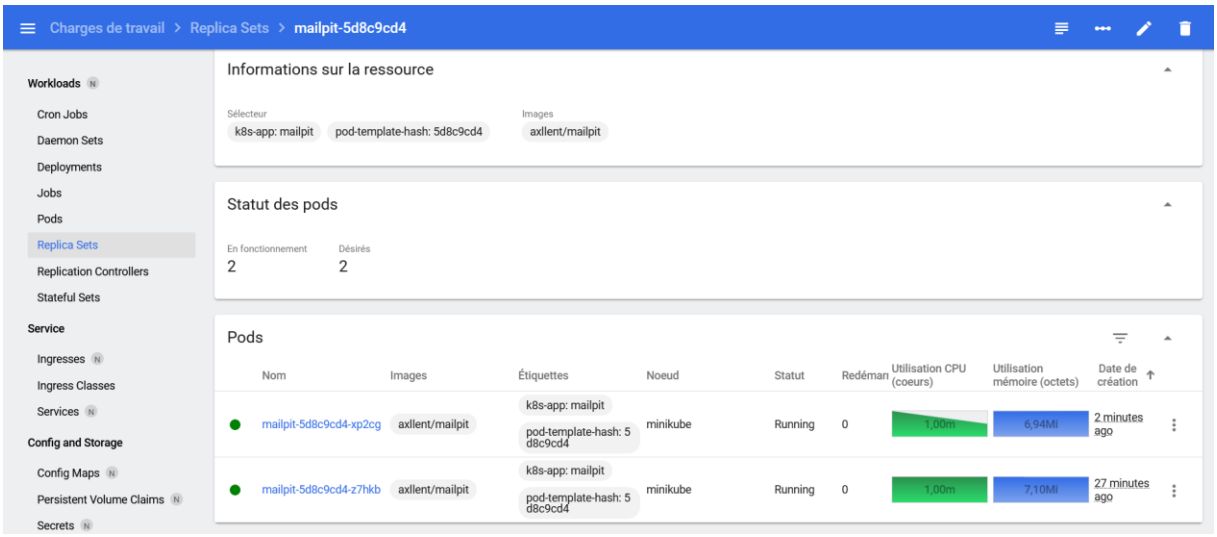
Deployment mailpit will be updated to reflect the desired replicas count.

Répliques désirées * 2

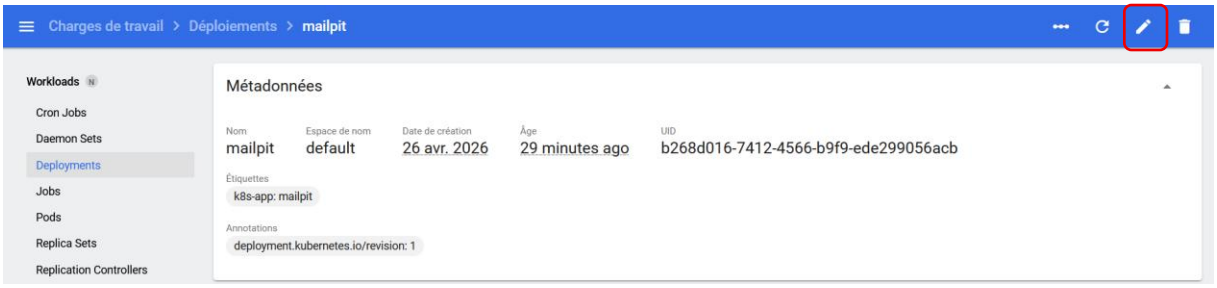
Répliques actuelles 1

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

Mettre à l'échelle Annuler



Mise à jour :



Éditer une ressource

YAML JSON

```yaml
kind: Deployment
apiVersion: apps/v1
metadata:
  name: mailpit
  namespace: default
  uid: b268d016-7412-4566-b9f9-ede299056acb
  resourceVersion: '64618'
  generation: 2
  creationTimestamp: '2026-04-26T17:54:48Z'
  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

Éditer une ressource

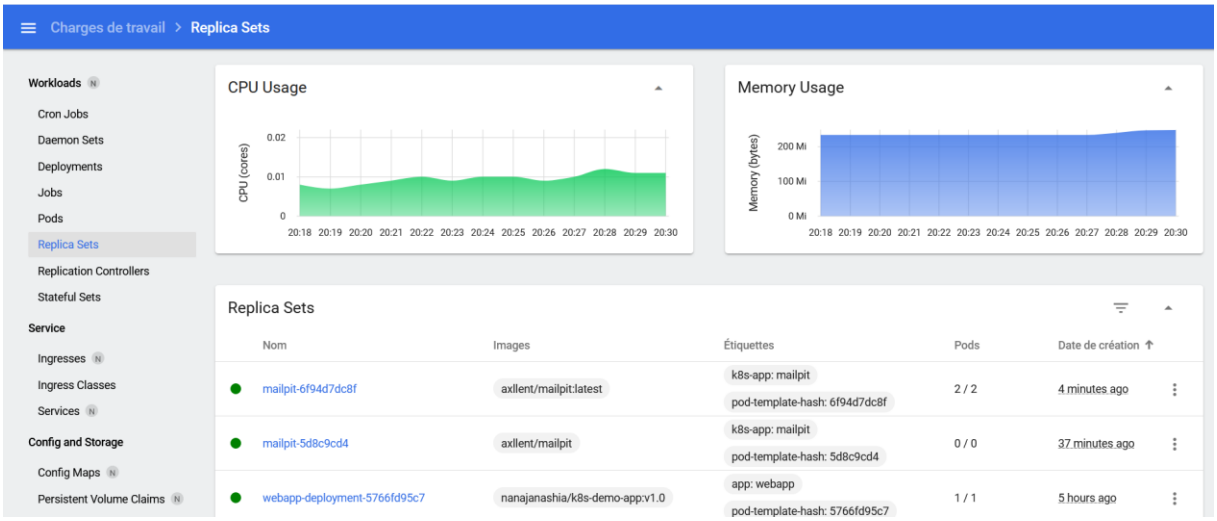
YAML JSON

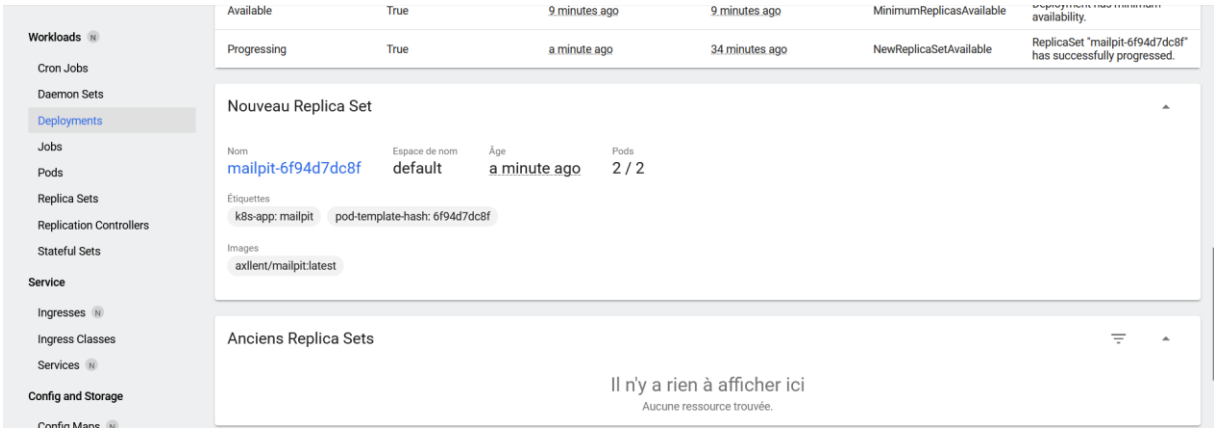
```yaml
            f:observedGeneration: {}
            f:readyReplicas: {}
            f:replicas: {}
            f:terminatingReplicas: {}
            f:updatedReplicas: {}
      subresource: status
spec:
  replicas: 2
  selector:
    matchLabels:
      k8s-app: mailpit
  template:
    metadata:
      name: mailpit
      labels:
        k8s-app: mailpit
    spec:
      containers:
        - name: mailpit
          image: axllent/mailpit:latest
          resources: {}
          terminationMessagePath: /dev/termination-log
          terminationMessagePolicy: File
```

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

Mettre à jour Annuler

Kubernetes va créer un nouveau ReplicaSet :





En résumé, la création d'un déploiement d'application se traduit par :

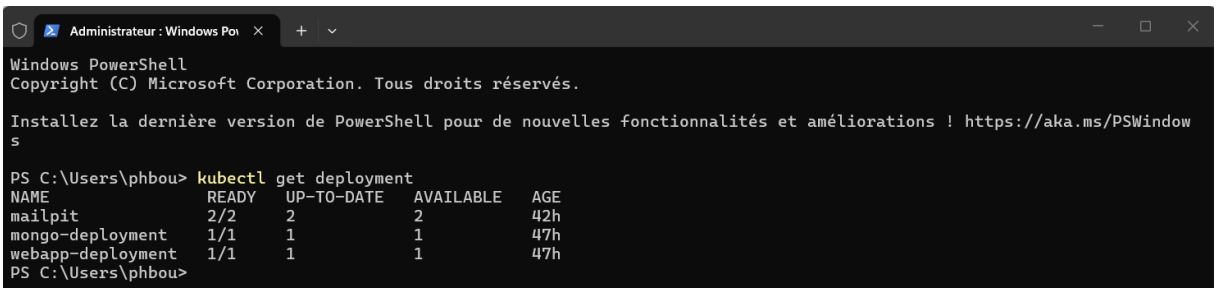
- La création d'un ReplicaSet ;
- La création d'un Pod associé au ReplicaSet ;
- Le démarrage d'un conteneur.

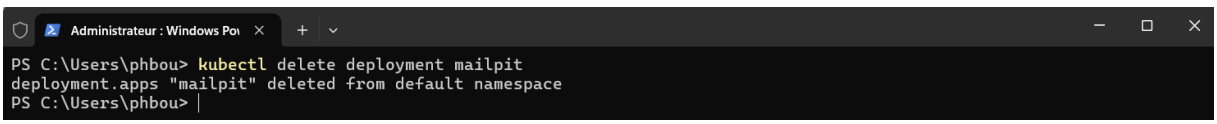
Une mise à jour du déploiement se traduira par la création d'un nouveau ReplicaSet

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

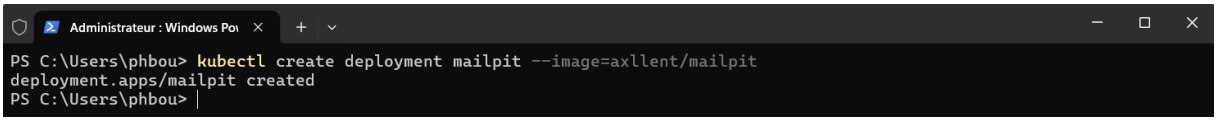
Le § précédent a permis le déploiement d'une application via le dashboard. Cette fois-ci, les opérations seront réalisées en ligne de commande via la commande kubectl.

Suppression d'un déploiement :

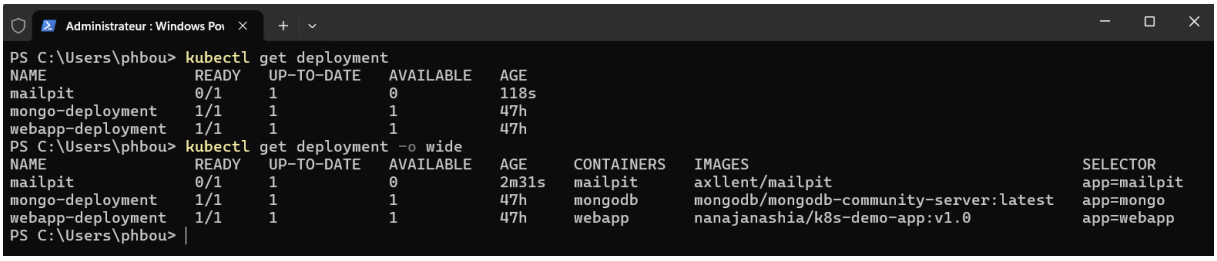


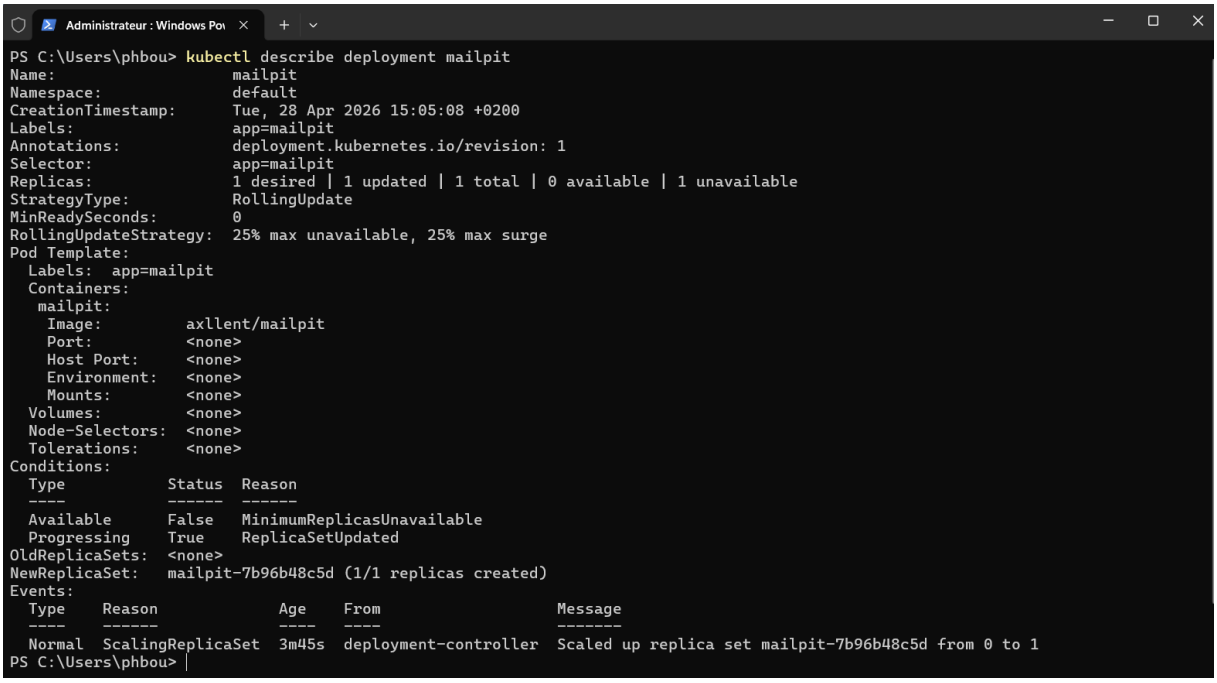


Création d'un déploiement

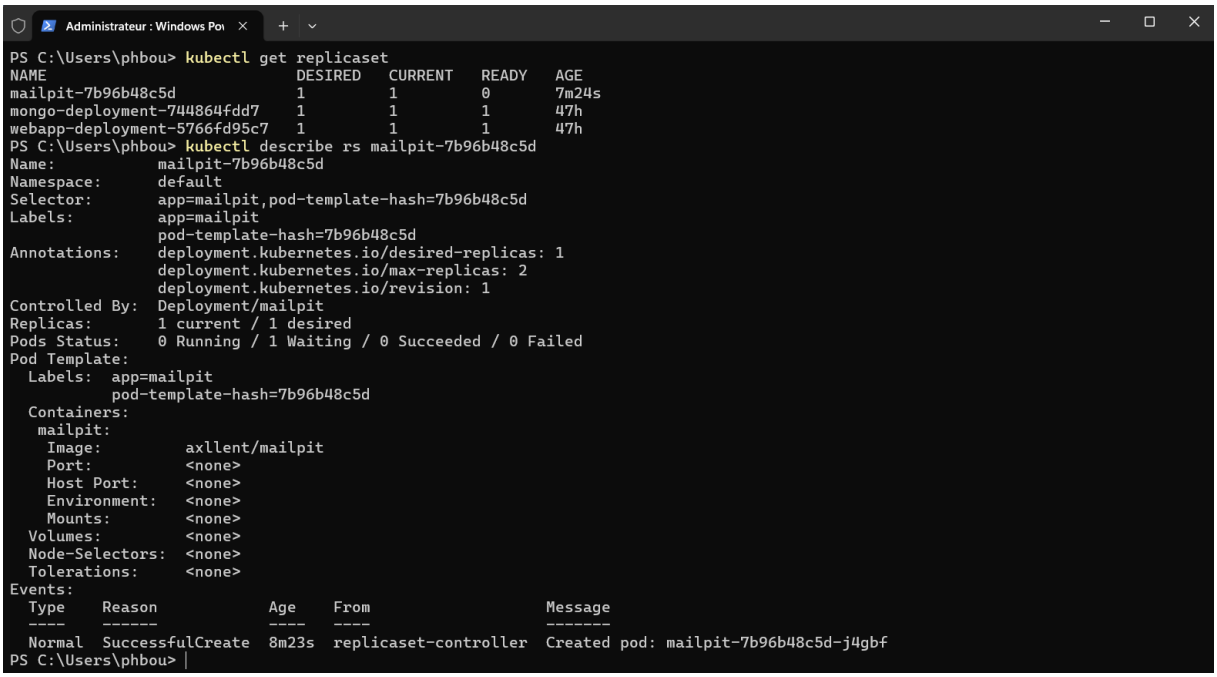


Etat du déploiement

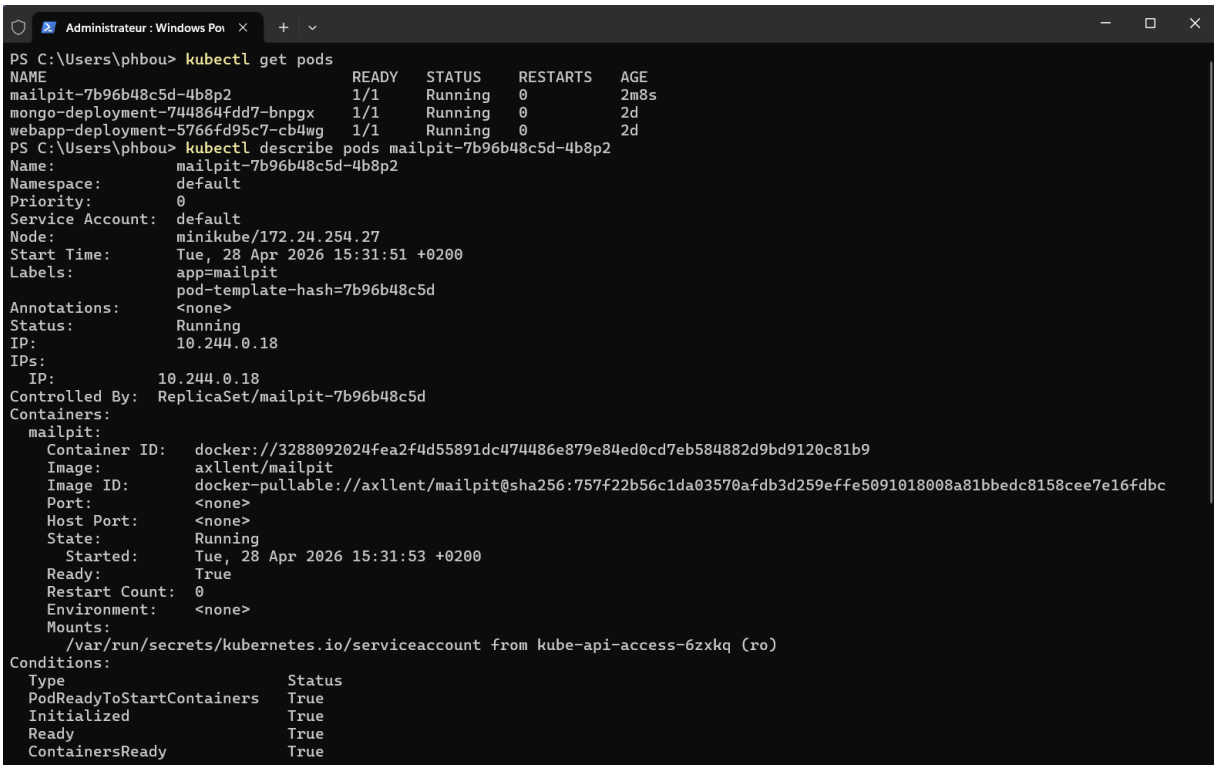




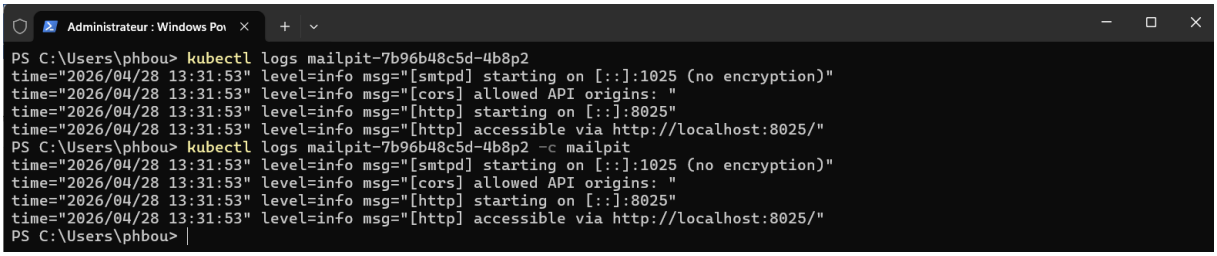
Réplicas :



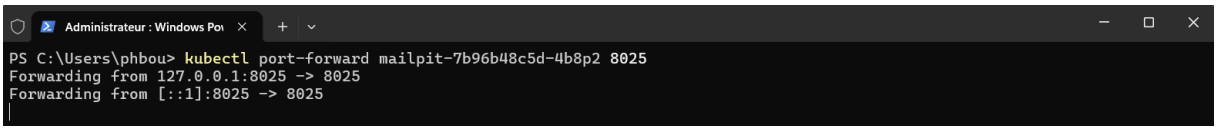
Etat du pod



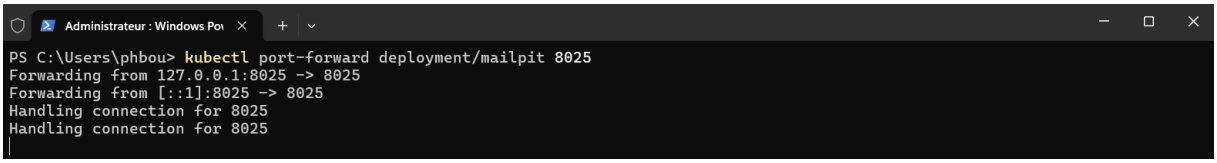
Accès aux logs du pod et des conteneurs

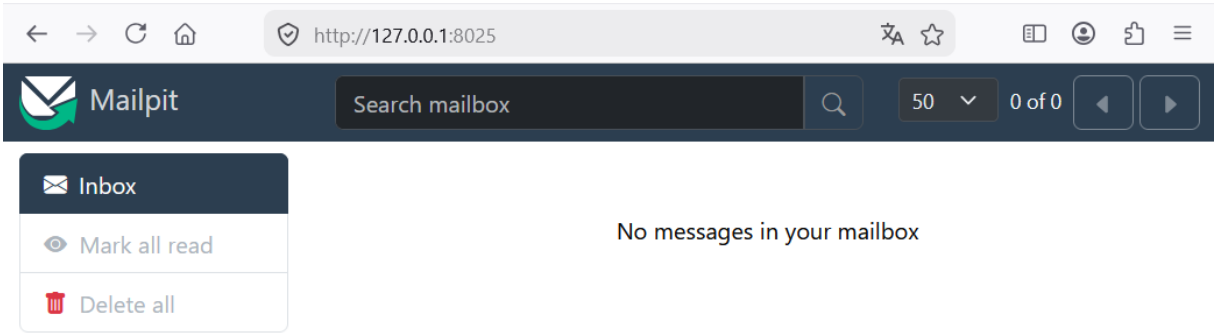


Accès à l'application Mailpit



Ou



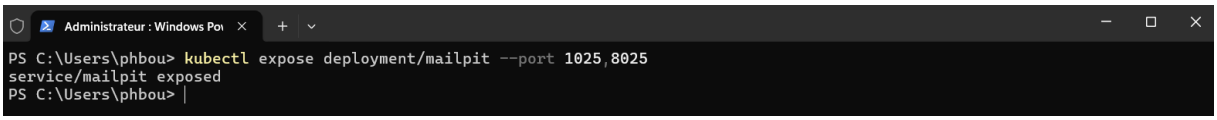


Exposition d'un déploiement via un service (adresse de service) : l'instruction expose

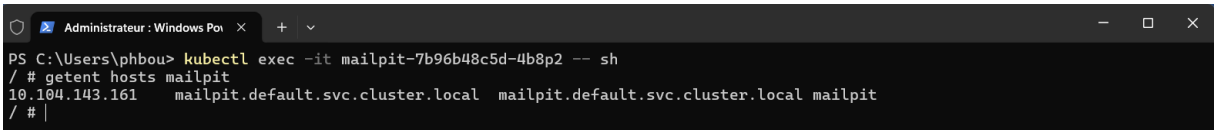
Précédemment, l'accès à l'application Mailpit s'est faite via le nom du pod. Or, le pod peut venir à disparaître et le nom changera lors de la création d'un nouveau pod.

Afin de simplifier l'accès à un service hébergé dans un pod, il sera donc nécessaire de passer par un nom de service.

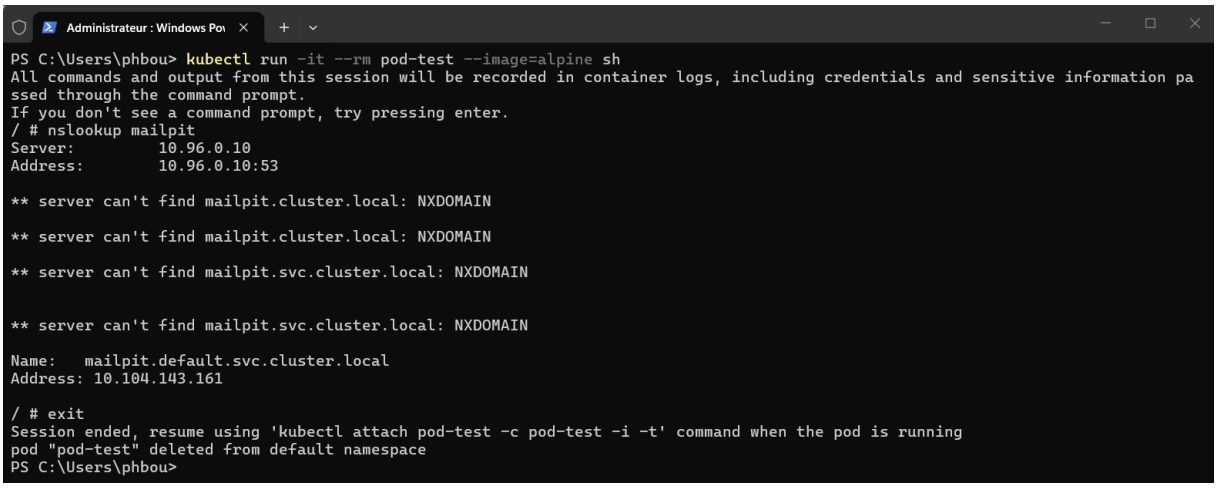
➔ Création d'une entrée DNS stable dans Kubernetes (l'entrée doit pointer sur le pod faisant tourner l'application Mailpit) et MAJ de l'entrée DNS du service en cas de changement de pod sans compter la répartition de charge dans le cas où le déploiement procéderait à la création de plusieurs pods.



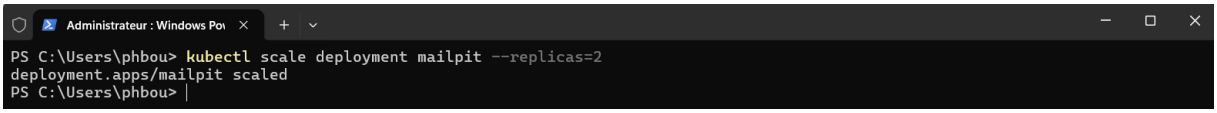
Il n'est pas possible d'interroger le serveur DNS de Kubernetes depuis l'extérieur du cluster. Il faut pour cela lancer une requête DNS depuis le pod de Mailpit en mode interactif.

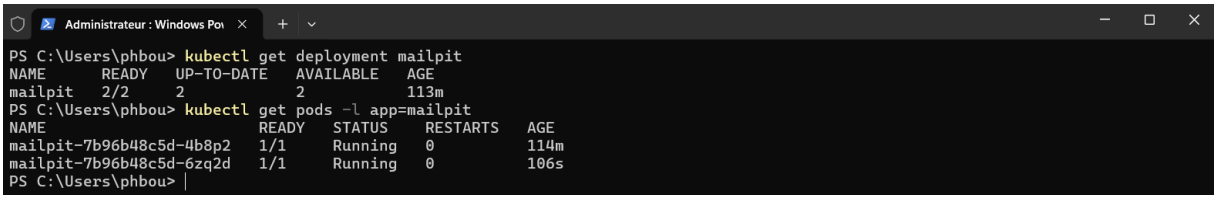


Lancement d'un pod de test :



Résilience et scalabilité (augmenter le niveau de résilience d'une application et augmenter la capacité de trafic d'une application pour répondre à des contraintes de charge)





Ingress et reverse proxy

L'application mailpit n'est pas accessible depuis l'extérieur sauf à utiliser la commande kubectl port-forward. L'objet Ingress a été conçu pour permettre cet accès.

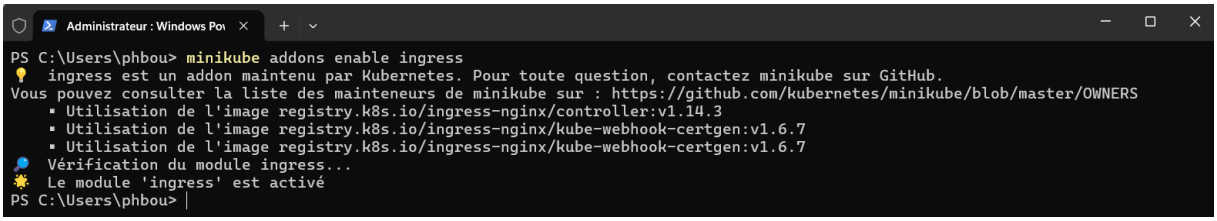
Un proxy inverse est un composant qui se situe en amont d'un autre permettant :

- D'accéder à une application inaccessible directement depuis l'extérieur ;
- De répartir la charge sur plusieurs machines.

Exemple : Haproxy, Nginx.

Minikube dispose d'un contrôleur Ingress s'appuyant sur Nginx qu'il suffit d'activer.

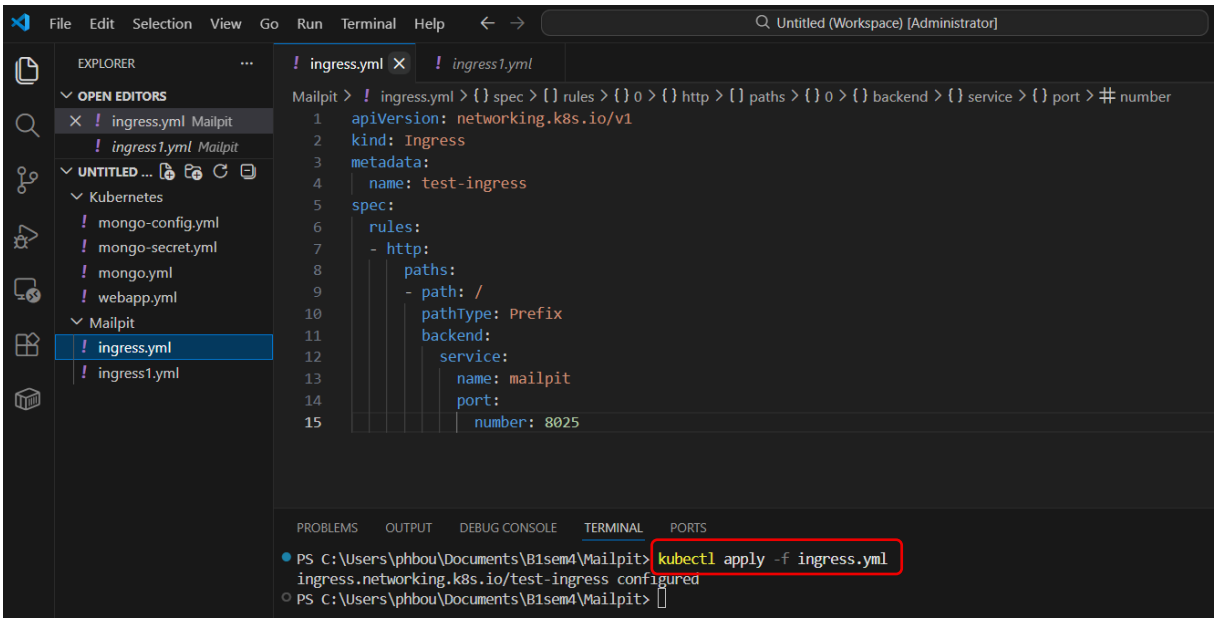
Activation du contrôleur Ingress dans Minikube :



```
PS C:\Users\phbou> kubectl get namespace
NAME                   STATUS   AGE
default                Active   4d6h
ingress-nginx          Active   4m22s
kube-node-lease        Active   4d6h
kube-public            Active   4d6h
kube-system            Active   4d6h
kubernetes-dashboard   Active   47h
PS C:\Users\phbou>
```

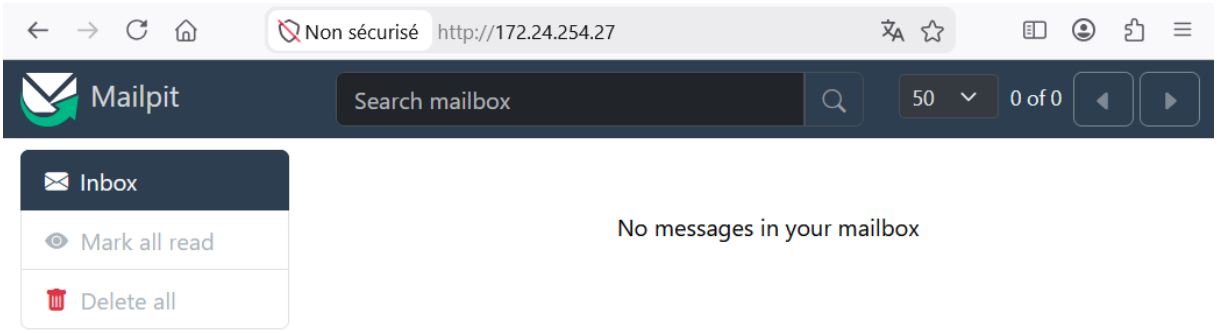
```
PS C:\Users\phbou> kubectl -n ingress-nginx get pods -l app.kubernetes.io/name
NAME                                        READY   STATUS      RESTARTS   AGE
ingress-nginx-admission-create-c6z74        0/1     Completed   0          5m52s
ingress-nginx-admission-patch-qbbmw         0/1     Completed   0          5m52s
ingress-nginx-controller-596f8778bc-gff7g   1/1     Running     0          5m52s
PS C:\Users\phbou>
```

Déclaration d'une règle Ingress et prise en compte du fichier YAML :




```
PS C:\Users\phbou> kubectl get ingress
NAME           CLASS   HOSTS   ADDRESS         PORTS   AGE
test-ingress   nginx   *       172.24.254.27   80      6m20s
PS C:\Users\phbou>
```

```
PS C:\Users\phbou> minikube ip
172.24.254.27
PS C:\Users\phbou>
```



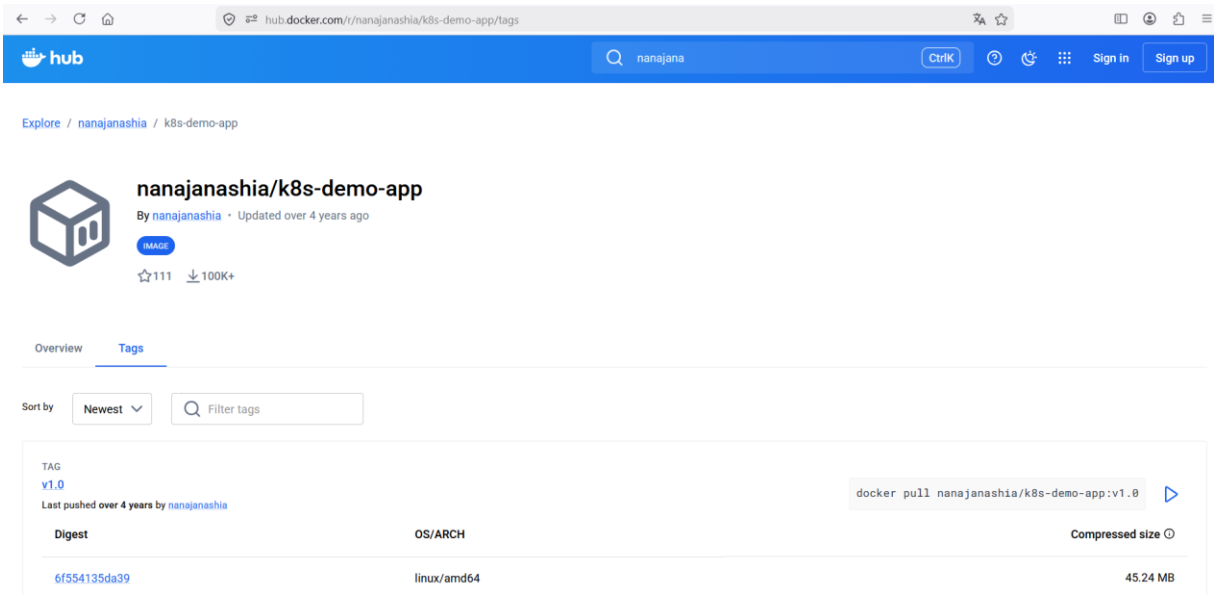
5. Automatisation de déploiement par fichier YAML.

Déploiement Nana :

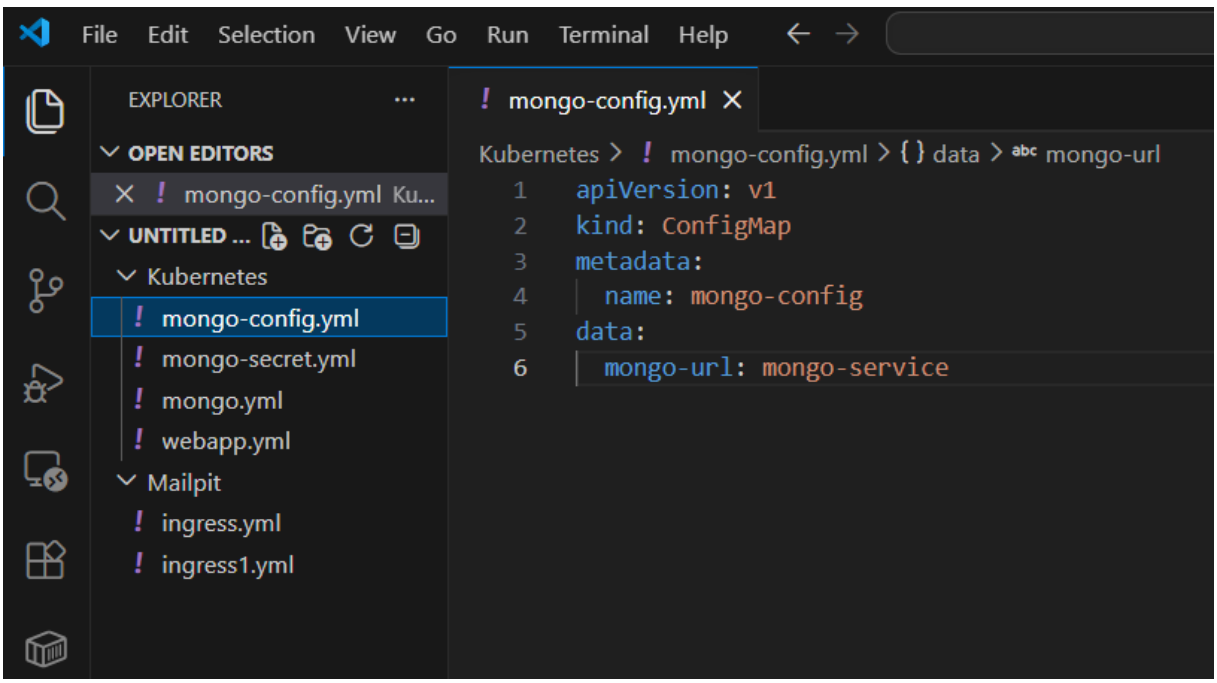
Précédemment, l'application Mailpit a été déployée avec le dashboard de kubernetes puis à l'aide de la commande kubectl avec l'option create.

Cette fois-ci la prise en compte de la création des objets sera faite à l'aide de la commande kubectl avec l'option apply -f suivi du nom du fichier YAML comme vu avec le fichier établi pour la création de l'objet Ingress.

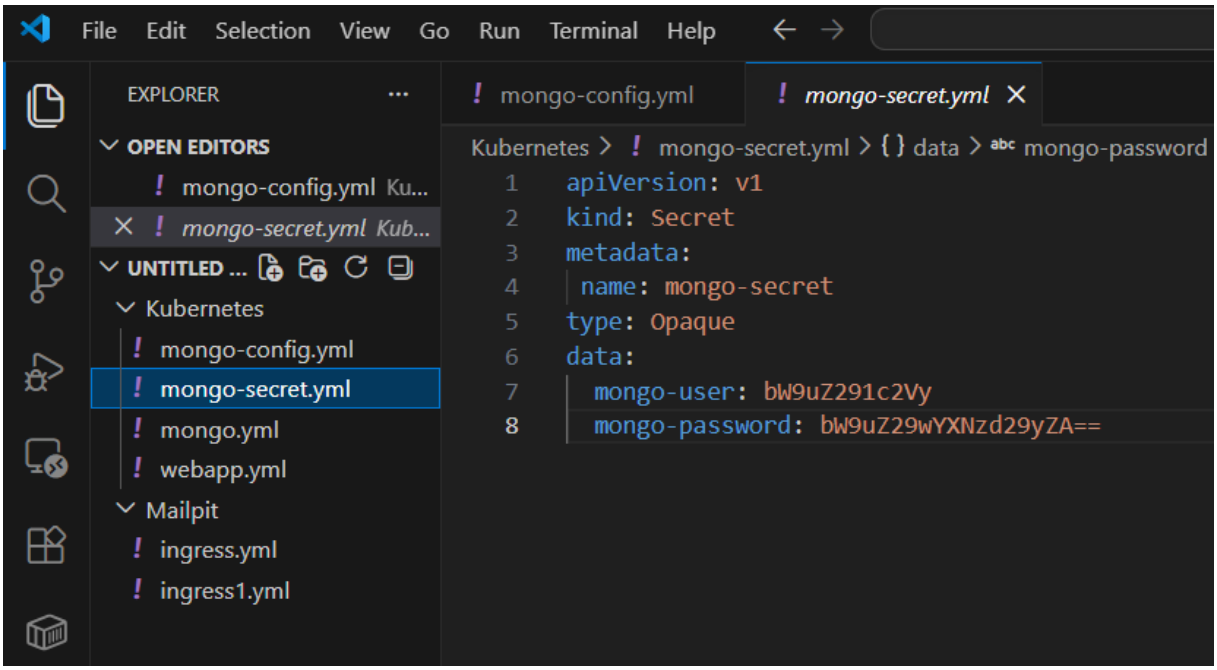
https://www.youtube.com/watch?v=s_o8dwzRlu4



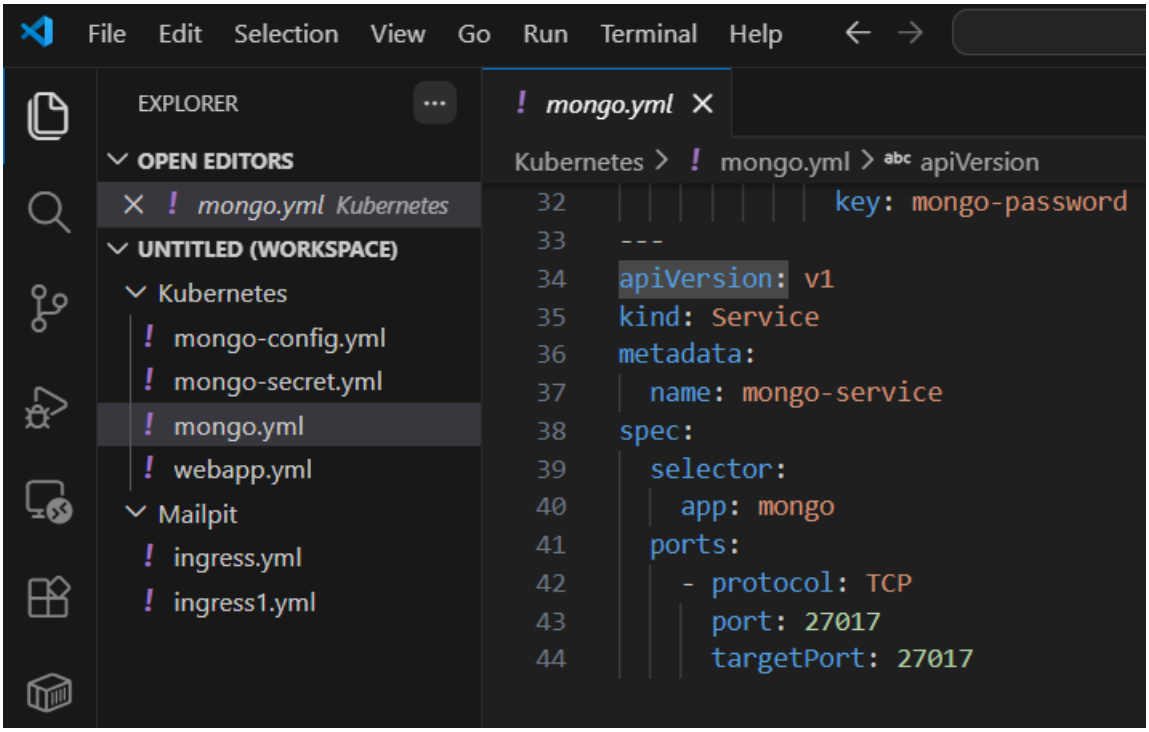
Composant de type ConfigMap :



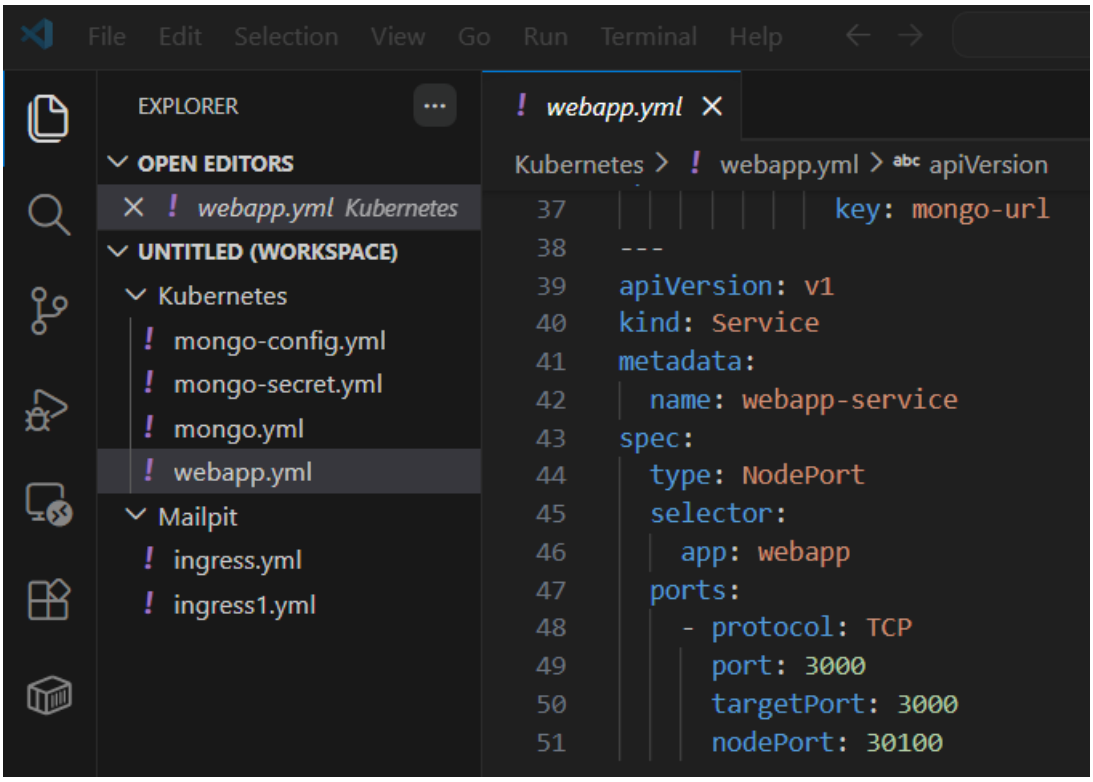
Composant de type Secret :



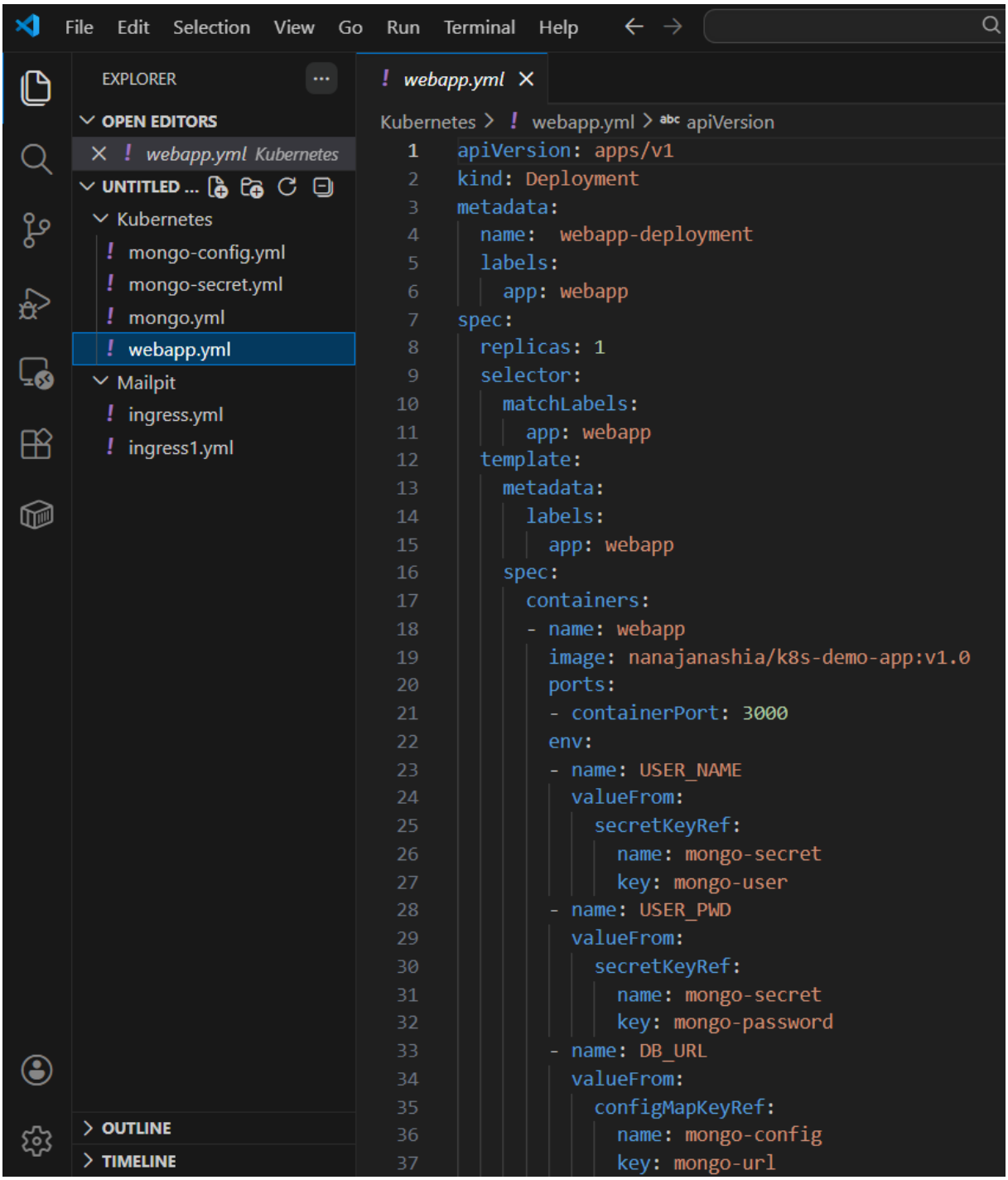
Créer un service interne : service qui va permettre la communication entre le pod front et le pod de la base de données



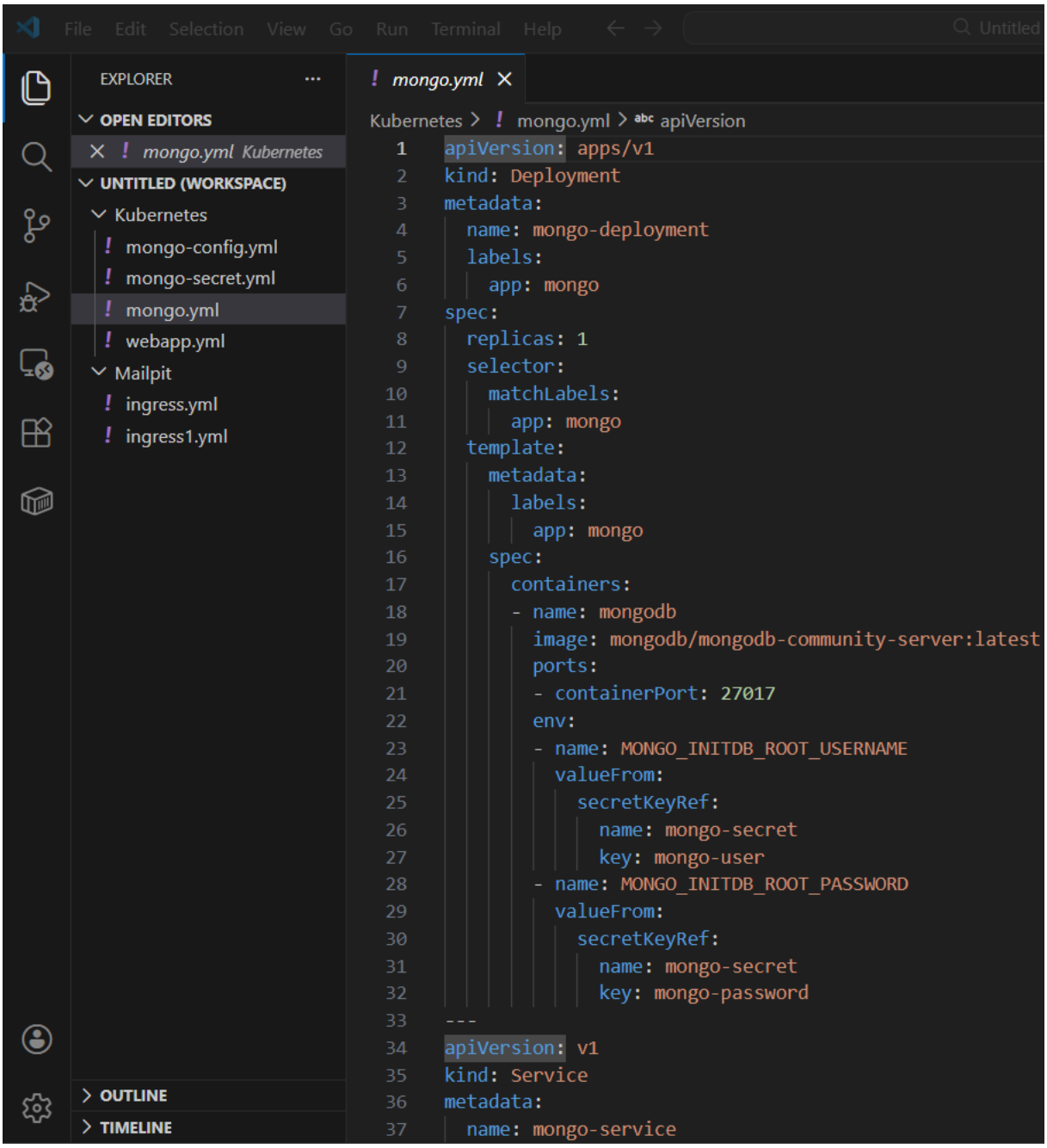
Créer un service externe : pod du front accessible depuis l'extérieur du cluster Kubernetes



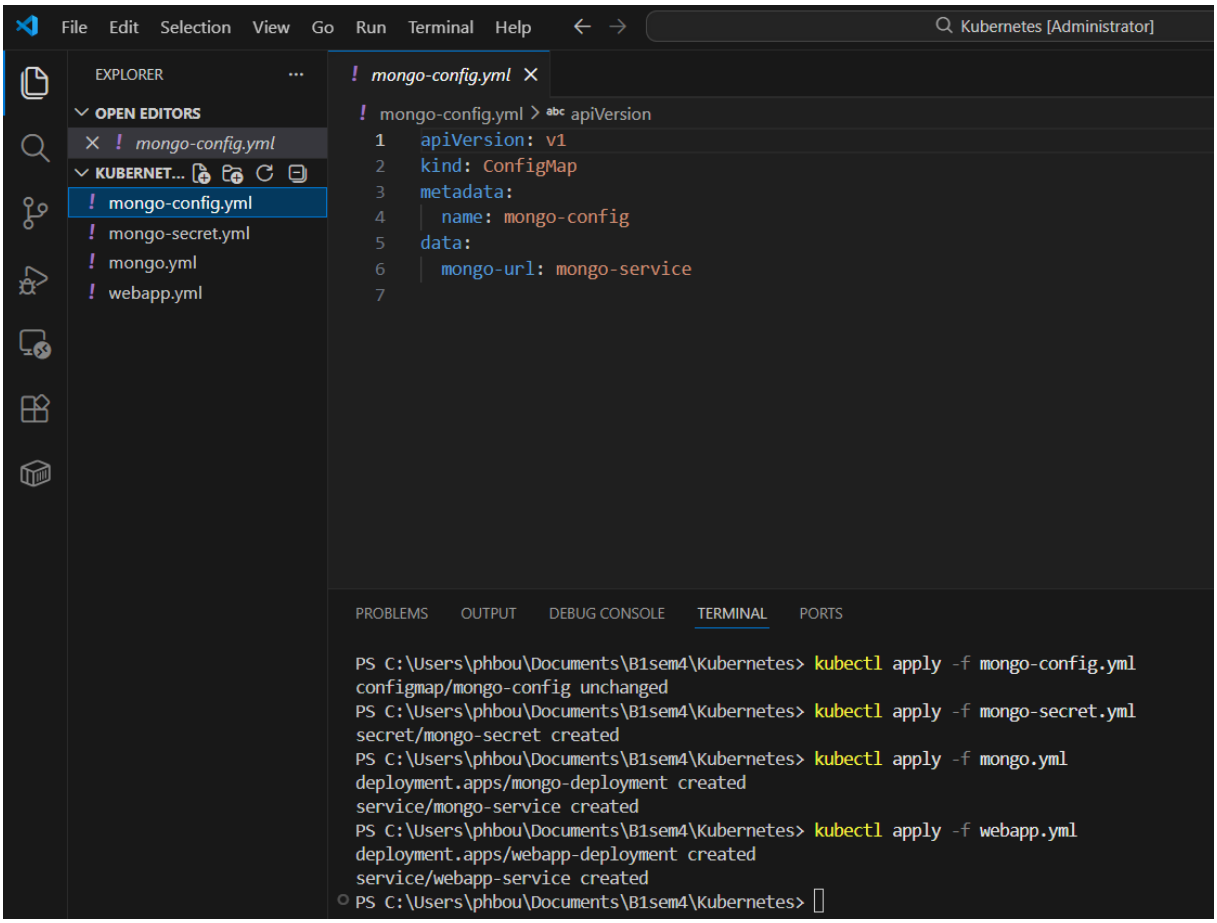
Déployer l'application du pod front (webapp) : créer un composant de type deployment



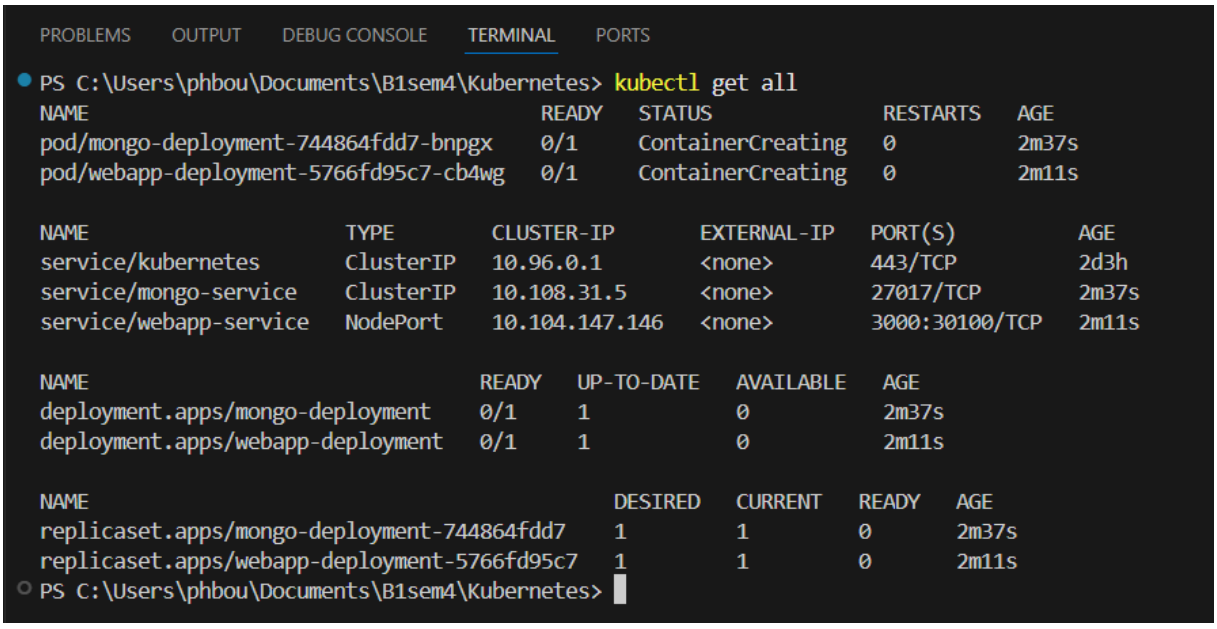
Déployer l'application mongo :



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



Lister les objets créés :



Lister les autres objets (configMap et Secret) :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl get configmap

NAME               DATA   AGE
kube-root-ca.crt   1      2d3h
mongo-config       1      25m
PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl get secret
NAME           TYPE     DATA   AGE
mongo-secret   Opaque   2      6m57s
PS C:\Users\phbou\Documents\B1sem4\Kubernetes>
```

Décrire le service webapp-service :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl describe service webapp-service
Name:                     webapp-service
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 app=webapp
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.104.147.146
IPs:                      10.104.147.146
Port:                     <unset>  3000/TCP
TargetPort:               3000/TCP
NodePort:                 <unset>  30100/TCP
Endpoints:                10.244.0.8:3000
Session Affinity:         None
External Traffic Policy:  Cluster
Internal Traffic Policy:  Cluster
Events:                   <none>
PS C:\Users\phbou\Documents\B1sem4\Kubernetes>
```

Décrire le pod associé :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl describe pod webapp-deployment-5766fd95c7-cb4wg
Name:             webapp-deployment-5766fd95c7-cb4wg
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/172.24.254.27
Start Time:       Sun, 26 Apr 2026 15:26:57 +0200
Labels:           app=webapp
                  pod-template-hash=5766fd95c7
Annotations:      <none>
Status:           Running
IP:               10.244.0.8
IPs:
  IP:           10.244.0.8
Controlled By:  ReplicaSet/webapp-deployment-5766fd95c7
Containers:
  webapp:
    Container ID:   docker://fe054eab81a00e84b465e70a315993f1bd5017285034a662a92d833e1588724c
    Image:          nanajanashia/k8s-demo-app:v1.0
    Image ID:       docker-pullable://nanajanashia/k8s-demo-app@sha256:6f554135da39ac00a1c2f43e44c2b0b54ca13d3d8044da969361e7781adb7f95
    Port:           3000/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Sun, 26 Apr 2026 15:31:53 +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-9nwkk (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-9nwkk:
```

Consulter les logs de ce pod :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl get pod
NAME                                 READY   STATUS    RESTARTS   AGE
mongo-deployment-744864fdd7-bnpgx    1/1     Running   0          16m
webapp-deployment-5766fd95c7-cb4wg   1/1     Running   0          15m
PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl logs webapp-deployment-5766fd95c7-cb4wg
app listening on port 3000!
PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl logs webapp-deployment-5766fd95c7-cb4wg -f
app listening on port 3000!
```

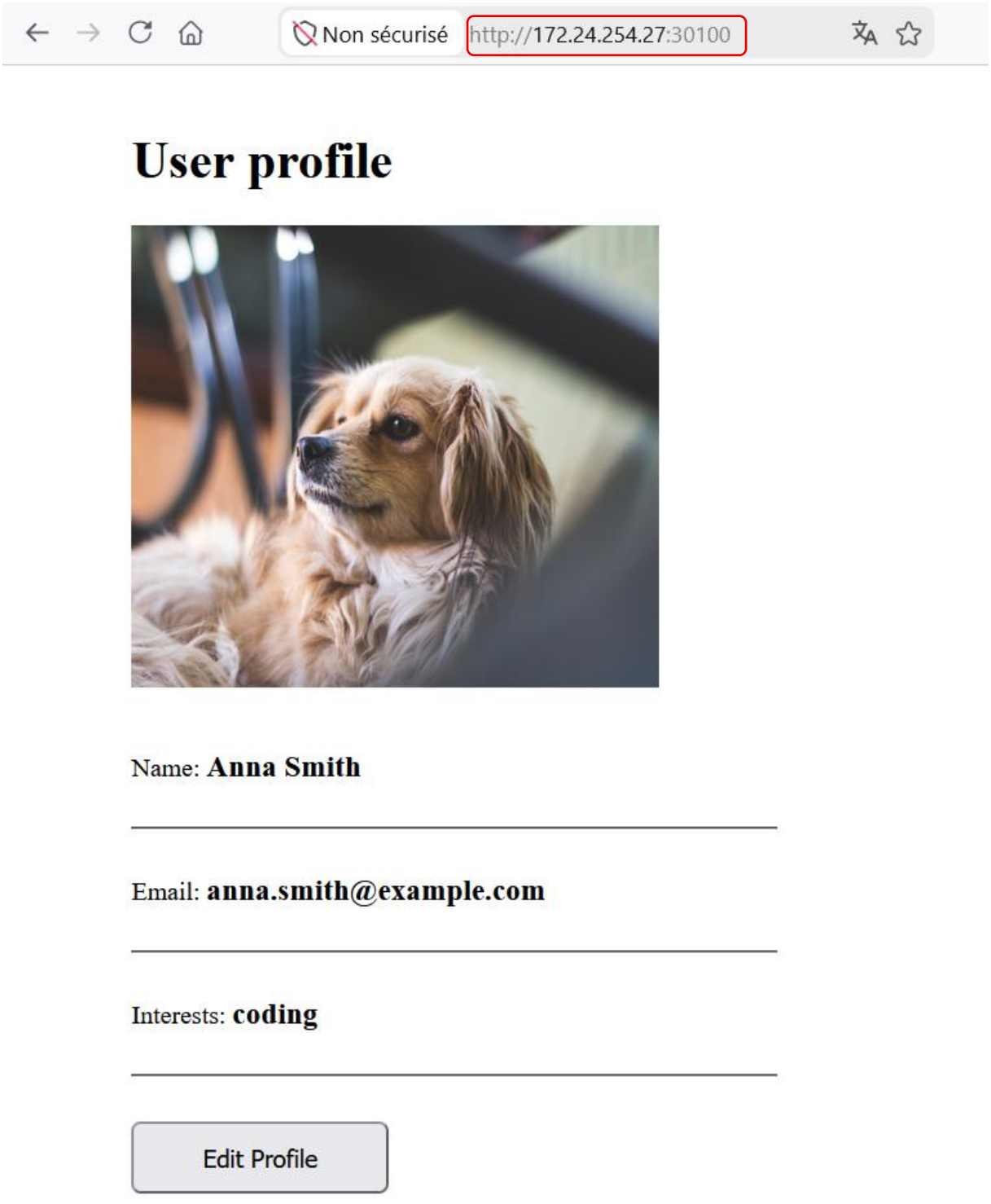
Lister les services :

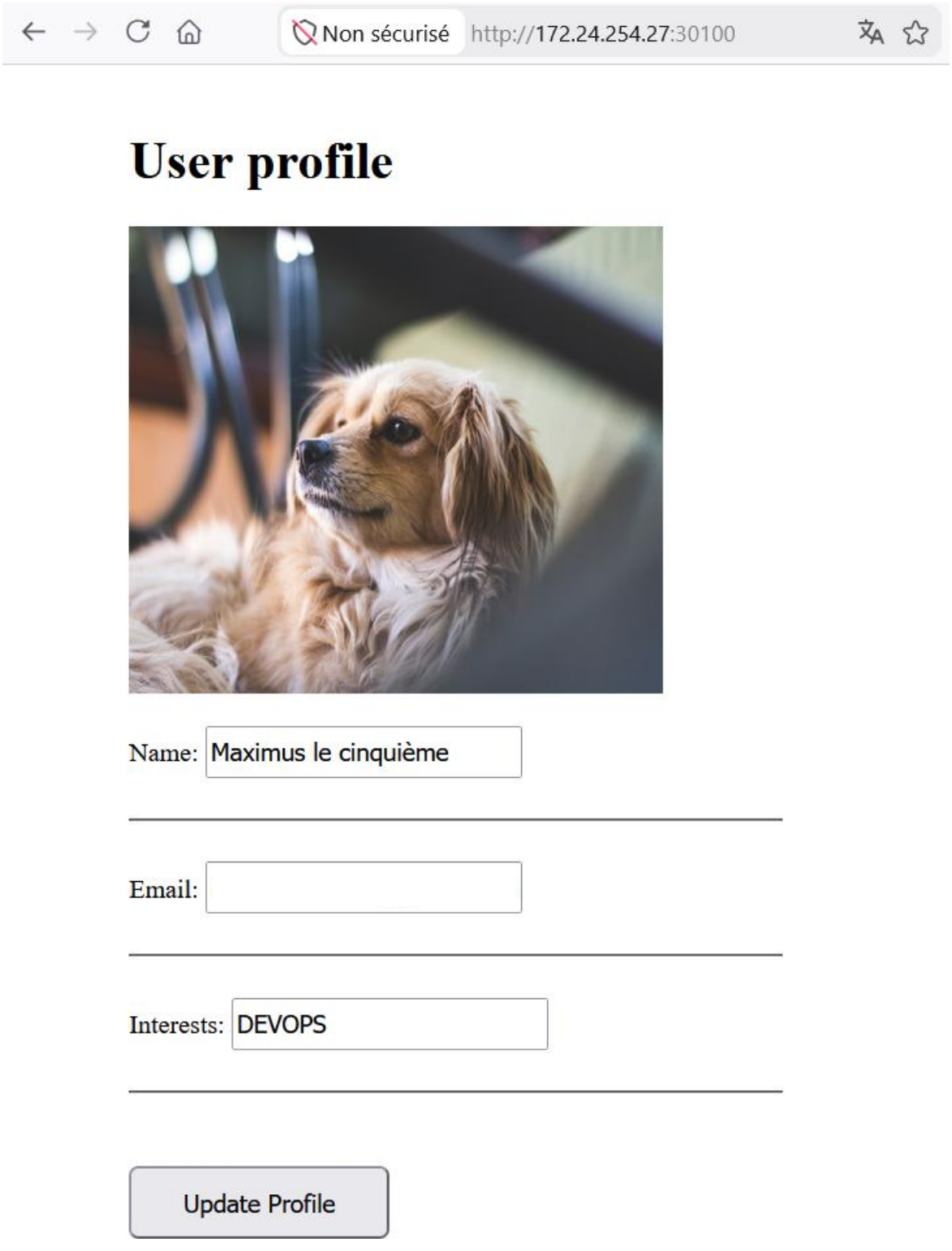
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl get service
NAME              TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes        ClusterIP   10.96.0.1        <none>        443/TCP          2d3h
mongo-service     ClusterIP   10.108.31.5      <none>        27017/TCP        17m
webapp-service    NodePort    10.104.147.146   <none>        3000:30100/TCP   17m
PS C:\Users\phbou\Documents\B1sem4\Kubernetes>
```

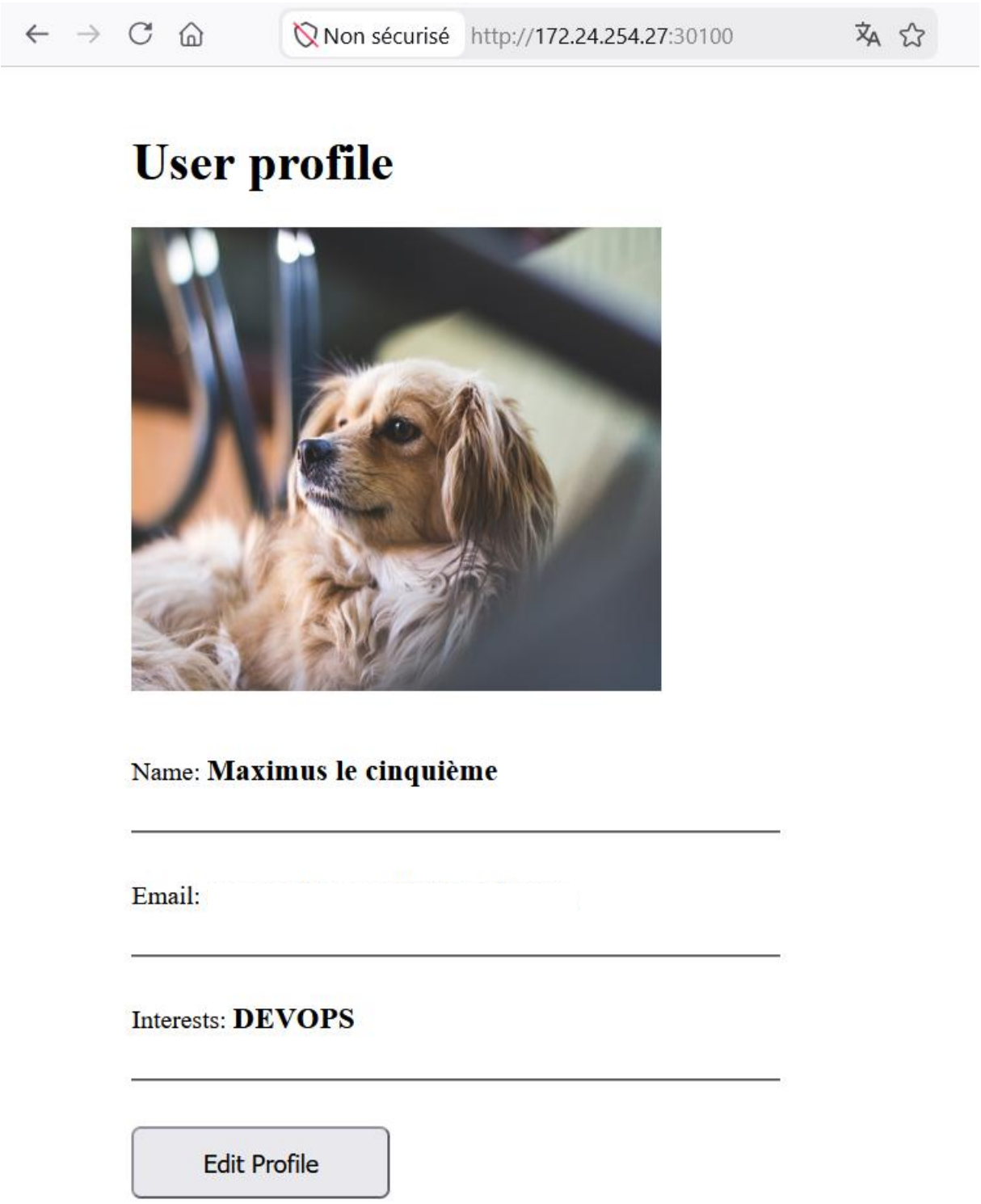
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\phbou\Documents\B1sem4\Kubernetes> minikube ip
172.24.254.27
PS C:\Users\phbou\Documents\B1sem4\Kubernetes>
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl get node
NAME       STATUS   ROLES           AGE    VERSION
minikube   Ready    control-plane   2d3h   v1.35.1
PS C:\Users\phbou\Documents\B1sem4\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   2d3h   v1.35.1   172.24.254.27   <none>        Buildroot 2025.02   6.6.95           docker://28.5.2
PS C:\Users\phbou\Documents\B1sem4\Kubernetes>
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl get svc -o wide
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE    SELECTOR
kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP          2d3h   <none>
mongo-service    ClusterIP   10.108.31.5      <none>        27017/TCP        23m    app=mongo
webapp-service   NodePort    10.104.147.146   <none>        3000:30100/TCP   22m    app=webapp
PS C:\Users\phbou\Documents\B1sem4\Kubernetes> kubectl get pod -o wide
NAME                                 READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
mongo-deployment-744864fdd7-bnpgx    1/1     Running   0          24m   10.244.0.7   minikube   <none>           <none>
webapp-deployment-5766fd95c7-cb4wg   1/1     Running   0          23m   10.244.0.8   minikube   <none>           <none>
PS C:\Users\phbou\Documents\B1sem4\Kubernetes>
```


Non sécurisé
http://172.24.254.27:30100
User profile
Name: Anna Smith
Email: anna.smith@example.com
Interests: coding
Edit Profile


Non sécurisé
http://172.24.254.27:30100
User profile
Name: Maximus le cinquième
Email:
Interests: DEVOPS
Update Profile



6. Application.

https://kubernetes.io/docs/tutorials/stateful-application/mysql-wordpress-persistent-volume/

https://www.juliafmorgado.com/fr/posts/d%C3%A9ploiement-de-wordpress-et-mysql-sur-kubernetes-avec-kind-guide-%C3%A9tape-par-%C3%A9tape/

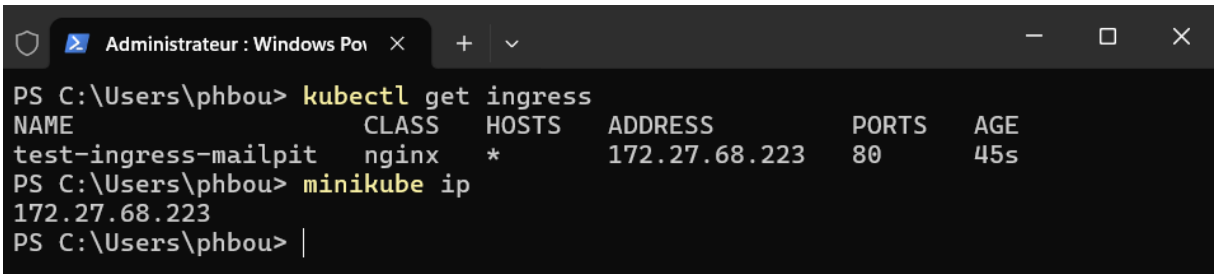
https://medium.com/@ogo.ulrich/d%C3%A9ploiement-de-wordpress-avec-mysql-sur-kubernetes-46d6e1e7755c

https://medium.com/@dipakrasal2009/deploying-wordpress-on-kubernetes-minikube-using-wordpress-and-mysql-docker-images-21b170972b34

➔ Déployer un blog WordPress et sa base de données MySQL en utilisant Minikube. Automatiser le déploiement avec des fichiers YAML et utiliser pour cela les objets PersistentVolumes et PersistentVolumeClaims afin d'être indépendant du cycle de vie des Pod et ainsi préserver les données en cas de redémarrage ou de suppression de Pods.

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

- Récupérer les informations sur les objets ingress :



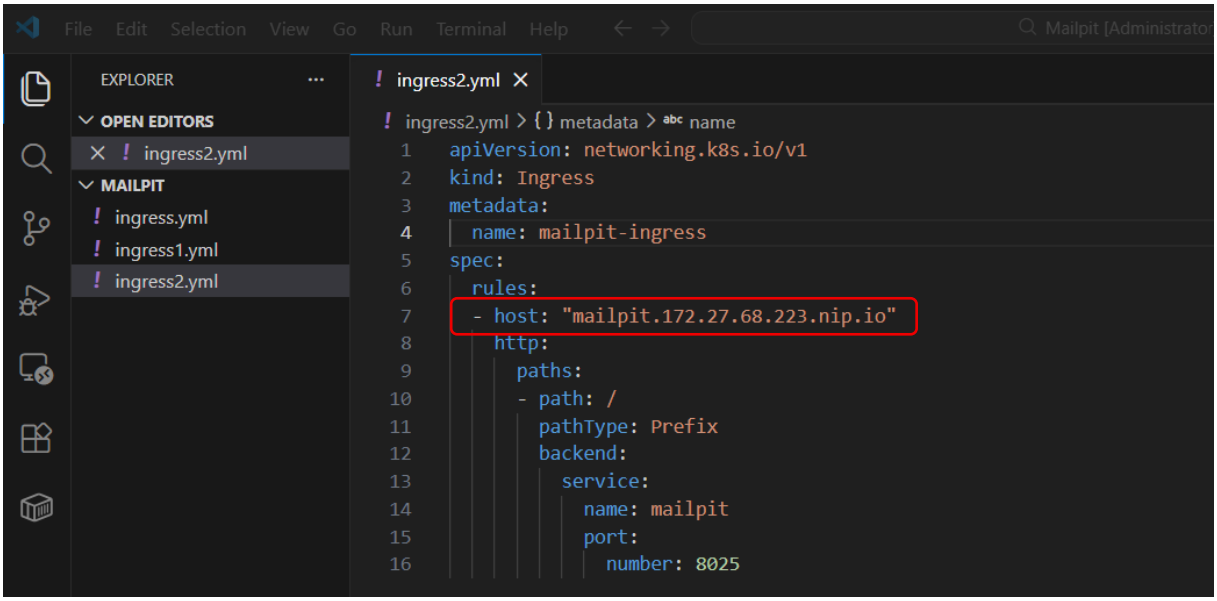
L'étoile indique que le contrôleur utilise Mailpit par défaut pour toutes les requêtes entrantes : tous les accès sont renvoyés vers Mailpit.

Dans une règle Ingress, le champ host permet de spécifier le nom de l'hôte virtuel. Il faut pour cela disposer d'un nom de domaine et d'une entrée DNS (mailpit.sio-exupery.fr par exemple).

Dans notre cas, nous ne disposons que d'une adresse IP.

➔ Nous pouvons utiliser le mécanisme nip.io qui utilise le domaine 172.27.68.223.nip.io et tous ses sous-domaines (mailpit.172.27.68.223.nip.io par exemple).

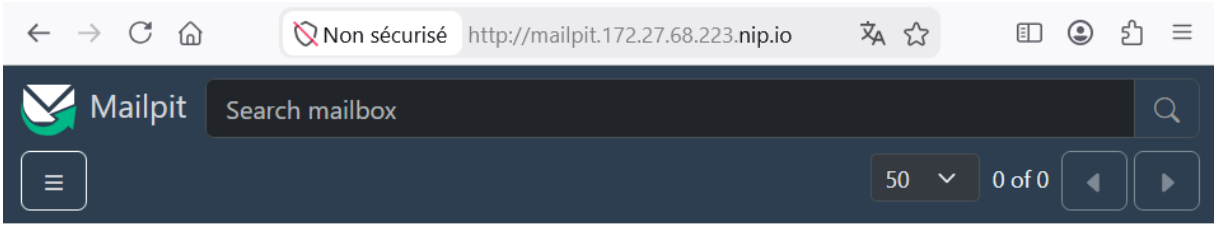
- Création d'un hôte virtuel pour Mailpit : modification du contenu du fichier ingress.yaml afin de pointer la règle Ingress sur l'entrée DNS figurant ci-dessus :



- Prise en compte de la modification :

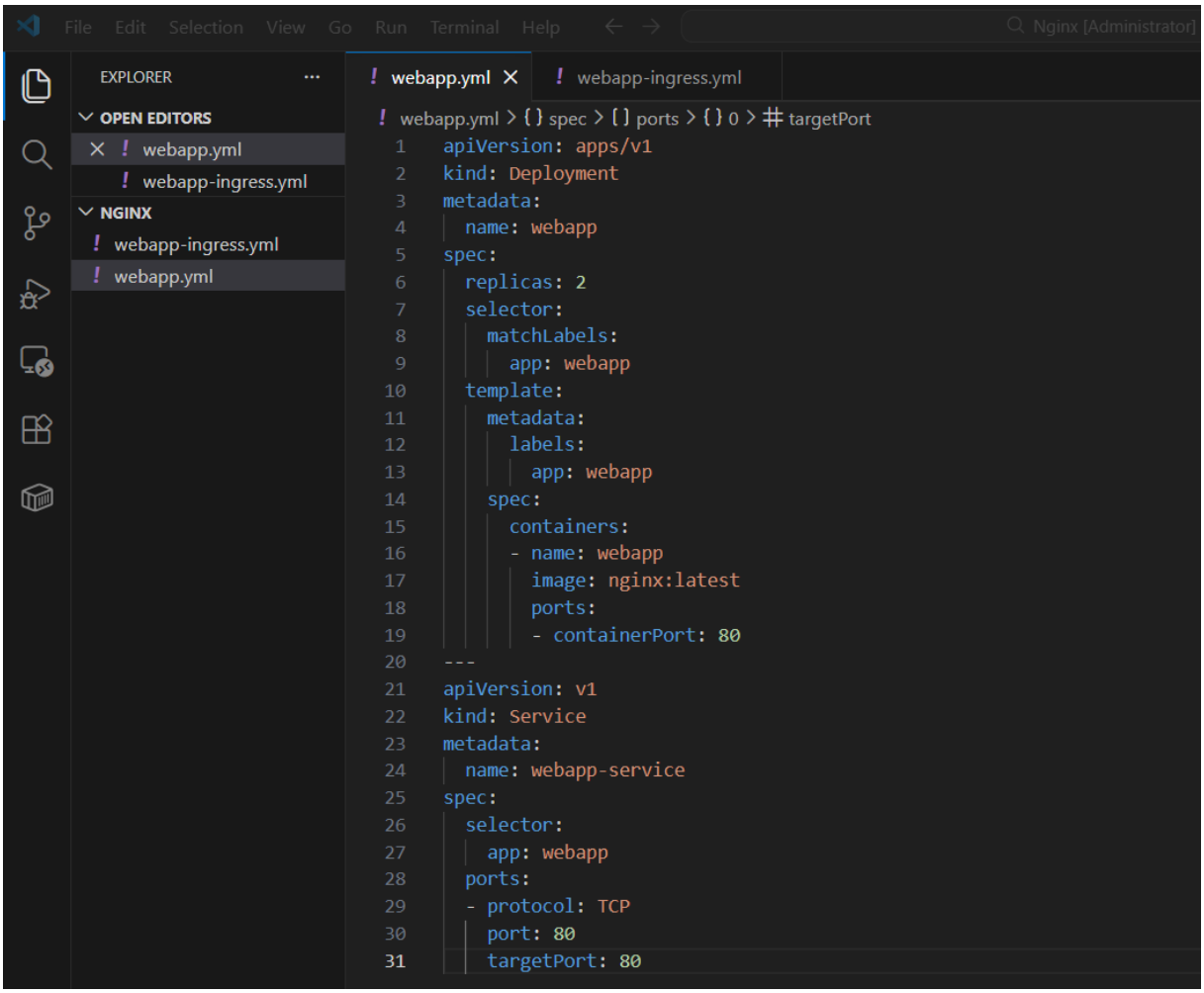
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\phbou\Documents\B1sem4\Mailpit> kubectl apply -f ingress2.yml
ingress.networking.k8s.io/mailpit-ingress created
PS C:\Users\phbou\Documents\B1sem4\Mailpi
```

- Tester le mécanisme de publication reposant sur un hôte virtuel :

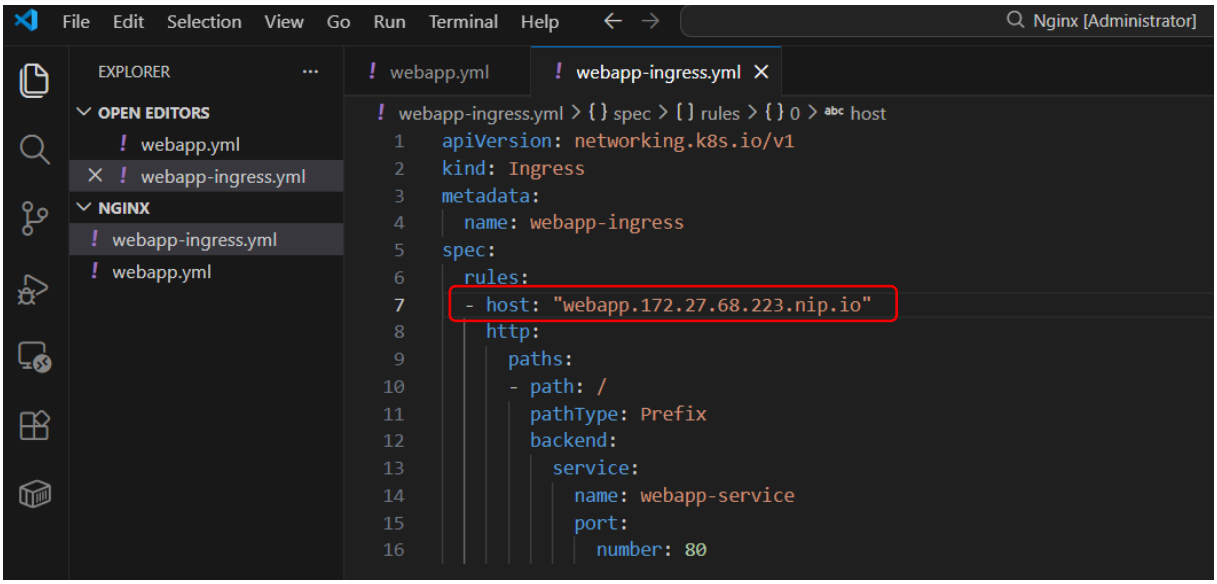


8. Hôtes virtuels et nom de domaine nip.io : autre exemple

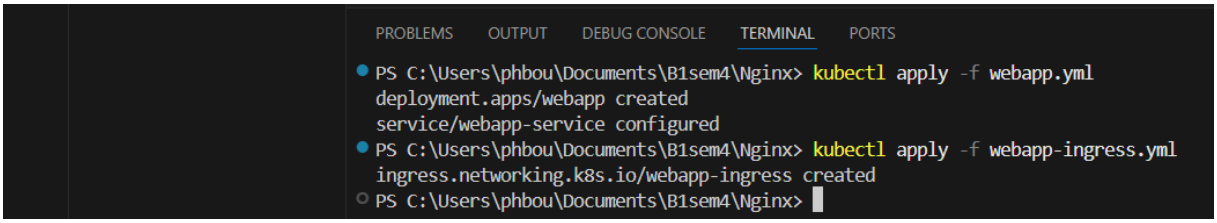
- Créer un objet Deployment et un objet Service :

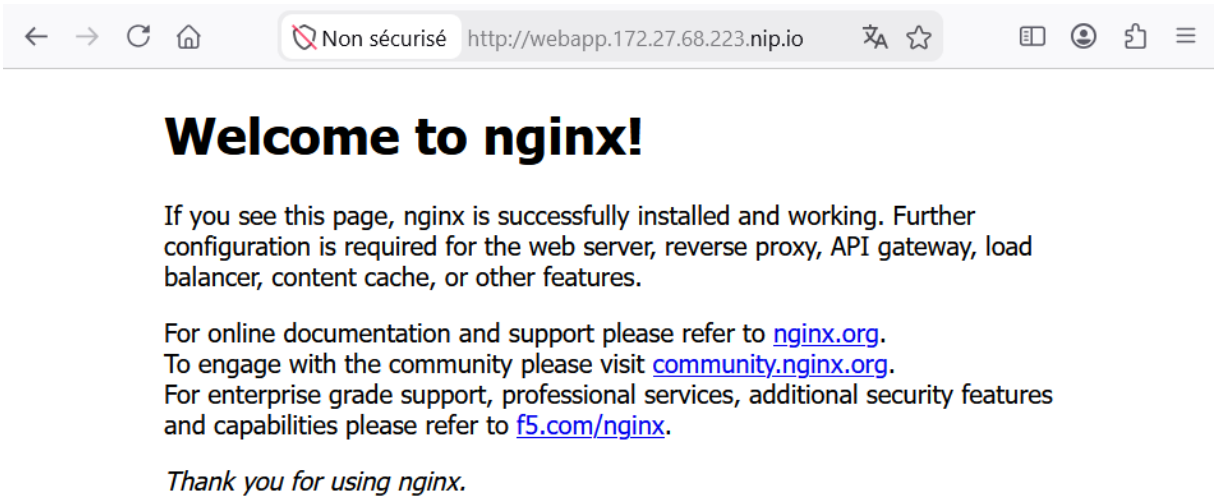


- Créer un objet Ingress avec le champ host spécifiant le nom d'hôte virtuel webapp.172.27.68.223.nip.io :



- Prise en compte de 2 fichiers YAML et test :

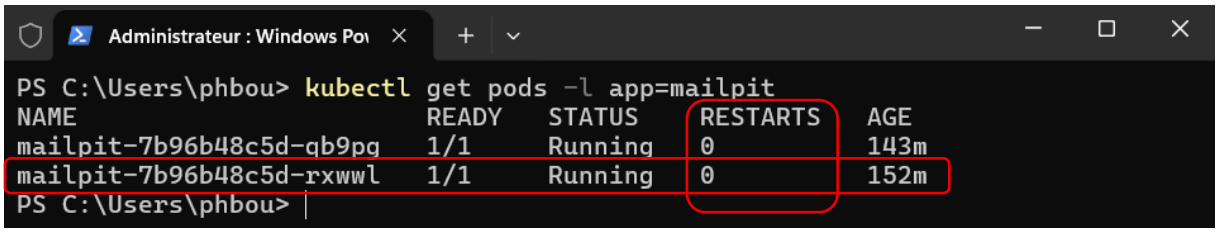




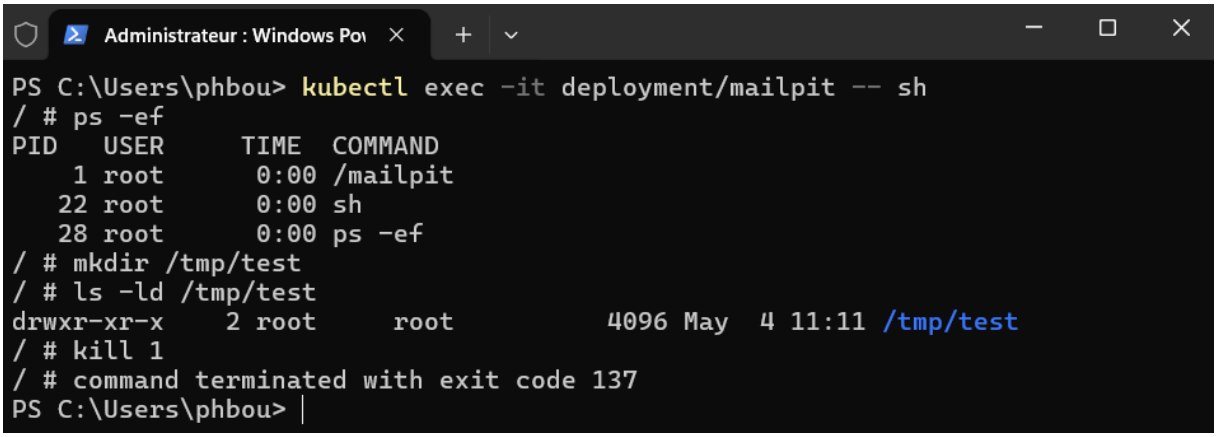
➔ Nous aurions pu créer un seul objet Ingress routant plusieurs domaines vers différents Services.

9. Cycle de vie d'un conteneur dans Kubernetes

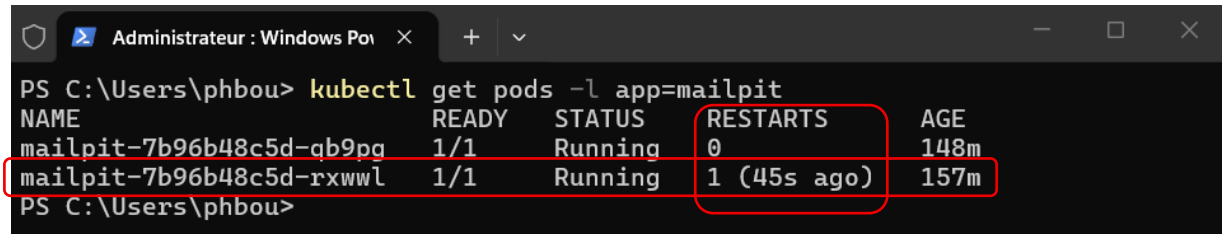
- Consulter l'état des pods de l'application portant le labell app=mailpit :



- Connexion au pod (indiquer le nom du pod ou le nom du déploiement). Une fois dans le contexte du conteneur de Mailpit, lister les processus, créer un répertoire, lister le répertoire (et non pas son contenu) puis tuer le processus Mailpit :

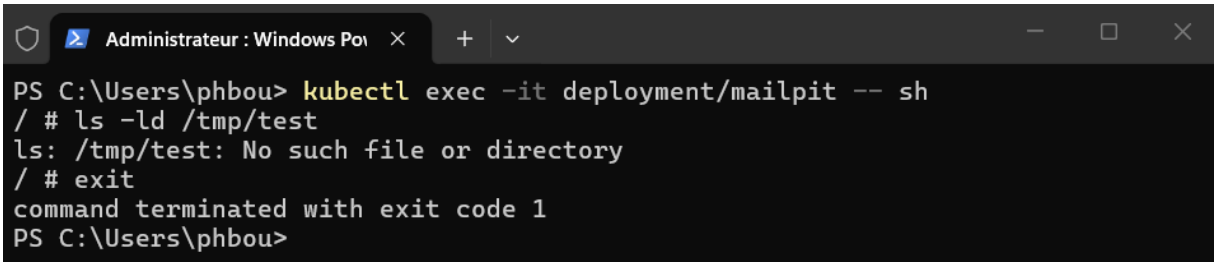


- Consulter l'état des pods de l'application :



Le pod a redémarré automatiquement. Il s'agit d'un nouveau pod. Le conteneur n'est plus le même.

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



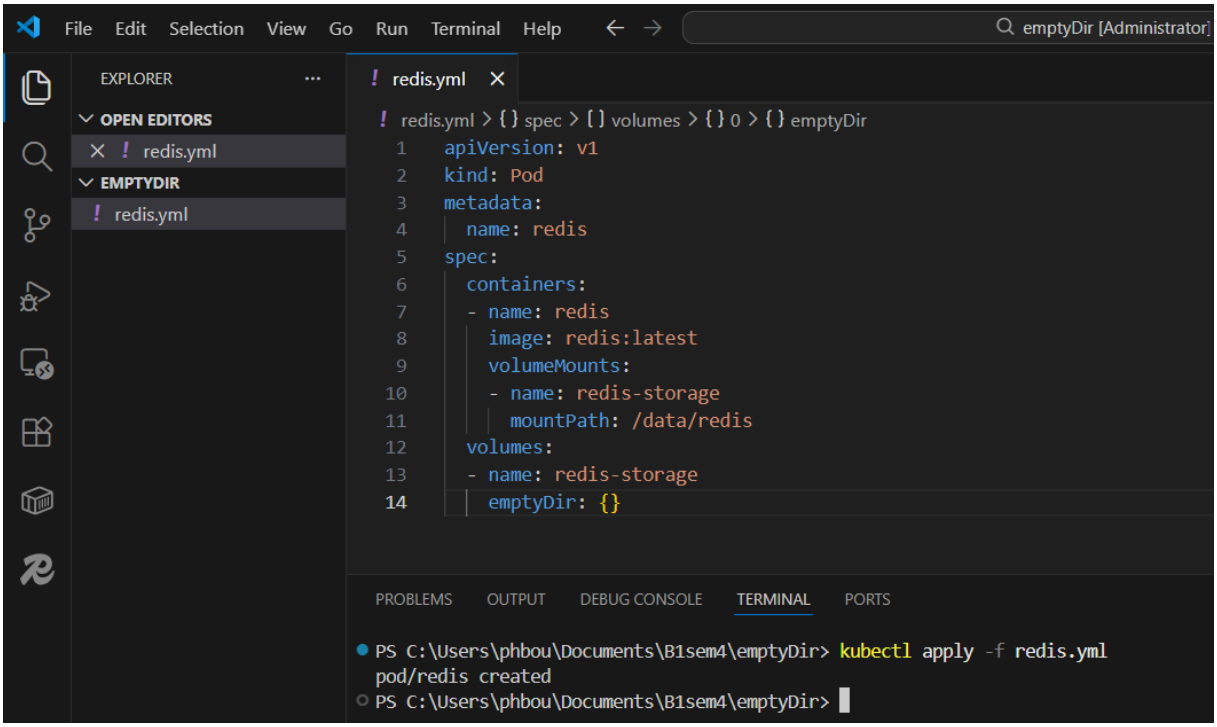
10. Persistance des données

1) Volume de type emptyDir

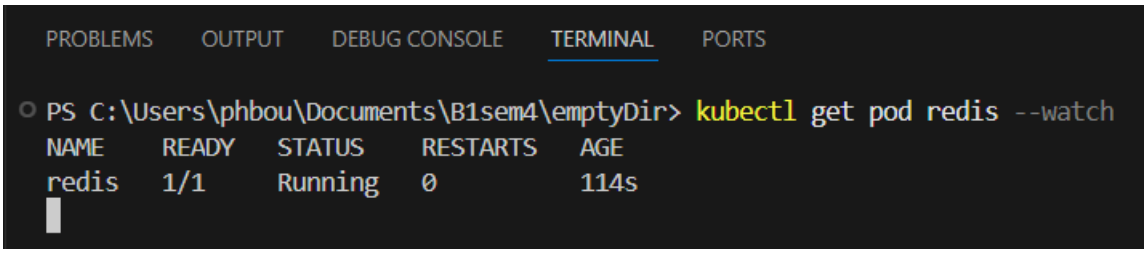
Un Volume de type emptyDir est un volume temporaire liée au cycle de vie du Pod. Il dure toute la vie du Pod même si le conteneur s'arrête et redémarre.

Exemple avec Redis :

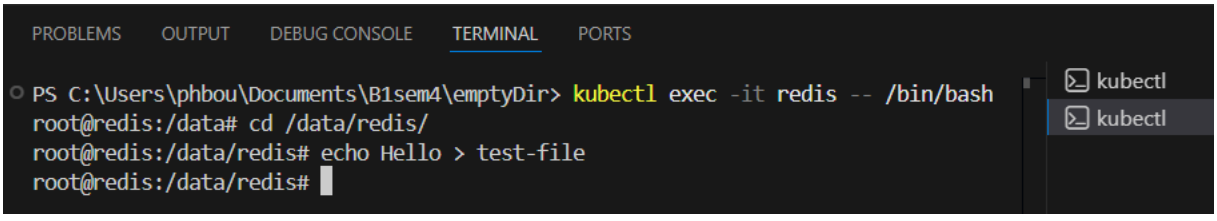
- Créer le fichier de configuration du Pod redis.yaml et créer le pod avec la commande kubectl apply -f :



- Vérifier que le conteneur du pod soit en cours d'exécution puis surveiller les modifications apportées au pod :



- Dans un second terminal, accéder à la console shell du conteneur en cours d'exécution puis créer un fichier dans /data/redis/ :



- Installer procps (faire un apt-get update au préalable) puis lister les processus en cours d'exécution :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    kubectl

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 [130 kB]
Fetched 10.0 MB in 8s (1226 kB/s)
Reading package lists... Done
root@redis:/data/redis#
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    kubectl

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
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    kubectl

root@redis:/data/redis# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
redis          1  0.5  0.5 160264 23296 ?        Ssl  16:47   0:03 redis-server *:6379
root          29  0.0  0.0   4320  3584 pts/0    Ss   16:51   0:00 /bin/bash
root          71  0.0  0.0      0     0 pts/0    Z    16:57   0:00 [dpkg-preconfigu] <de
root         161  0.0  0.0   6388  3456 pts/0    R+   16:57   0:00 ps aux
root@redis:/data/redis#
```

- Arrêter le processus Redis :

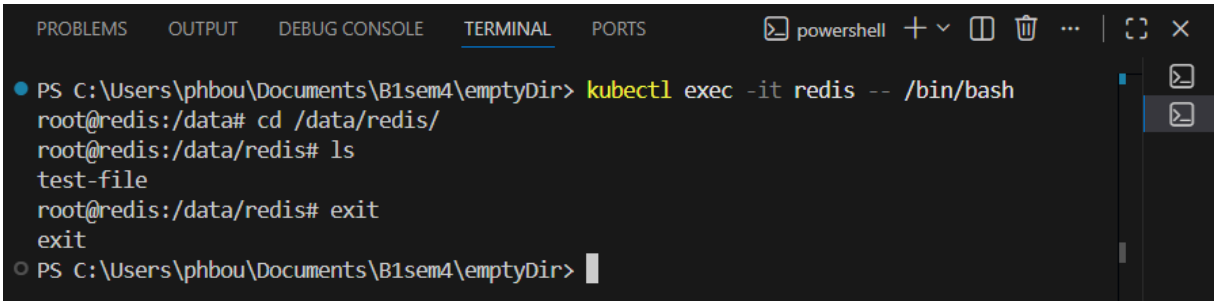
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    powershell

root@redis:/data/redis# kill 1
root@redis:/data/redis# command terminated with exit code 137
PS C:\Users\phbou\Documents\B1sem4\emptyDir>
```

- Revenir dans le premier terminal et constater les changements apportés au Pod de Redis :

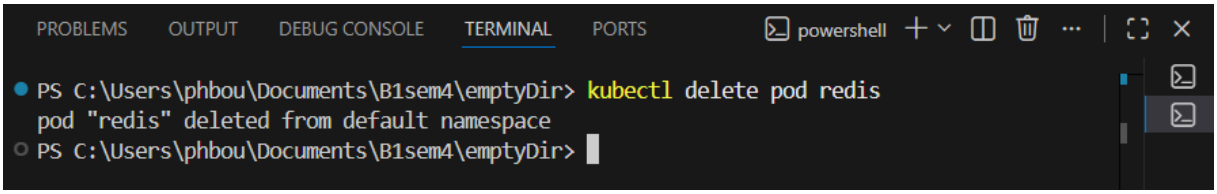
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                    kubectl

PS C:\Users\phbou\Documents\B1sem4\emptyDir> kubectl get pod redis --watch
NAME    READY   STATUS    RESTARTS      AGE
redis   1/1     Running   0             114s
redis   0/1     Completed   0             17m
redis   1/1     Running     1 (3s ago)    17m
```

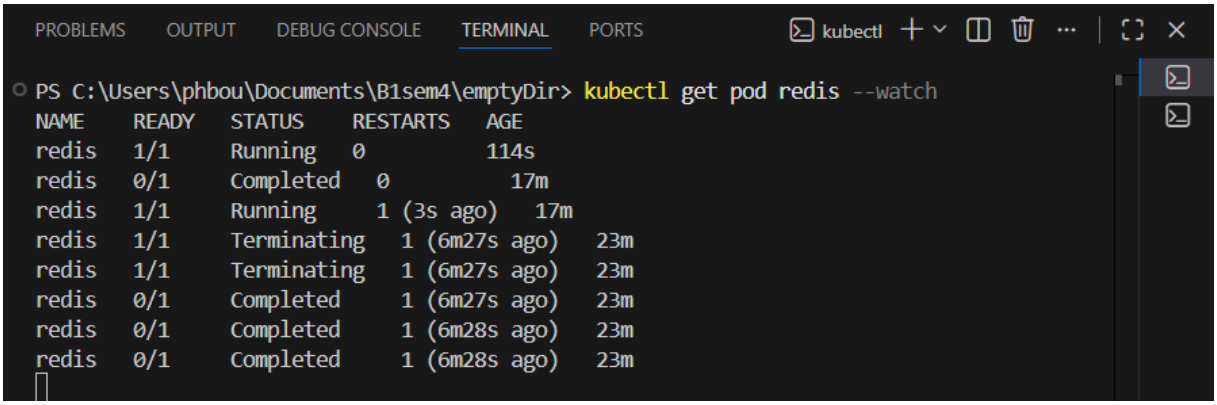
Le conteneur est terminé et redémarré.

- Accéder à la console shell du conteneur redémarré et constater que le fichier test-file est toujours présent dans le répertoire /data/redis :



- Supprimer le pod :





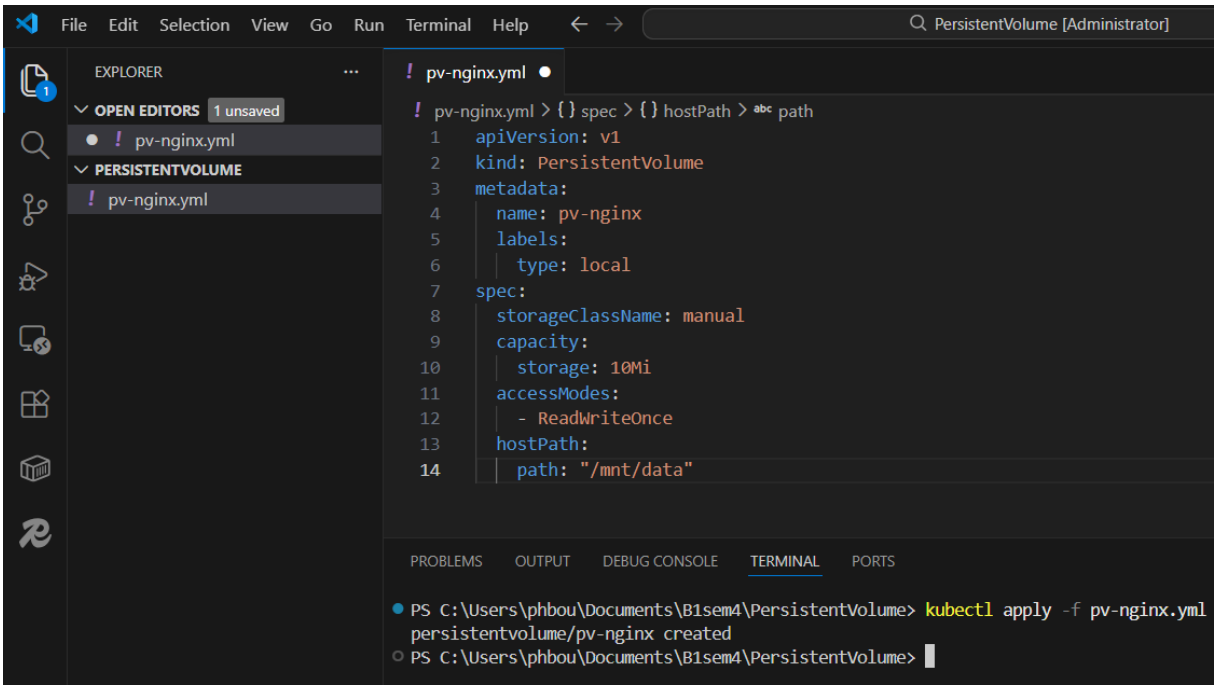
2) Volume de type PV (PersistentVolume) et PVC (PersistentVolumeClaim)

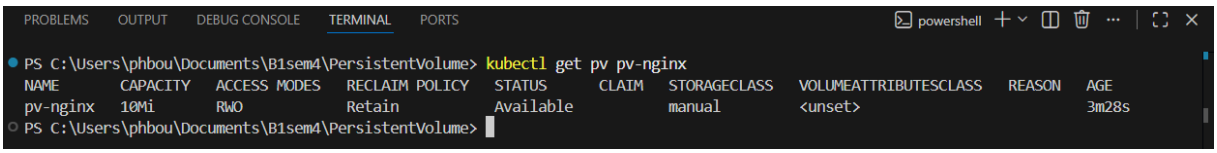
- Accéder à minikube via la commande minikube ssh puis créer un dossier /mnt/data. Dans le dossier /mnt/data, créer un fichier index.html. Tester l'existence du fichier index.html puis fermer la session shell sur le nœud du cluster.

- Créer un PersistentVolume de type hostpath

Un volume de type hostPath est utilisé pour le développement et les tests sur un cluster à noeud unique. Dans notre cas, le PersistentVolume de type hostPath va utiliser un dossier sur le nœud minikube pour simuler un stockage réseau.

Dans un cluster de production, vous n'utiliseriez pas le type *hostPath*. Un administrateur de cluster utiliserait un disque persistant Google Compute Engine, un partage NFS ou un volume Amazon Elastic Block Store.

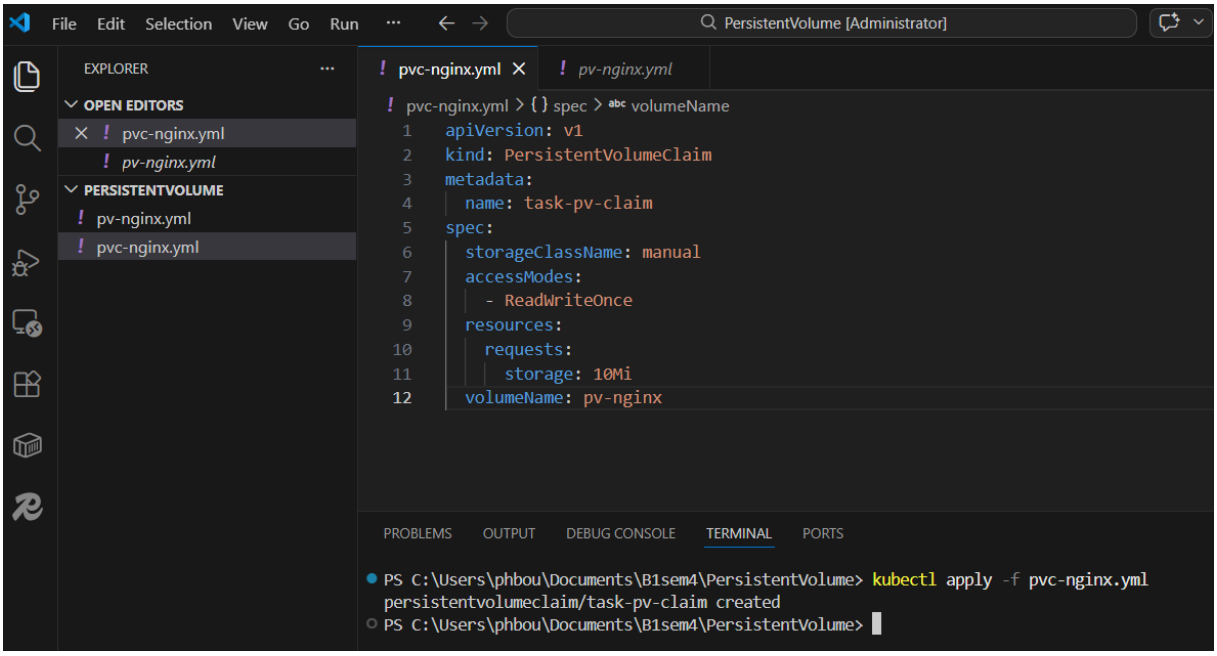


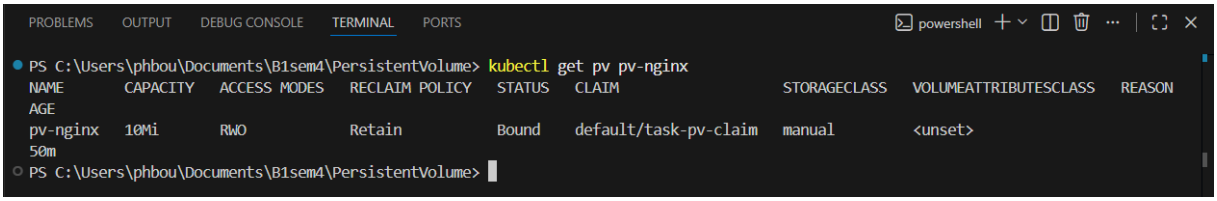


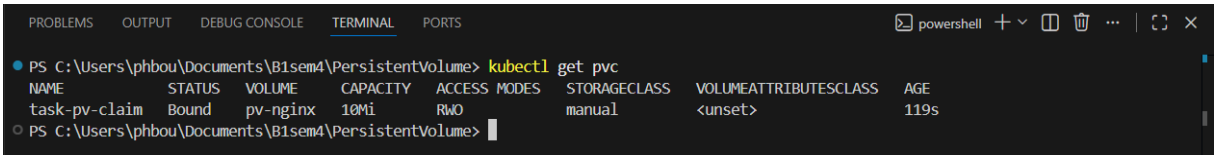
Le STATUS du PersitentVolume est Available. Cela signifie qu'il n'a pas encore été attaché à un PersistentVolumeClaim.

- Créer un PersistentVolumeClaim (demande de stockage)

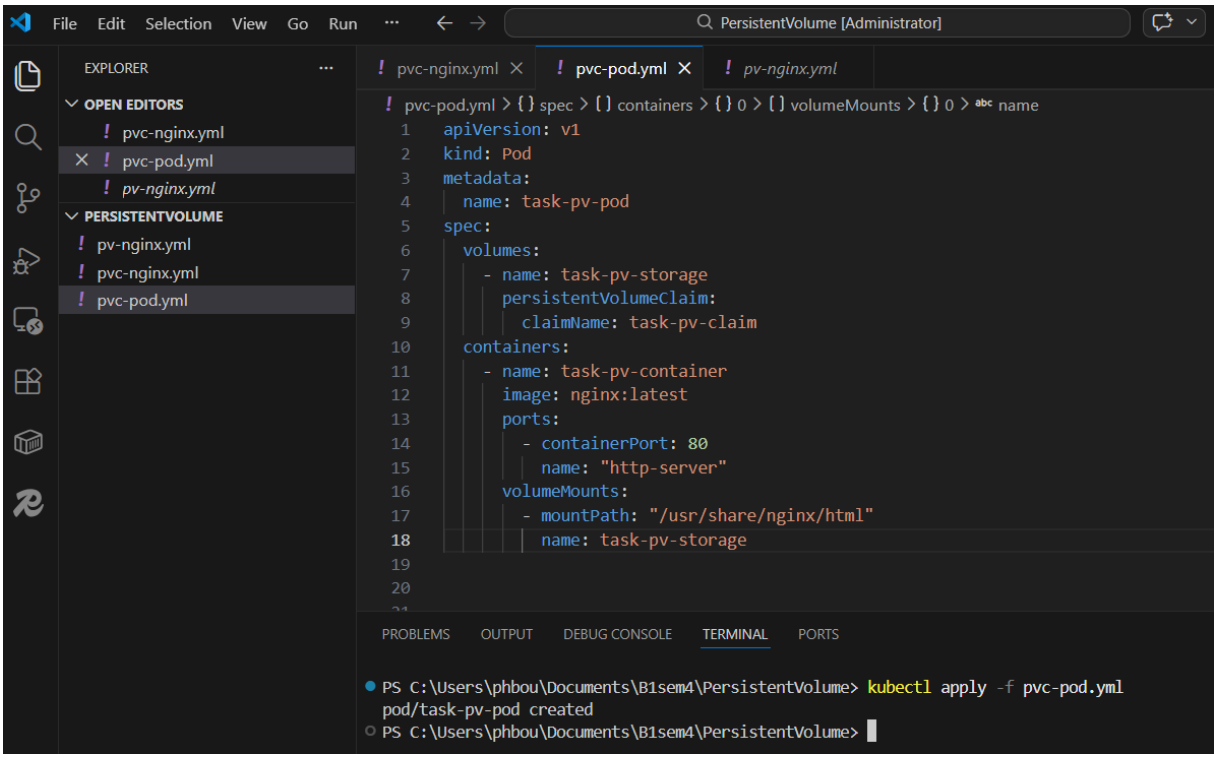
Les Pods utilisent les PersistentVolumeClaims pour demander un accès à du stockage physique.

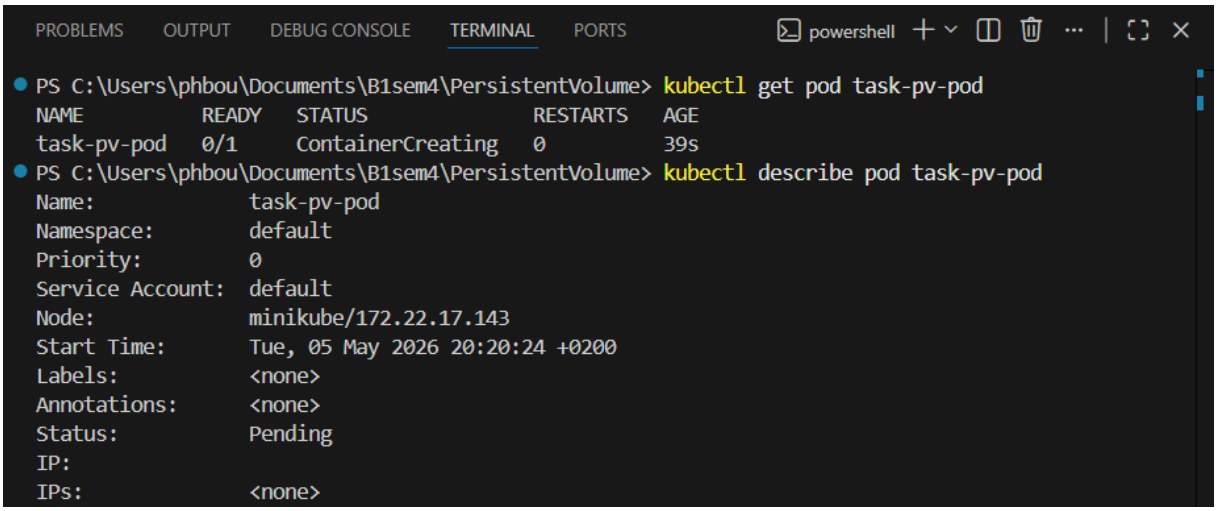


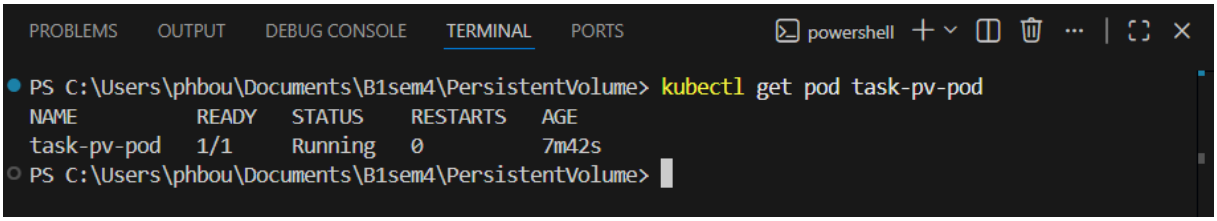


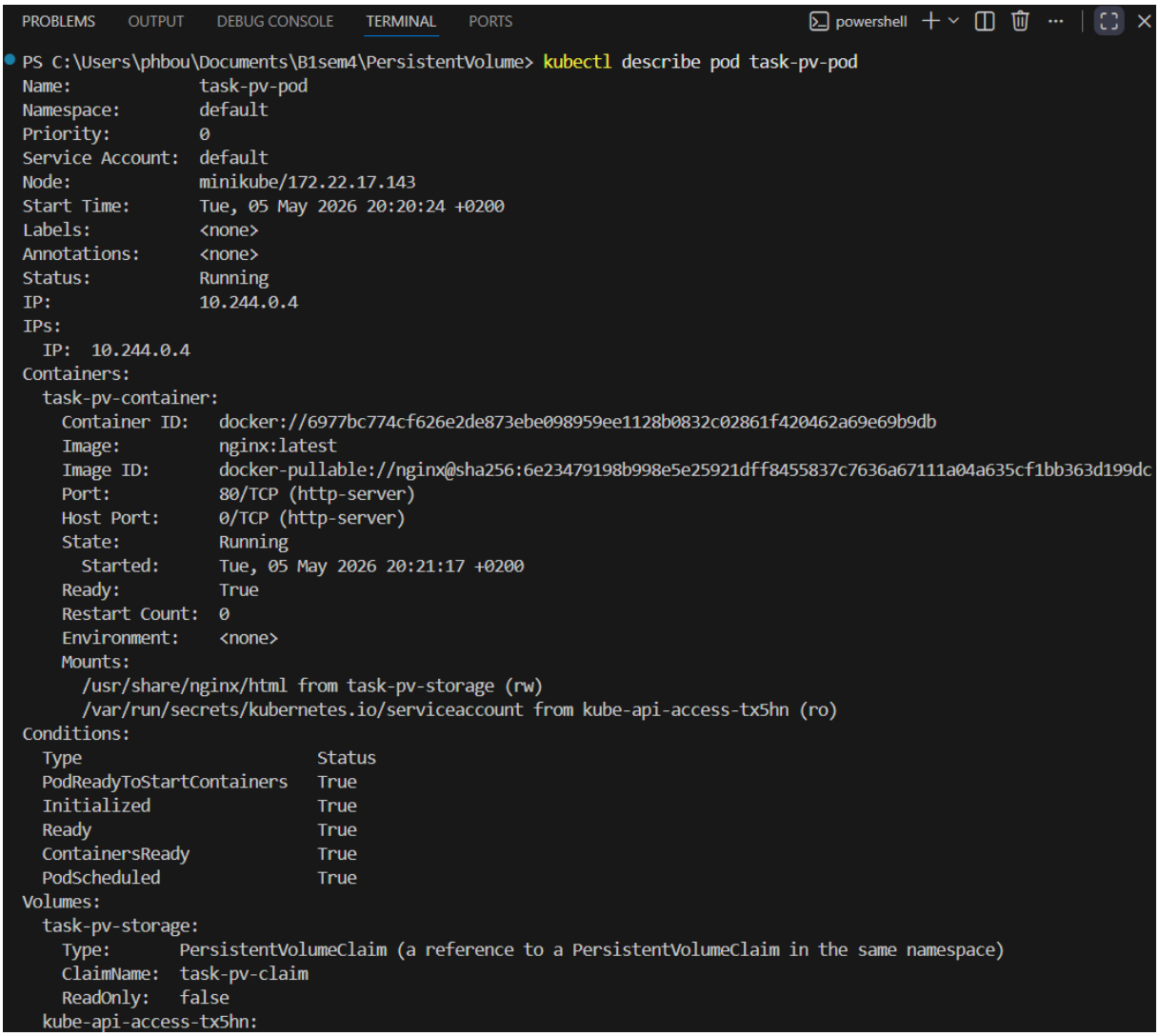


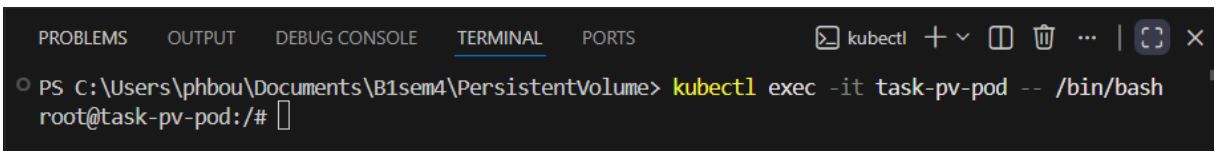
- Créer un Pod qui utilise le PersistentVolumeClaim comme volume de stockage

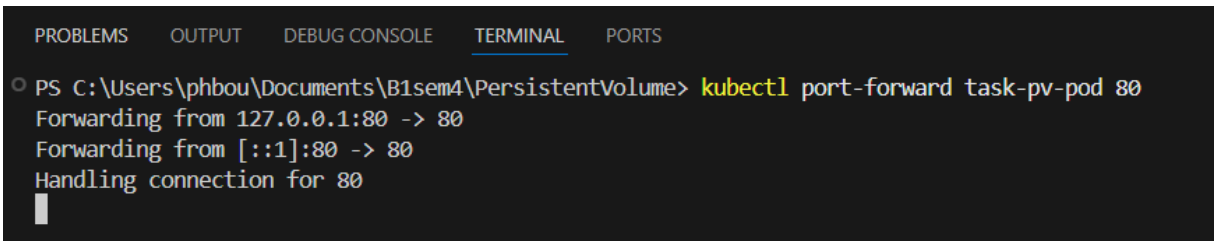

File Edit Selection View Go Run
PersistentVolume [Administrator]
EXPLORER
OPEN EDITORS
pvc-nginx.yml
pvc-pod.yml
pv-nginx.yml
PERSISTENTVOLUME
pv-nginx.yml
pvc-nginx.yml
pvc-pod.yml
pvc-nginx.yml
pvc-pod.yml
pv-nginx.yml
pvc-pod.yml > { } spec > [ ] containers > { } 0 > [ ] volumeMounts > { } 0 > abc name
1 apiVersion: v1
2 kind: Pod
3 metadata:
4 name: task-pv-pod
5 spec:
6 volumes:
7 - name: task-pv-storage
8 persistentVolumeClaim:
9 claimName: task-pv-claim
10 containers:
11 - name: task-pv-container
12 image: nginx:latest
13 ports:
14 - containerPort: 80
15 name: "http-server"
16 volumeMounts:
17 - mountPath: "/usr/share/nginx/html"
18 name: task-pv-storage
19
20
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\phbou\Documents\B1sem4\PersistentVolume> kubectl apply -f pvc-pod.yml
pod/task-pv-pod created
PS C:\Users\phbou\Documents\B1sem4\PersistentVolume>

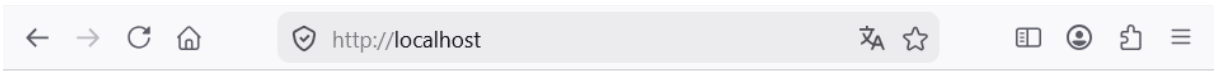

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\Users\phbou\Documents\B1sem4\PersistentVolume> kubectl get pod task-pv-pod
NAME READY STATUS RESTARTS AGE
task-pv-pod 0/1 ContainerCreating 0 39s
PS C:\Users\phbou\Documents\B1sem4\PersistentVolume> kubectl describe pod task-pv-pod
Name: task-pv-pod
Namespace: default
Priority: 0
Service Account: default
Node: minikube/172.22.17.143
Start Time: Tue, 05 May 2026 20:20:24 +0200
Labels: <none>
Annotations: <none>
Status: Pending
IP:
IPs: <none>


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\Users\phbou\Documents\B1sem4\PersistentVolume> kubectl get pod task-pv-pod
NAME READY STATUS RESTARTS AGE
task-pv-pod 1/1 Running 0 7m42s
PS C:\Users\phbou\Documents\B1sem4\PersistentVolume>

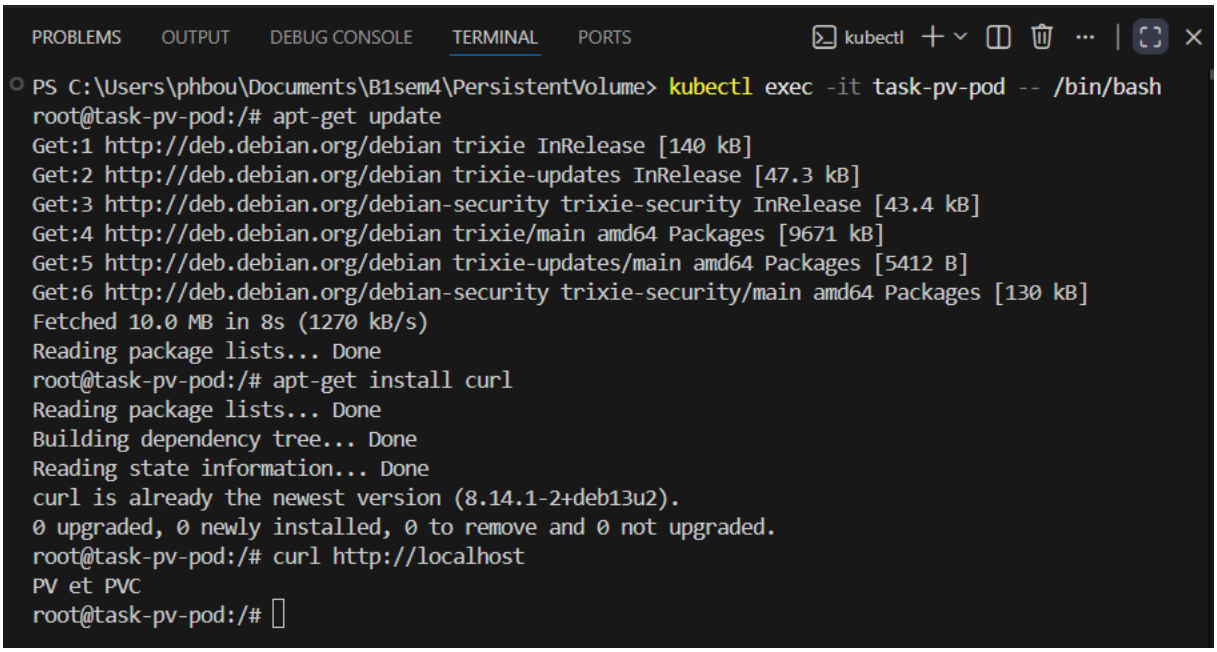

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell
PS C:\Users\phbou\Documents\B1sem4\PersistentVolume> kubectl describe pod task-pv-pod
Name: task-pv-pod
Namespace: default
Priority: 0
Service Account: default
Node: minikube/172.22.17.143
Start Time: Tue, 05 May 2026 20:20:24 +0200
Labels: <none>
Annotations: <none>
Status: Running
IP: 10.244.0.4
IPs:
IP: 10.244.0.4
Containers:
task-pv-container:
Container ID: docker://6977bc774cf626e2de873ebe098959ee1128b0832c02861f420462a69e69b9db
Image: nginx:latest
Image ID: docker-pullable://nginx@sha256:6e23479198b998e5e25921dff8455837c7636a67111a04a635cf1bb363d199dc
Port: 80/TCP (http-server)
Host Port: 0/TCP (http-server)
State: Running
Started: Tue, 05 May 2026 20:21:17 +0200
Ready: True
Restart Count: 0
Environment: <none>
Mounts:
/usr/share/nginx/html from task-pv-storage (rw)
/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-tx5hn (ro)
Conditions:
Type Status
PodReadyToStartContainers True
Initialized True
Ready True
ContainersReady True
PodScheduled True
Volumes:
task-pv-storage:
Type: PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
ClaimName: task-pv-claim
ReadOnly: false
kube-api-access-tx5hn:


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS kubectl
PS C:\Users\phbou\Documents\B1sem4\PersistentVolume> kubectl exec -it task-pv-pod -- /bin/bash
root@task-pv-pod:/#


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\phbou\Documents\B1sem4\PersistentVolume> kubectl port-forward task-pv-pod 80
Forwarding from 127.0.0.1:80 -> 80
Forwarding from [::1]:80 -> 80
Handling connection for 80


http://localhost

PV et PVC



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

1. Mises en place d'une réplication entre pods