

# Compte Rendu - TP3 Kubernetes

Étudiant : Jules DEBÉE

## Introduction

Cette partie du TP présente la création d'une tâche planifiée avec **CronJob** et la configuration d'un **Horizontal Pod Autoscaler (HPA)** sur une application déployée dans Kubernetes. L'objectif est de planifier une tâche répétitive avec CronJob et de démontrer l'ajustement automatique du nombre de pods avec HPA en fonction de la charge sur une application.

## Capture d'écran

### 1. Exposer l'Application Localement :

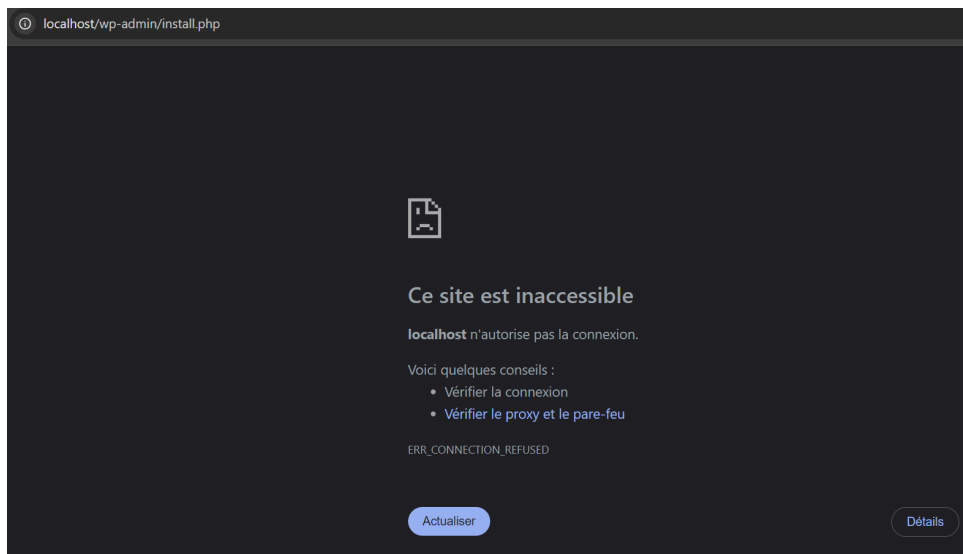
`kubectl expose deployment mon-application --port=80 --type=LoadBalancer`

```
jules@LEGIONJULES: /mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/TP-Kubernetes/TP2/TP$ kubectl expose deployment mon-application --port=80 --type=LoadBalancer
service/mon-application exposed
```

### 2. Connexion au pod Wordpress et vérifier la présence des 4 variables d'environnement avec les valeurs définies :

```
jules@LEGIONJULES: /mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/TP-Kubernetes/TP2/TP$ kubectl get pods -n middle
NAME                                READY   STATUS    RESTARTS   AGE
wordpress-cd654d6b4-gfdcb         1/1     Running   0           4m50s
jules@LEGIONJULES: /mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/TP-Kubernetes/TP2/TP$ kubectl exec -it wordpress-cd654d6b4-gfdcb -n middle -- env | grep WORDPRESS
WORDPRESS_DB_HOST=mysql.database.svc.cluster.local
WORDPRESS_DB_USER=jules
WORDPRESS_DB_NAME=wordpress
WORDPRESS_DB_PASSWORD=123456
WORDPRESS_SERVICE_PORT=80
WORDPRESS_PORT=tcp://10.102.3.191:80
WORDPRESS_PORT_80_TCP_PROTO=tcp
WORDPRESS_PORT_80_TCP_PORT=80
WORDPRESS_PORT_80_TCP_ADDR=10.102.3.191
WORDPRESS_SERVICE_HOST=10.102.3.191
WORDPRESS_PORT_80_TCP=tcp://10.102.3.191:80
jules@LEGIONJULES: /mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/TP-Kubernetes/TP2/TP$
```

### 3. Accéder à la page d'accueil de votre site depuis votre navigateur :



Note : Cela accède à wordpress mais demande les installations, vue avec le prof.

## Question

? Avant de poursuivre, expliquez le résultat souhaité.

Lors de l'application du fichier YAML hpa-application.yml, une ressource HPA est créée pour surveiller le déploiement mon-application utilisant l'image nginx, avec un minimum de 1 réplique et un maximum de 5 répliques, tout en surveillant l'utilisation du CPU avec un seuil d'activation fixé à 50 % pour ajuster dynamiquement le nombre de pods en fonction de la charge.

? Vérifiez le HPA. Quelle commande avez-vous utilisé ?

```
jules@LEGIONJULES:/mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/tp-kubernetes/TP3/TP$ kubectl get hpa
NAME                REFERENCE                TARGETS          MINPODS  MAXPODS  REPLICAS  AGE
mon-application      Deployment/mon-application  cpu: <unknown>/50%   1         5         1         9m34s
```

? Expliquez les différentes colonnes retournées par la commande

- **NAME** : Nom du HPA.
- **REFERENCE** : L'objet Kubernetes sur lequel l'HPA s'applique (ex. Deployment sur cette capture).
- **TARGETS** : L'utilisation actuelle des ressources (ex CPU comme ici).
- **MINPODS** : Nombre minimal de pods définis.
- **MAXPODS** : Nombre maximal de pods définis.
- **REPLICAS** : Nombre actuel de pods.
- **AGE** : Temps écoulé depuis la création de l'HPA.

? Votre HPA fonctionne-t-il comme souhaité ? Ajustez le si nécessaire

Vue avec le prof en séance :

```
jules@LEGIONJULES:/mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/tp-kubernetes/TP3/TP$ kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
date-cronjob-28810884-5hzws         0/1     Completed           0           2m13s
date-cronjob-28810885-42dp7         0/1     Completed           0           73s
date-cronjob-28810886-4rwxl         0/1     Completed           0           13s
mon-application-647487cb9-k2lqx     0/1     ContainerCreating   0           3s
mon-application-d7897c884-wrd7d     1/1     Running             0           5m33s
nginx-deployment-6884d88685-8kvpl   1/1     Running             1 (6h33m ago)  14d
nginx-deployment-6884d88685-jt7qb   1/1     Running             1 (6h33m ago)  14d
nginx-deployment-6884d88685-mt7g8   1/1     Running             1 (6h33m ago)  14d
simulation-utilisateur              1/1     Running             0           9m1s
```

```
jules@LEGIONJULES:/mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/tp-kubernetes/TP3/TP$ kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
date-cronjob-28810884-5hzws         0/1     Completed           0           2m34s
date-cronjob-28810885-42dp7         0/1     Completed           0           94s
date-cronjob-28810886-4rwxl         0/1     Completed           0           34s
mon-application-647487cb9-k2lqx     1/1     Running             0           24s
nginx-deployment-6884d88685-8kvpl   1/1     Running             1 (6h33m ago)  14d
nginx-deployment-6884d88685-jt7qb   1/1     Running             1 (6h33m ago)  14d
nginx-deployment-6884d88685-mt7g8   1/1     Running             1 (6h33m ago)  14d
simulation-utilisateur              1/1     Running             0           9m22s
```

```
jules@LEGIONJULES:/mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/tp-kubernetes/TP3/TP$ kubectl describe hpa
Name:                                mon-application
Namespace:                           default
Labels:                               <none>
Annotations:                           <none>
CreationTimestamp:                    Fri, 11 Oct 2024 13:05:17 +0200
Reference:                            Deployment/mon-application
Metrics:                              ( current / target )
  resource cpu on pods  (as a percentage of request):  1000% (10m) / 1%
Min replicas:                               1
Max replicas:                               5
Deployment pods:                            1 current / 4 desired
Conditions:
  Type            Status  Reason                        Message
  ----            -
  AbleToScale     True    SucceededRescale              the HPA controller was able to update the target scale to 4
  ScalingActive   True    ValidMetricFound              the HPA was able to successfully calculate a replica count from cpu resource utilization (percentage of request)
  ScalingLimited  True    ScaleUpLimit                  the desired replica count is increasing faster than the maximum scale rate

Events:
  Type    Reason                  Age    From                      Message
  ----    -
  Warning FailedGetResourceMetric  5m54s (x561 over 145m) horizontal-pod-autoscaler failed to get cpu utilization: unable to get metrics for resource cpu: unable to fetch metrics from resource metrics API: the server could not find the requested resource (get pods.metrics.k8s.io)
  Warning FailedGetResourceMetric  54s (x7 over 2m24s) horizontal-pod-autoscaler failed to get cpu utilization: unable to get metrics for resource cpu: no metric s returned from resource metrics API
```