

Lab2 – Exploiter Kubernetes

Brahim HAMDI

Introduction

Dans ce Lab nous montrerons comment :

- Créer un pod, deployment et daemonset
- Scaler un pod
- Déployer une nouvelle version

Création de pod

Le pod est la plus petite unité gérée par kubernetes. Dans cette partie, nous créerons un pod avec 2 méthodes différentes :

- Commande impérative
- En utilisant un fichier yaml

1. Nous commençons par la création d'un pod avec la commande impérative.

- Lister les pods du namespace par défaut

```
brahim@Training:~$ kubectl get pod
No resources found in default namespace.
brahim@Training:~$
```

- Créer un pod nommé *first-pod* qui exécute l'image *particule/helloworld:1.0.0*, et vérifier que le pod a été bien créé.

```
brahim@Training:~$ kubectl run first-pod --image=particule/helloworld:1.0.0
pod/first-pod created
brahim@Training:~$ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
first-pod     1/1     Running   0           95s
brahim@Training:~$
```

- Sur quel nœud le pod tourne t-il ? Quel est son IP ?

```
brahim@Training:~$ kubectl get pod -o wide
NAME          READY   STATUS    RESTARTS   AGE    IP           NODE          NOMINATED NODE   READINESS GATES
first-pod     1/1     Running   0           2m55s  10.244.2.19  k8s-worker2   <none>           <none>
brahim@Training:~$
```

- Afficher des informations détaillées sur le pod *first-pod* :

```
brahim@Training:~$ kubectl describe pod first-pod
Name:          first-pod
Namespace:     default
Priority:       0
Service Account: default
Node:          k8s-worker2/192.168.205.102
Start Time:    Tue, 23 May 2023 19:45:46 +0200
Labels:        run=first-pod
Annotations:    <none>
Status:        Running
IP:            10.244.2.19
IPs:
  IP: 10.244.2.19
Containers:
  first-pod:
    Container ID:  containerd://69dfbf2456b69e518ea2703c3e34d6c4df3117a0607d84bbaf8dbab36a4a8f09
    Image:         particule/helloworld:1.0.0
    Image ID:      docker.io/particule/helloworld@sha256:608d88072f6cfe75a3f1c59354737dc0c17d0d845031b8d33ede344bb59e7a2c
```

```
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    Pulled       4m46s kubelet        Container image "particule/helloworld:1.0.0" already present on machine
Normal    Created      4m46s kubelet        Created container first-pod
Normal    Started      4m46s kubelet        Started container first-pod
Normal    Scheduled    4m44s default-scheduler Successfully assigned default/first-pod to k8s-worker2
```

- Supprimer le pod.

```
brahim@Training:~$ kubectl delete pod first-pod
pod "first-pod" deleted

brahim@Training:~$
brahim@Training:~$ kubectl get pod
No resources found in default namespace.
brahim@Training:~$
```

2. Avec la deuxième méthode, on définit le pod dans un fichier yaml puis on l'applique. Cette méthode permet de modifier, réutiliser et distribuer le manifest yaml.

- Ecrire le fichier *pod1.yaml* qui définit le pod crée précédemment avec les même paramètres. Vous pouvez copier et modifier le yaml du lien suivant : <https://raw.githubusercontent.com/kubernetes/website/main/content/en/examples/pods/simple-pod.yaml>

```
apiVersion: v1
kind: Pod
metadata:
  name: first-pod
spec:
  containers:
  - name: first-container
    image: particule/helloworld:1.0.0
    ports:
    - containerPort: 80
```

- Appliquer le fichier *pod1.yaml*, puis vérifier que le pod est bien crée. Sur quel nœud le pod tourne t-il ?

```
brahim@Training:~/Lab2$ kubectl apply -f pod1.yaml
pod/first-pod created
brahim@Training:~/Lab2$
brahim@Training:~/Lab2$ kubectl get pod -o wide
NAME      READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
first-pod 1/1     Running   0          62s   10.244.1.19  k8s-worker1 <none>          <none>
```

- Est-il possible de créer 2 répliquas du pod *first-pod* ?

```
brahim@Training:~/Lab2$ kubectl scale
error: required flag(s) "replicas" not set
brahim@Training:~/Lab2$ kubectl scale --help
Set a new size for a deployment, replica set, replication controller, or stateful set.
```

On ne peut pas répliquer un pod. Mais il possible de répliquer un déploiement, rs, rc et statefulset.

- Supprimer le pod.

```
brahim@Training:~/Lab2$ kubectl delete pod first-pod
pod "first-pod" deleted
brahim@Training:~/Lab2$ kubectl get pod
No resources found in default namespace.
brahim@Training:~/Lab2$
```

Création de deployment

Le deployment est un objet de haut niveau d'abstraction. Il contrôle et pilote les ReplicaSets et les Pods. Il ajoute plusieurs fonctionnalités, comme le scaling et le rollback.

3. Nous allons utiliser la commande impérative pour créer un déploiement qui va créer lui même un ReplicaSet et un Pod qui exécute l'image *particule/helloworld:1.0.0*

- Exécuter la commande suivante pour créer le déploiement *first-deployment*, qui exécute l'image *particule/helloworld:1.0.0*, puis vérifier que le déploiement est crée. Quels sont les autres objets créés ?

```
brahim@Training:~/Lab2$ kubectl create deployment first-deployment --image=particule/helloworld:1.0.0
deployment.apps/first-deployment created
brahim@Training:~/Lab2$ kubectl get deployment
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
first-deployment    1/1     1            1           8s
brahim@Training:~/Lab2$ kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/first-deployment-bd7fc7fc8-kw5xh    1/1     Running   0           13s

NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP    71m

NAME                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/first-deployment    1/1     1            1           13s

NAME                DESIRED   CURRENT   READY   AGE
replicaset.apps/first-deployment-bd7fc7fc8    1         1         1       13s
brahim@Training:~/Lab2$
```

- Afficher des informations détaillées sur le déploiement *first-deployment*

```
brahim@Training:~/Lab2$ kubectl describe deployment.apps/first-deployment
Name:                first-deployment
Namespace:           default
CreationTimestamp:   Tue, 23 May 2023 20:32:39 +0200
Labels:              app=first-deployment
Annotations:         deployment.kubernetes.io/revision: 1
Selector:            app=first-deployment
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=first-deployment
  Containers:
    helloworld:
      Image:   particule/helloworld:1.0.0
      Port:   <none>
      Host Port: <none>
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
  Conditions:
    Type           Status    Reason
    ----           -
    Available      True     MinimumReplicasAvailable
    Progressing    True     NewReplicaSetAvailable
OldReplicaSets:   <none>
NewReplicaSet:    first-deployment-bd7fc7fc8 (1/1 replicas created)
Events:
  Type    Reason          Age    From                      Message
  ----    -
  Normal  ScalingReplicaSet  12s    deployment-controller     Scaled up replica set first-deployment-bd7fc7fc8 to 1
brahim@Training:~/Lab2$
```

- Supprimer le déploiement *first-deployment*. Les ReplicaSets et les Pods sont-ils supprimés aussi ?

```
brahim@Training:~/Lab2$ kubectl delete deployment.apps/first-deployment
deployment.apps "first-deployment" deleted
brahim@Training:~/Lab2$ kubectl get all
NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP    94m
brahim@Training:~/Lab2$ kubectl get pod
No resources found in default namespace.
brahim@Training:~/Lab2$ kubectl get replicaset
No resources found in default namespace.
brahim@Training:~/Lab2$
```

4. Maintenant, on va définir le déploiement créer précédemment dans un fichier yaml, et on va l'appliquer par la suite.

- Créer le fichier *deploy1.yaml* qui définit le déploiement *first-deployment* en haut niveau d'abstraction et le template du pod *first-pod* en bas niveau.

Vous pouvez vous copier et modifier le yaml du lien suivant :

<https://raw.githubusercontent.com/kubernetes/website/main/content/en/examples/controllers/nginx-deployment.yaml>

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: first-deployment
  labels:
    app: app1
spec:
  selector:
    matchLabels:
      app: app1
  template:
    metadata:
      labels:
        app: app1
    spec:
      containers:
        - name: container1
          image: particule/helloworld:1.0.0
          ports:
            - containerPort: 80
```

- Appliquer le fichier yaml et vérifier la création du déploiement en plus du ReplicaSet et du Pod

```
brahim@Training:~/Lab2$ kubectl apply -f deploy1.yaml
deployment.apps/first-deployment created
brahim@Training:~/Lab2$ kubectl get all
NAME                                     READY   STATUS    RESTARTS   AGE
pod/first-deployment-685cd76f87-m5xg2   1/1     Running   0           6s

NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP    143m

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/first-deployment        1/1     1             1           6s

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/first-deployment-685cd76f87  1         1         1       6s
brahim@Training:~/Lab2$
```

- Sur quel nœud le pod est-il crée ? Quels sont les labels utilisés ?

```
brahim@Training:~/Lab2$ kubectl get all -o wide
NAME                                     READY   STATUS    RESTARTS   AGE   IP             NODE       NOMINATED NODE   READINESS GATES
pod/first-deployment-685cd76f87-m5xg2   1/1     Running   0           2m21s  10.244.1.20    k8s-worker1  <none>           <none>

NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE   SELECTOR
service/kubernetes  ClusterIP   10.96.0.1    <none>        443/TCP    145m  <none>

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES                               SELECTOR
deployment.apps/first-deployment        1/1     1             1           2m21s  container1   particule/helloworld:1.0.0         app=app1

NAME                                     DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES                               SELECTOR
replicaset.apps/first-deployment-685cd76f87  1         1         1       2m21s  container1   particule/helloworld:1.0.0         app=app1,pod-template-hash=685cd76f87
brahim@Training:~/Lab2$
```

Scaler les pods

Dans cette partie nous allons découvrir la puissance de kubernetes de supporter la montée en charge en répliquant les pods avec le Deployment, le DaemonSet ou le StatefulSet.

5. On va appliquer le scaling sur le déploiement crée précédemment.

- Créer 3 réplicas du pod appartenant à ce déploiement. Sur quels nœuds les 3 replicas sont-ils créés ? Quel est l'IP de chaque replicas ?

```
brahim@Training:~/Lab2$ kubectl scale deployment.apps/first-deployment --replicas=3
deployment.apps/first-deployment scaled
brahim@Training:~/Lab2$ kubectl get all -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
pod/first-deployment-685cd76f87-jnqf8	1/1	Running	0	22s	10.244.2.21	k8s-worker2	<none>	<none>
pod/first-deployment-685cd76f87-lvh6k	1/1	Running	0	22s	10.244.2.22	k8s-worker2	<none>	<none>
pod/first-deployment-685cd76f87-m5xg2	1/1	Running	0	6m	10.244.1.20	k8s-worker1	<none>	<none>

```
brahim@Training:~/Lab2$ kubectl get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	149m	<none>

```
brahim@Training:~/Lab2$ kubectl get deployment
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE	CONTAINERS	IMAGES	SELECTOR
deployment.apps/first-deployment	3/3	3	3	6m	container1	particule/helloworld:1.0.0	app=app1

```
brahim@Training:~/Lab2$ kubectl get replicaset
```

NAME	DESIRED	CURRENT	READY	AGE	CONTAINERS	IMAGES	SELECTOR
replicaset.apps/first-deployment-685cd76f87	3	3	3	6m	container1	particule/helloworld:1.0.0	app=app1,pod-template-hash=685cd76f87

- Exposer les pods du déploiement sur un port TCP de chaque nœud du cluster. Quel est le numéro de ce port ?

```
brahim@Training:~/Lab2$ kubectl expose deployment first-deployment --port=80 --type=NodePort
service/first-deployment exposed
brahim@Training:~/Lab2$ kubectl get all -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
pod/first-deployment-685cd76f87-jnqf8	1/1	Running	0	21m	10.244.2.21	k8s-worker2	<none>	<none>
pod/first-deployment-685cd76f87-lvh6k	1/1	Running	0	21m	10.244.2.22	k8s-worker2	<none>	<none>
pod/first-deployment-685cd76f87-m5xg2	1/1	Running	0	26m	10.244.1.20	k8s-worker1	<none>	<none>

```
brahim@Training:~/Lab2$ kubectl get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
service/first-deployment	NodePort	10.99.174.166	<none>	80:31609/TCP	6s	app=app1
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	170m	<none>

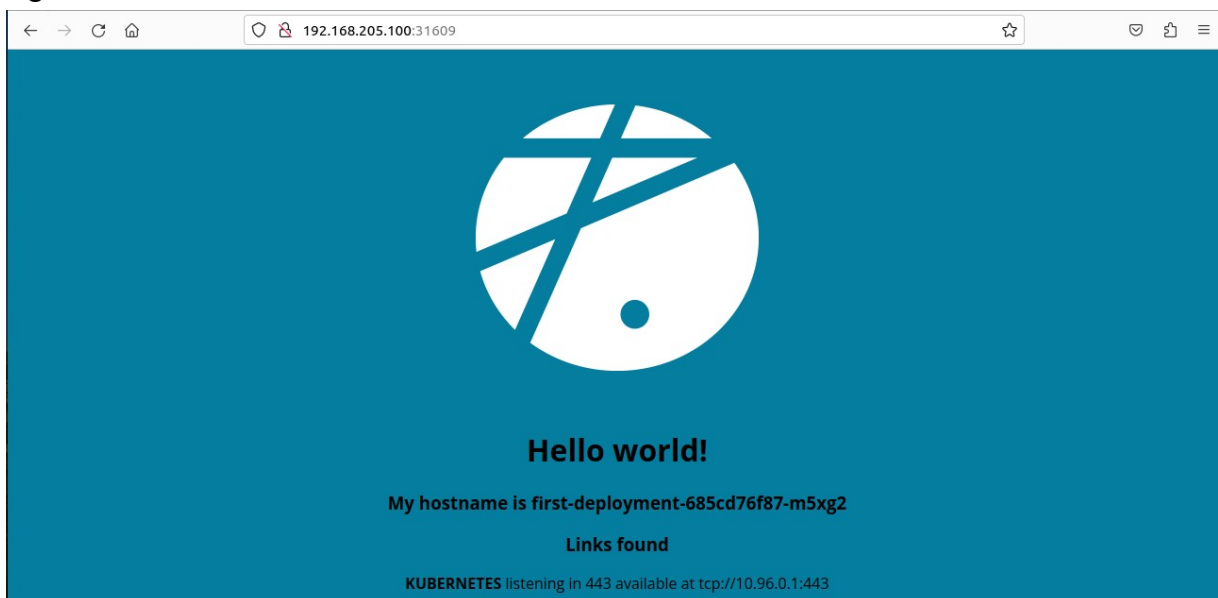
```
brahim@Training:~/Lab2$ kubectl get deployment
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE	CONTAINERS	IMAGES	SELECTOR
deployment.apps/first-deployment	3/3	3	3	26m	container1	particule/helloworld:1.0.0	app=app1

```
brahim@Training:~/Lab2$ kubectl get replicaset
```

NAME	DESIRED	CURRENT	READY	AGE	CONTAINERS	IMAGES	SELECTOR
replicaset.apps/first-deployment-685cd76f87	3	3	3	26m	container1	particule/helloworld:1.0.0	app=app1,pod-template-hash=685cd76f87

- En utilisant le couple *IP_Node:Port_Node*, afficher l'interface de l'application sur votre navigateur.



Déploiement d'une nouvelle version

On va découvrir maintenant la stratégie de Kubernetes à déployer une nouvelle version d'une application et la possibilité de faire un rollback.

6. Vous allez migrer votre application vers une nouvelle version.

◦ Dans le fichier `deploy1.yaml` remplacer la version de l'image *particule/helloworld* de la version *1.0.0* vers la version *2.0.0*

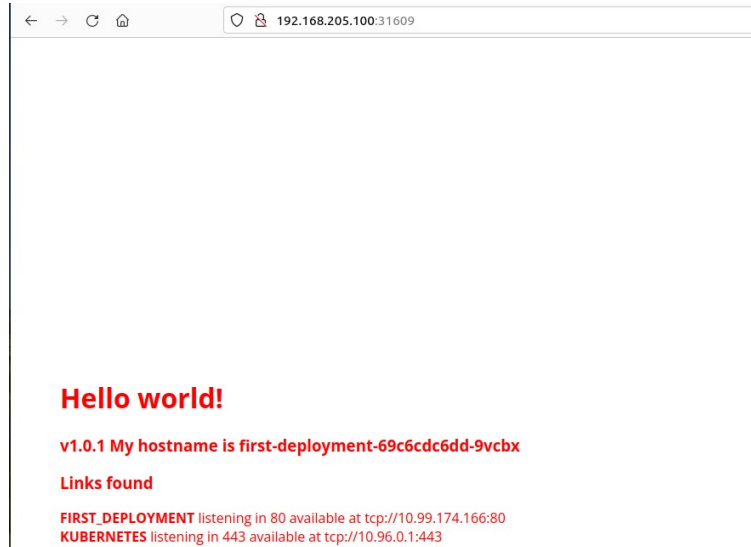
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: first-deployment
  labels:
    app: app1
spec:
  selector:
    matchLabels:
      app: app1
  template:
    metadata:
      labels:
        app: app1
    spec:
      containers:
        - name: container1
          image: particule/helloworld:2.0.0
          ports:
            - containerPort: 80
```

◦ Une fois sauvegardé, appliquer le fichier et rapidement observer la stratégie par défaut adoptée par kubernetes.

```
brahim@Training:~/Lab2$ kubectl apply -f deploy1.yaml
deployment.apps/first-deployment configured
brahim@Training:~/Lab2$
```

```
brahim@Training:~/Lab2$ kubectl describe deployment.apps/first-deployment
Name: first-deployment
Namespace: default
CreationTimestamp: Tue, 23 May 2023 21:44:27 +0200
Labels: app=app1
Annotations: deployment.kubernetes.io/revision: 2
Selector: app=app1
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels: app=app1
  Containers:
    container1:
      Image: particule/helloworld:2.0.0
      Port: 80/TCP
      Host Port: 0/TCP
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
Conditions:
  Type          Status Reason
  ----          -
  Available     True   MinimumReplicasAvailable
  Progressing   True   NewReplicaSetAvailable
OldReplicaSets: first-deployment-685cd76f87 (0/0 replicas created)
NewReplicaSet:  first-deployment-69c6cdc6dd (3/3 replicas created)
Events:
  Type          Reason          Age    From          Message
  ----          -
  Normal        ScalingReplicaSet 37m    deployment-controller Scaled up replica set first-deployment-685cd76f87 to 1
  Normal        ScalingReplicaSet 31m    deployment-controller Scaled up replica set first-deployment-685cd76f87 to 3 from 1
  Normal        ScalingReplicaSet <invalid> deployment-controller Scaled up replica set first-deployment-69c6cdc6dd to 1
  Normal        ScalingReplicaSet <invalid> deployment-controller Scaled down replica set first-deployment-685cd76f87 to 2 from 3
  Normal        ScalingReplicaSet <invalid> deployment-controller Scaled up replica set first-deployment-69c6cdc6dd to 2 from 1
  Normal        ScalingReplicaSet <invalid> deployment-controller Scaled down replica set first-deployment-685cd76f87 to 1 from 2
  Normal        ScalingReplicaSet <invalid> deployment-controller Scaled up replica set first-deployment-69c6cdc6dd to 3 from 2
  Normal        ScalingReplicaSet <invalid> deployment-controller Scaled down replica set first-deployment-685cd76f87 to 0 from 1
brahim@Training:~/Lab2$
```

- Afficher l'interface de la nouvelle version de l'application.



7. Si la nouvelle version ne fonctionne pas correctement, kubernetes fait un *rollback* automatique vers l'ancienne version. C'est pas le cas ici, donc on va faire un rollback manuellement.

- Faire un rollback vers l'ancienne version *particule/helloworld:1.0.0*

```
brahim@Training:~/Lab2$ kubectl rollout status deployment/first-deployment
deployment "first-deployment" successfully rolled out
brahim@Training:~/Lab2$ kubectl rollout undo deployment/first-deployment
deployment.apps/first-deployment rolled back
brahim@Training:~/Lab2$
```

```
brahim@Training:~/Lab2$ kubectl describe deployment.apps/first-deployment
Name: first-deployment
Namespace: default
CreationTimestamp: Tue, 23 May 2023 21:44:27 +0200
Labels: app=app1
Annotations: deployment.kubernetes.io/revision: 3
Selector: app=app1
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels: app=app1
  Containers:
    container1:
      Image: particule/helloworld:1.0.0
      Port: 80/TCP
      Host Port: 0/TCP
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
Conditions:
  Type           Status  Reason
  ----           -
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets: first-deployment-69c6cdc6dd (0/0 replicas created)
NewReplicaSet:  first-deployment-685cd76f87 (3/3 replicas created)
Events:
  Type    Reason             Age   From              Message
  ----    -
  Normal  ScalingReplicaSet  47m   deployment-controller  Scaled up replica set first-deployment-685cd76f87 to 1
  Normal  ScalingReplicaSet  41m   deployment-controller  Scaled up replica set first-deployment-685cd76f87 to 3 from 1
```

- Vérifier sur l'interface de l'application qu'on est sur l'ancienne version

