

Limitation des ressources dans un namespace

Dans cet exercice, nous allons créer un namespace et ajouter des quotas afin limiter les ressources pouvant être utilisées dans celui-ci.

Création d'un namespace

Créez le namespace *test* en utilisant la commande suivante:

```
$ kubectl create namespace test
```

Quota d'utilisation des ressources

Copiez, dans le fichier *quota.yaml*, le contenu ci-dessous.

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: quota
spec:
  hard:
    requests.cpu: "1"
    requests.memory: 1Gi
    limits.cpu: "2"
    limits.memory: 2Gi
```

Celui-ci définit une ressource de type *ResourceQuota* qui limitera l'utilisation de la mémoire et du cpu dans le namespace associé. Au sein de celui-ci:

- chaque container devra spécifier des demandes et des limites pour la RAM et le cpu
- l'ensemble des containers ne pourra pas demander plus de 1GB de RAM
- l'ensemble des containers ne pourra pas utiliser plus de 2GB de RAM
- l'ensemble des containers ne pourra pas demander plus d'1 cpu
- l'ensemble des containers ne pourra pas utiliser plus de 2 cpus

En utilisant la commande suivante, créez cette nouvelle ressource en l'associant au

namespace *test*.

```
$ kubectl apply -f quota.yaml --namespace=test
```

Lancement d'un Pod

Créez un fichier `pod-quota-1.yaml`, et copiez le contenu suivant:

```
apiVersion: v1
kind: Pod
metadata:
  name: quota-mem-cpu-1
  namespace: test
spec:
  containers:
  - name: www
    image: nginx:1.14-alpine
    resources:
      limits:
        memory: "800Mi"
        cpu: "800m"
      requests:
        memory: "600Mi"
        cpu: "400m"
```

L'unique container de ce Pod définit des demandes et des limites pour la RAM et le CPU.

Créez ce Pod avec la commande suivante:

```
$ kubectl apply -f pod-quota-1.yaml
```

Vérifiez que le Pod a été créé correctement:

```
$ kubectl get po -n test
```

NAME	READY	STATUS	RESTARTS	AGE
quota-mem-cpu-demo	1/1	Running	0	11s

Vérification de l'utilisation du quota

Utilisez la commande suivante pour voir les ressources RAM et CPU utilisée au sein du namespace:

```
$ kubectl get resourcequota quota --namespace=test --output=yaml
```

Vous devriez obtenir un résultat proche de celui ci-dessous:

```
apiVersion: v1
kind: ResourceQuota
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"ResourceQuota","metadata":{"annotations":
      {},"name":"quota","namespace":"test"},"spec":{"hard":
      {"limits.cpu":"2","limits.memory":"2Gi","requests.cpu":"1","requests.memory":"1Gi"}}
  creationTimestamp: "2019-04-02T07:19:40Z"
  name: quota
  namespace: test
  resourceVersion: "55335"
  selfLink: /api/v1/namespaces/test/resourcequotas/quota
  uid: ae0370e7-5517-11e9-a569-0800273c95f3
spec:
  hard:
    limits.cpu: "2"
    limits.memory: 2Gi
    requests.cpu: "1"
    requests.memory: 1Gi
status:
  hard:
    limits.cpu: "2"
    limits.memory: 2Gi
    requests.cpu: "1"
    requests.memory: 1Gi
  used:
    limits.cpu: 800m
    limits.memory: 800Mi
    requests.cpu: 400m
    requests.memory: 600Mi
```

Sous la clé *used*, on peut voir que l'utilisation en RAM et cpu correspondent à celles spécifiées dans le Pod qui vient d'être créé.

Lancement d'un 2ème Pod

Créez un fichier `pod-quota-2.yaml`, et copiez le contenu suivant:

```
apiVersion: v1
kind: Pod
metadata:
  name: quota-mem-cpu-2
  namespace: test
spec:
  containers:
  - name: db
    image: redis:alpine
    resources:
      limits:
        memory: "1Gi"
        cpu: "800m"
      requests:
        memory: "700Mi"
        cpu: "400m"
```

Créez ce Pod avec la commande suivante:

```
$ kubectl apply -f pod-quota-2.yaml
```

Vous devriez obtenir un message semblable à celui ci-dessous

```
Error from server (Forbidden): error when creating "pod-quota-2.yaml": pods
"quota-mem-cpu-2" is forbidden: exceeded quota: quota, requested:
requests.memory=700Mi, used: requests.memory=600Mi, limited: requests.memory=1Gi
```

Ce nouveau Pod ne peut pas être créé car il n'y a pas assez de ressources disponibles dans le namespace `test`

Cleanup

Supprimez le namespace avec la commande suivante:

```
$ kubectl delete ns test
```