Limitar el consumo de los recursos

Limitar la Ram y la CPU de un pod

Para limitar la ram de un pod usaremos en el manifiesto del pod las opciones **request** y **limits**.

La opción **request** es para definir la cantidad de memoria que asignamos al pod y la opción limits define el máximo de memoria que podría llegar a usar ese pod.

```
apiVersion: v1
kind: Pod
metadata:
  name: limite-memoria
spec:
  containers:
  - name: limite-memoria
    image: polinux/stress
    resources:
      limits:
        memory: "200Mi"   # Definimos el límite de memoria que puede usar el
pod a 200M
      requests:
        memory: "100Mi"   #Definimos la cantidad de memoria que reservamos
para este pod
    command: ["stress"]
    args: ["--vm", "1", "--vm-bytes", "150M", "--vm-hang", "1"]
```

Para limitar la cantidad de CPU a utilizar

```
apiVersion: v1
kind: Pod
metadata:
  name: cpu-demo
spec:
  containers:
  - name: cpu-demo-ctr
    image: vish/stress
    resources:
      limits:
        cpu: "1"
      requests:
        cpu: "0.5"
    args:
    - -cpus
    - "2"
```

LimitRange

Un LimitRange permite poner limitaciones :

- Limitar la cantidad mínima y máxima de recursos que puede usar un pod o un contenedor en un namespace
- Definir los valores por defecto de **request** y **limits** para un recurso en el namespace

Veamos un ejemplo para definir los limits y los request

```
apiVersion: v1
kind: Namespace
metadata:
  name: dev
  labels:
    name: dev
---
apiVersion: v1
kind: LimitRange
metadata:
  name: mem-cpu-limit-range
  namespace: dev
spec:
  limits:
  - default:
      memory: 512Mi
      cpu: 1
    defaultRequest:
      memory: 256Mi
      cpu: 0.5
    type: Container
---
apiVersion: v1
kind: Pod
metadata:
  name: podtest3
  namespace: dev
  labels:
    app: backend
    env: dev
spec:
  containers:
  - name: cont1
    image: nginx:alpine
```

Ejemplo para definir unos valores mínimos y máximos

```
apiVersion: v1
kind: Namespace
metadata:
```

```
  name: prod
  labels:
    name: prod
---
apiVersion: v1
kind: LimitRange
metadata:
  name: min-max
  namespace: prod     #Definimos el namespace donde vamos a aplicar este
limite
spec:
  limits:
  - max:
      memory: 1Gi
      cpu: 1
    min:
      memory: 100M
      cpu: 100m
    type: Container
---
apiVersion: v1
kind: Pod
metadata:
  name: podtest3
  namespace: prod               #Definimos el namespace para el pod
  labels:
    app: backend
    env: dev
spec:
  containers:
  - name: cont1
    image: nginx:alpine
    resources:
      limits:
        memory: 50M
        cpu: 50m
```

para ver los límites

```
kubectl get limitrange -n <nombredelnamespace>
```



Limitrange sólo se aplica en el namespace en el que este definido por lo que es muy importante que los pods se definan también dentro del namespace donde lo hemos declarado

ResourceQuota

Limita el consumo de recursos por namespace

```
apiVersion: v1
kind: Namespace
metadata:
  name: uat
  labels:
    name: uat
---
apiVersion: v1
kind: ResourceQuota
metadata:
  name: res-quota
  namespace: uat
spec:
  hard:
    requests.cpu: "1"
    requests.memory: 1Gi
    limits.cpu: "2"
    limits.memory: 2Gi
```



En un RequestQuota es obligatorio definir los **requests** y los **limits**

Tambien es posible limitar el número de pods de un namespace

```
---
apiVersion: v1
kind: Namespace
metadata:
  name: pre
  labels:
    name: pre
---
apiVersion: v1
kind: ResourceQuota
metadata:
  name: pod-demo
  namespace: pre
spec:
  hard:
    pods: "3"
```

Referencias

- https://github.com/ricardoandre97/k8s-resources/blob/master/limits-requests/limit-ram.yaml
- https://github.com/ricardoandre97/k8s-resources/blob/master/limitRange/
- https://v1-17.docs.kubernetes.io/docs/concepts/policy/limit-range/

From:
https://wiki.intrusos.info/ - **LCWIKI**

Permanent link:
https://wiki.intrusos.info/doku.php?id=virtualizacion:kubernetes:limitar

Last update: **182023/01/ 13:37**