Resource Allocation & Quotas

» Limits and Requests

Overview

By default, each container has unlimited access to the host machine's available CPU cycles.

Requests

Observe the allocatable resources set to the worker node in your environment.

```
kubectl get nodes -l node-role.kubernetes.io/node -o jsonpath={.items..allocatable}; echo
```

Create a pod requesting vCPU of `200m` (millicpu/millicores, equivalent to half a vCPU) and memory of `250Mi` (Mebibytes).

Add CPU/Memory limitations as well.

```
kubectl run resource-pod --image nginx:latest --restart Never --requests cpu=200m,memory=250Mi --limits cpu=250m,memory=500Mi
```

Verify that the pod was successfully created.

```
kubectl get pods -o wide
```

Execute the `dd` program within the container to generate CPU load.

```
kubectl exec -it resource-pod /bin/dd if=/dev/zero of=/dev/null
```

In a separate terminal: Watch CPU resource usage increase and cap at 250m (millicpus).

Type `CTRL+c` to stop watching the pod when you are done.

```
watch -n .1 kubectl top pod resource-pod
```

Delete the pod when you are done.

```
kubectl delete pod resource-pod
```

Limits

Create a new namespace called `limit-example .`

```
kubectl create namespace limit-example
```

Set a compute resource limit/request on the namespace with the following manifest.

```
cat >> limits.yaml <<EOF
apiVersion: v1
kind: LimitRange
metadata:
  name: mylimits
spec:
  limits:
  - type: Pod
    max:
      cpu: "2"
      memory: 1Gi
    min:
      cpu: 200m
      memory: 6Mi
  - type: Container
    default:
      cpu: 300m
      memory: 200Mi
    defaultRequest:
      cpu: 200m
      memory: 100Mi
    max:
      cpu: "2"
      memory: 1Gi
    min:
      cpu: 100m
      memory: 3Mi
EOF
```

Create the new `LimitRange` in the limit-example namespace.

```
kubectl create -f limits.yaml -n limit-example
```

Get the basic information for the `LimitRange` resource in the `limit-example` namespace.

```
kubectl get limitranges -n limit-example
```

Verify the limits are set on the namespace.

```
kubectl describe limitranges -n limit-example
```

Create a pod within the `limit-example` namespace without specifying any requests or limits.

```
kubectl run ns-resource-pod --image nginx:latest --restart Never -n limit-example
```

Verify that the pod was successfully created.

```
kubectl get pods -n limit-example
```

Execute the `dd` program within the container to generate CPU load.

```
kubectl exec -it ns-resource-pod -n limit-example /bin/dd if=/dev/zero of=/dev/null
```

In a separate terminal: Watch CPU resource usage increase and cap at 300m (millicpus).

Type `CTRL+c` to stop watching the pod when you are done.

```
watch -n .1 kubectl top pod ns-resource-pod -n limit-example
```

Clean Up

Delete the pod.

```
kubectl delete pod ns-resource-pod -n limit-example
```

View any remaining resources.

```
kubectl get all
```