

NAME : GOURAV DAS

SAP : 500122586

BATCH : B2 DEVOPS

LAB EXERCISE 14- IMPLEMENTING RESOURCE QUOTA IN KUBERNETES

OBJECTIVE:

IN KUBERNETES, RESOURCE QUOTAS ARE USED TO CONTROL THE RESOURCE CONSUMPTION OF NAMESPACES. THEY HELP IN MANAGING AND ENFORCING LIMITS ON THE USAGE OF RESOURCES LIKE CPU, MEMORY, AND THE NUMBER OF OBJECTS (E.G., PODS, SERVICES) WITHIN A NAMESPACE. THIS EXERCISE WILL GUIDE YOU THROUGH CREATING AND MANAGING RESOURCE QUOTAS TO LIMIT THE RESOURCES USED BY APPLICATIONS IN A SPECIFIC NAMESPACE.

STEP 1: UNDERSTAND RESOURCE QUOTAS

RESOURCE QUOTAS ALLOW YOU TO:

- LIMIT THE AMOUNT OF CPU AND MEMORY A NAMESPACE CAN USE.
- CONTROL THE NUMBER OF CERTAIN TYPES OF RESOURCES (E.G., PODS, SERVICES, PERSISTENTVOLUMECLAIMS) IN A NAMESPACE.
- PREVENT A NAMESPACE FROM CONSUMING MORE RESOURCES THAN ALLOCATED, ENSURING FAIR USAGE ACROSS MULTIPLE TEAMS OR APPLICATIONS.

STEP 2: CREATE A NAMESPACE

FIRST, CREATE A NAMESPACE WHERE YOU WILL APPLY THE RESOURCE QUOTA. THIS HELPS IN ISOLATING AND CONTROLLING RESOURCE USAGE WITHIN THAT SPECIFIC NAMESPACE.

CREATE A YAML FILE NAMED **QUOTA-NAMESPACE.YAML** WITH THE FOLLOWING CONTENT:

```
APIVERSION: v1
KIND: NAMESPACE
METADATA:
  NAME: myns
```

APPLY THE YAML TO CREATE THE NAMESPACE:

```
KUBECTL APPLY -F QUOTA-NAMESPACE.YAML
PS D:\Coding\ClassWork\k8s> kubectl apply -f quota-namespace.yaml
namespace/myns created
```

VERIFY THAT THE NAMESPACE IS CREATED:

```
KUBECTL GET NAMESPACES
PS D:\Coding\ClassWork\k8s> kubectl get namespaces
NAME          STATUS  AGE
default        Active  14m
kube-node-lease  Active  14m
kube-public    Active  14m
kube-system    Active  14m
kubernetes-dashboard  Active  13m
myns          Active  13s
```

YOU SHOULD SEE QUOTA-EXAMPLE LISTED IN THE OUTPUT.

STEP 3: DEFINE A RESOURCE QUOTA

NEXT, CREATE A RESOURCE QUOTA YAML FILE NAMED **RESOURCE-QUOTA.YAML** WITH THE FOLLOWING CONTENT:

```
APIVERSION: v1
KIND: RESOURCEQUOTA
METADATA:
  NAME: MYNS-QUOTA  # THE NAME OF THE RESOURCE QUOTA.
  NAMESPACE: MYNS # THE NAMESPACE TO WHICH THE RESOURCE QUOTA WILL APPLY.
SPEC:
  HARD:
    REQUESTS.CPU: "2"  # THE TOTAL CPU RESOURCE REQUESTS ALLOWED IN THE NAMESPACE (2 CORES).
    REQUESTS.MEMORY: "4Gi" # THE TOTAL MEMORY RESOURCE REQUESTS ALLOWED IN THE NAMESPACE (4 GiB).
    LIMITS.CPU: "4"      # THE TOTAL CPU RESOURCE LIMITS ALLOWED IN THE NAMESPACE (4 CORES).
    LIMITS.MEMORY: "8Gi" # THE TOTAL MEMORY RESOURCE LIMITS ALLOWED IN THE NAMESPACE (8 GiB).
  PODS: "10"          # THE TOTAL NUMBER OF PODS ALLOWED IN THE NAMESPACE.
  PERSISTENTVOLUMECLAIMS: "5" # THE TOTAL NUMBER OF PERSISTENTVOLUMECLAIMS ALLOWED IN THE NAMESPACE.
  CONFIGMAPS: "10"    # THE TOTAL NUMBER OF CONFIGMAPS ALLOWED IN THE NAMESPACE.
  SERVICES: "5"      # THE TOTAL NUMBER OF SERVICES ALLOWED IN THE NAMESPACE.
```

STEP 4: APPLY THE RESOURCE QUOTA

APPLY THE RESOURCE QUOTA YAML TO THE NAMESPACE:

```
KUBECTL APPLY -F RESOURCE-QUOTA.YAML
PS D:\Coding\ClassWork\k8s> kubectl apply -f resource-quota.yaml
resourcequota/myns-quota created
```

VERIFY THAT THE RESOURCE QUOTA IS APPLIED:

```
KUBECTL GET RESOURCEQUOTA -N MYNS
PS D:\Coding\ClassWork\k8s> kubectl get resourcequota -n myns
NAME      REQUEST          AGE          LIMIT
myns-quota  configmaps: 1/10, persistentvolumeclaims: 0/5, pods: 0/10, requests.cpu: 0/2, requests.memory: 0/4Gi, services: 0/5  limits.cpu: 0/4, limits.memory: 0/8Gi  14s
```

TO SEE THE DETAILS OF THE APPLIED RESOURCE QUOTA:

```
KUBECTL DESCRIBE RESOURCEQUOTA MYNS-QUOTA -N MYNS
PS D:\Coding\ClassWork\k8s> kubectl describe resourcequota myns-quota -n myns
Name:          myns-quota
Namespace:     myns
Resource       Used   Hard
-----
configmaps    1      10
limits.cpu    0      4
limits.memory 0      8Gi
persistentvolumeclaims 0      5
pods          0      10
requests.cpu  0      2
requests.memory 0      4Gi
services      0      5
```

STEP 5: TEST THE RESOURCE QUOTA

LET'S CREATE SOME RESOURCES IN THE QUOTA-EXAMPLE NAMESPACE TO SEE HOW THE RESOURCE QUOTA AFFECTS THEM.

DEPLOY A REPLICASET WITH RESOURCE REQUESTS AND LIMITS

CREATE A YAML FILE NAMED **NGINX-REPLICASET-QUOTA.YAML** WITH THE FOLLOWING CONTENT:

```
APIVERSION: APPS/V1
KIND: REPLICASET
METADATA:
  NAME: NGINX-REPLICASET
  NAMESPACE: MYNS
SPEC:
  REPLICAS: 5      # DESIRED NUMBER OF POD REPLICAS.
  SELECTOR:
    MATCHLABELS:
      APP: NGINX
  TEMPLATE:
    METADATA:
      LABELS:
```

```
APP: NGINX

SPEC:

CONTAINERS:

- NAME: NGINX

IMAGE: NGINX:LATEST

PORTS:

- CONTAINERPORT: 80

RESOURCES:      # DEFINE RESOURCE REQUESTS AND LIMITS.

REQUESTS:

MEMORY: "100Mi"

CPU: "100M"

LIMITS:

MEMORY: "200Mi"

CPU: "200M"
```

EXPLANATION:

THIS REPLICASET REQUESTS A TOTAL OF 500M CPU AND 500Mi MEMORY ACROSS 5 REPLICAS.

IT ALSO LIMITS EACH REPLICA TO USE A MAXIMUM OF 200M CPU AND 200Mi MEMORY.

APPLY THIS YAML TO CREATE THE REPLICASET:

```
KUBECTL APPLY -F NGINX-REPLICASET-QUOTA.YAML
```

```
PS D:\Coding\ClassWork\k8s> kubectl apply -f nginx-replicaset-quota.yaml
replicaset.apps/nginx-replicaset created
```

CHECK THE STATUS OF THE PODS AND ENSURE THEY ARE CREATED WITHIN THE CONSTRAINTS OF THE RESOURCE QUOTA:

```
KUBECTL GET PODS -N MYNS
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-replicaset-8614t	0/1	ContainerCreating	0	14s
nginx-replicaset-97858	1/1	Running	0	14s
nginx-replicaset-vmxx5	0/1	ContainerCreating	0	14s
nginx-replicaset-vqjtz	1/1	Running	0	14s
nginx-replicaset-zmrssx	1/1	Running	0	14s

TO DESCRIBE THE PODS AND SEE THEIR RESOURCE ALLOCATIONS:

KUBECTL DESCRIBE PODS -L APP=NGINX -N QUOTA-EXAMPLE

```
PS D:\Coding\ClassWork\k8s> kubectl describe pods -l app=nginx -n myns
Name:           nginx-replicaset-8614t
Namespace:      myns
Priority:       0
Service Account: default
Node:           minikube/172.28.227.29
Start Time:     Mon, 23 Feb 2026 10:44:20 +0530
Labels:         app=nginx
Annotations:   <none>
Status:         Running
IP:             10.244.0.9
IPs:
  IP:           10.244.0.9
Controlled By: ReplicaSet/nginx-replicaset
Containers:
  nginx:
    Container ID:  docker://bcae1799b4103d4ad42dc4da2c1dbb5cd4609886a07a2319129d2e7d6ab72989
    Image:         nginx:latest
    Image ID:     docker-pullable://nginx@sha256:341bf0f3ce6c5277d6002cf6e1fb0319fa4252add24ab6a0e262e0056d313208
    Port:          80/TCP
    Host Port:    0/TCP
    State:         Running
    Started:      Mon, 23 Feb 2026 10:44:35 +0530
```

ATTEMPT TO EXCEED THE RESOURCE QUOTA

TRY CREATING ADDITIONAL RESOURCES TO SEE IF THEY ARE REJECTED WHEN EXCEEDING THE QUOTA. FOR EXAMPLE, CREATE MORE PODS OR INCREASE THE CPU/MEMORY REQUESTS TO EXCEED THE QUOTA LIMITS.

CREATE A YAML FILE NAMED **NGINX-EXTRA-POD.YAML** WITH THE FOLLOWING CONTENT:

```
APIVersion: v1
Kind: Pod
Metadata:
  Name: nginx-extra-pod
  Namespace: myns
```

SPEC:

CONTAINERS:

- NAME: NGINX

IMAGE: NGINX:LATEST

RESOURCES:

REQUESTS:

MEMORY: "3Gi" # REQUESTS A LARGE AMOUNT OF MEMORY.

CPU: "2" # REQUESTS A LARGE AMOUNT OF CPU.

LIMITS:

MEMORY: "4Gi"

CPU: "2"

APPLY THIS YAML TO CREATE THE POD:

KUBECTL APPLY -F NGINX-EXTRA-POD.YAML

```
PS D:\Coding\ClassWork\k8s> kubectl apply -f nginx-extra-pod.yaml
Error from server (Forbidden): error when creating "nginx-extra-pod.yaml": pods "nginx-extra-pod" is forbidden: exceeded quota: myns-quota, requested: requests.cpu=2, used: requests.cpu=500m, limited: requests.cpu=2
```

THIS SHOULD FAIL DUE TO EXCEEDING THE RESOURCE QUOTA. CHECK THE EVENTS TO SEE THE FAILURE REASON:

KUBECTL GET EVENTS -N QUOTA-EXAMPLE

LAST SEEN	TYPE	REASON	OBJECT	MESSAGE
6m48s	Normal	Scheduled	pod/nginx-replicaset-8614t	Successfully assigned myns/nginx-replicaset-8614t to minikube
6m48s	Normal	Pulling	pod/nginx-replicaset-8614t	Pulling image "nginx:latest"
6m33s	Normal	Pulled	pod/nginx-replicaset-8614t	Successfully pulled image "nginx:latest" in 3.588s (15.187s including waiting). Image size: 160850673 bytes.
6m33s	Normal	Created	pod/nginx-replicaset-8614t	Container created
6m33s	Normal	Started	pod/nginx-replicaset-8614t	Container started
6m48s	Normal	Scheduled	pod/nginx-replicaset-97858	Successfully assigned myns/nginx-replicaset-97858 to minikube
6m48s	Normal	Pulling	pod/nginx-replicaset-97858	Pulling image "nginx:latest"
6m44s	Normal	Pulled	pod/nginx-replicaset-97858	Successfully pulled image "nginx:latest" in 3.993s (3.993s including waiting). Image size: 160850673 bytes.
6m44s	Normal	Created	pod/nginx-replicaset-97858	Container created
6m44s	Normal	Started	pod/nginx-replicaset-97858	Container started
6m48s	Normal	Scheduled	pod/nginx-replicaset-vmxx5	Successfully assigned myns/nginx-replicaset-vmxx5 to minikube
6m48s	Normal	Pulling	pod/nginx-replicaset-vmxx5	Pulling image "nginx:latest"
6m29s	Normal	Pulled	pod/nginx-replicaset-vmxx5	Successfully pulled image "nginx:latest" in 3.615s (18.792s including waiting). Image size: 160850673 bytes.
6m29s	Normal	Created	pod/nginx-replicaset-vmxx5	Container created
6m29s	Normal	Started	pod/nginx-replicaset-vmxx5	Container started
6m48s	Normal	Scheduled	pod/nginx-replicaset-vqjtz	Successfully assigned myns/nginx-replicaset-vqjtz to minikube
6m48s	Normal	Pulling	pod/nginx-replicaset-vqjtz	Pulling image "nginx:latest"
6m40s	Normal	Pulled	pod/nginx-replicaset-vqjtz	Successfully pulled image "nginx:latest" in 3.929s (7.826s including waiting). Image size: 160850673 bytes.
6m40s	Normal	Created	pod/nginx-replicaset-vqjtz	Container created
6m40s	Normal	Started	pod/nginx-replicaset-vqjtz	Container started
6m48s	Normal	Scheduled	pod/nginx-replicaset-zmrsx	Successfully assigned myns/nginx-replicaset-zmrsx to minikube
6m48s	Normal	Pulling	pod/nginx-replicaset-zmrsx	Pulling image "nginx:latest"
6m36s	Normal	Pulled	pod/nginx-replicaset-zmrsx	Successfully pulled image "nginx:latest" in 3.786s (11.599s including waiting). Image size: 160850673 bytes.
6m36s	Normal	Created	pod/nginx-replicaset-zmrsx	Container created
6m36s	Normal	Started	pod/nginx-replicaset-zmrsx	Container started
6m48s	Normal	SuccessfulCreate	replicaset/nginx-replicaset	Created pod: nginx-replicaset-97858
6m48s	Normal	SuccessfulCreate	replicaset/nginx-replicaset	Created pod: nginx-replicaset-vqjtz
6m48s	Normal	SuccessfulCreate	replicaset/nginx-replicaset	Created pod: nginx-replicaset-zmrsx
6m48s	Normal	SuccessfulCreate	replicaset/nginx-replicaset	Created pod: nginx-replicaset-vmxx5
6m48s	Normal	SuccessfulCreate	replicaset/nginx-replicaset	Created pod: nginx-replicaset-8614t

LOOK FOR ERROR MESSAGES INDICATING THAT THE POD CREATION WAS DENIED DUE TO RESOURCE CONSTRAINTS.

STEP 6: CLEAN UP RESOURCES

TO DELETE THE RESOURCES YOU CREATED:

```
KUBECTL DELETE -f NGINX-REPLICASET-QUOTA.YAML
```

```
KUBECTL DELETE -f NGINX-EXTRA-POD.YAML
```

```
KUBECTL DELETE -f RESOURCE-QUOTA.YAML
```

```
KUBECTL DELETE NAMESPACE MYNS
```

```
PS D:\Coding\ClassWork\k8s> kubectl delete -f nginx-replicaset-quota.yaml
replicaset.apps "nginx-replicaset" deleted from myns namespace
PS D:\Coding\ClassWork\k8s> kubectl delete -f nginx-extra-pod.yaml
Error from server (NotFound): error when deleting "nginx-extra-pod.yaml": pods "nginx-extra-pod" not found
PS D:\Coding\ClassWork\k8s> kubectl delete -f resource-quota.yaml
resourcequota "myns-quota" deleted from myns namespace
PS D:\Coding\ClassWork\k8s> kubectl delete namespace myns
namespace "myns" deleted
```