



Exercise 9.4: Using a ResourceQuota to Limit PVC Count and Usage

The flexibility of cloud-based storage often requires limiting consumption among users. We will use the ResourceQuota object to both limit the total consumption as well as the number of persistent volume claims.

1. Begin by deleting the deployment we had created to use NFS, the pv and the pvc.

```
student@lfs458-node-1a0a:~$ kubectl delete deploy nginx-nfs
deployment "nginx-nfs" deleted

student@lfs458-node-1a0a:~$ kubectl delete pvc pvc-one
persistentvolumeclaim "pvc-one" deleted

student@lfs458-node-1a0a:~$ kubectl delete pv pvvol-1
persistentvolume "pvvol-1" deleted
```

2. Create a yaml file for the ResourceQuota object. Set the storage limit to ten claims with a total usage of 500Mi.

```
student@lfs458-node-1a0a:~$ vim storage-quota.yaml

apiVersion: v1
kind: ResourceQuota
metadata:
  name: storagequota
spec:
  hard:
    persistentvolumeclaims: "10"
    requests.storage: "500Mi"
```

3. Create a new namespace called small. View the namespace information prior to the new quota. Either the long name with double dashes --namespace or the nickname ns work for the resource.

```
student@lfs458-node-1a0a:~$ kubectl create --namespace small
namespace "small" created

student@lfs458-node-1a0a:~$ kubectl describe ns small
Name:                small
Labels:              <none>
Annotations:         <none>
Status:              Active

No resource quota.

No resource limits.
```

4. Create a new pv and pvc in the small namespace.

```
student@lfs458-node-1a0a:~$ kubectl create -f PVol.yaml -n small
persistentvolume "pvvol-1" created

student@lfs458-node-1a0a:~$ kubectl create -f pvc.yaml -n small
persistentvolumeclaim "pvc-one" created
```

5. Create the new resource quota, placing this object into the low-usage-limit namespace.

```
student@lfs458-node-1a0a:~$ kubectl create -f storage-quota.yaml \
-n small
resourcequota "storagequota" created
```

6. Verify the `small` namespace has quotas. Compare the output to the same command above.

```
student@lfs458-node-1a0a:~$ kubectl describe ns small
Name:          small
Labels:        <none>
Annotations:   <none>
Status:        Active
```

Resource Quotas

```
Name:          storagequota
Resource       Used   Hard
-----
persistentvolumeclaims 1    10
requests.storage  200Mi 500Mi
```

No resource limits.

7. Remove the namespace line from the `nfs-pod.yaml` file. Should be around line 11 or so. This will allow us to pass other namespaces on the command line.

```
student@lfs458-node-1a0a:~$ vim nfs-pod.yaml
```

8. Create the container again.

```
student@lfs458-node-1a0a:~$ kubectl create -f nfs-pod.yaml \
-n small
deployment "nginx-nfs" created
```

9. Determine if the deployment has a running pod.

```
student@lfs458-node-1a0a:~$ kubectl get deploy --namespace=small
NAME          DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx-nfs     1         1         1             0           43s

student@lfs458-node-1a0a:~$ kubectl describe deploy nginx-nfs \
-n small
<output_omitted>
```

10. Look to see if the pods are ready.

```
student@lfs458-node-1a0a:~$ kubectl get po --namespace=small
NAME                                READY   STATUS    RESTARTS   AGE
nginx-nfs-2854978848-g3khf         1/1     Running   0           37s
```

11. Ensure the Pod is running and is using the NFS mounted volume.

```
student@lfs458-node-1a0a:~$ kubectl describe po \
nginx-nfs-2854978848-g3khf -n small
Name:          nginx-nfs-2854978848-g3khf
Namespace:     small
<output_omitted>

Mounts:
  /opt from nfs-vol (rw)
<output_omitted>
```

12. View the quota usage of the namespace

```
student@lfs458-node-1a0a:~$ kubectl describe ns small
<output_omitted>
```

```
Resource Quotas
Name:
Resource          storagequota
Used              Hard
-----
persistentvolumeclaims 1      10
requests.storage      200Mi 500Mi
```

```
No resource limits.
```

13. Create a 300M file inside of the `/opt/sfw` directory on the host and view the quota usage again. Note that with NFS the size of the share is not counted against the deployment.

```
student@lfs458-node-1a0a:~$ sudo dd if=/dev/zero \
    of=/opt/sfw/bigfile bs=1M count=300
300+0 records in
300+0 records out
314572800 bytes (315 MB, 300 MiB) copied, 0.196794 s, 1.6 GB/s
```

```
student@lfs458-node-1a0a:~$ kubectl describe ns small
<output_omitted>
Resource Quotas
Name:
Resource          storagequota
Used              Hard
-----
persistentvolumeclaims 1      10
requests.storage      200Mi 500Mi
<output_omitted>
```

```
student@lfs458-node-1a0a:~$ du -h /opt/
301M    /opt/sfw
41M     /opt/cni/bin
41M     /opt/cni
341M    /opt/
```

14. Now let us illustrate what happens when a deployment requests more than the quota. Begin by shutting down the existing deployment.

```
student@lfs458-node-1a0a:~$ kubectl get deploy -n small
NAME          DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
nginx-nfs     1        1        1            1          11m

student@lfs458-node-1a0a:~$ kubectl delete deploy nginx-nfs -n small
deployment "nginx-nfs" deleted
```

15. Once the Pod has shut down view the resource usage of the namespace again. Note the storage did not get cleaned up when the pod was shut down.

```
student@lfs458-node-1a0a:~$ kubectl describe ns small
<output_omitted>
Resource Quotas
Name:
Resource          storagequota
Used              Hard
-----
persistentvolumeclaims 1      10
requests.storage      200Mi 500Mi
```

16. Remove the pvc then view the pv it was using. Note the RECLAIM POLICY and STATUS.

```
student@lfs458-node-1a0a:~$ kubectl get pvc -n small
NAME      STATUS    VOLUME   CAPACITY   ACCESSMODES   STORAGECLASS   AGE
pvc-one   Bound     pvvol-1   1Gi        RWX            small           19m
```

```
student@lfs458-node-1a0a:~$ kubectl delete pvc pvc-one -n small
persistentvolumeclaim "pvc-one" deleted
```

```
student@lfs458-node-1a0a:~$ kubectl get pv -n small
NAME      CAPACITY   ACCESSMODES   RECLAIMPOLICY   STATUS   CLAIM
STORAGECLASS   REASON   AGE
pvvol-1   1Gi        RWX           Retain          Released   small/pvc-one   44m
```

17. Dynamically provisioned storage uses the ReclaimPolicy of the StorageClass which could be Delete, Retain, or some types allow Recycle. Manually created persistent volumes default to Retain unless set otherwise at creation. The default storage policy is to retain the storage to allow recovery of any data. To change this begin by viewing the yaml output.

```
student@lfs458-node-1a0a:~$ kubectl get pv/pvvol-1 -o yaml
....
  path: /opt/sfw
  server: lfs458-node-1a0a
  persistentVolumeReclaimPolicy: Retain
status:
  phase: Released
```

18. Currently we will need to delete and re-create the object. Future development on a deleter plugin is planned. We will re-create the volume and allow it to use the Retain policy, then change it once running.

```
student@lfs458-node-1a0a:~$ kubectl delete pv/pvvol-1
persistentvolume "pvvol-1" deleted

student@lfs458-node-1a0a:~$ grep Retain PVol.yaml
  persistentVolumeReclaimPolicy: Retain

student@lfs458-node-1a0a:~$ kubectl create -f PVol.yaml
persistentvolume "pvvol-1" created
```

19. We will use kubectl patch to change the retention policy to Delete. The yaml output from before can be helpful in getting the correct syntax.

```
student@lfs458-node-1a0a:~$ kubectl patch pv pvvol-1 -p \
'{"spec":{"persistentVolumeReclaimPolicy":"Delete"}}'
persistentvolume "pvvol-1" patched

student@lfs458-node-1a0a:~$ kubectl get pv/pvvol-1
NAME      CAPACITY   ACCESSMODES   RECLAIMPOLICY   STATUS   CLAIM
STORAGECLASS   REASON   AGE
pvvol-1   1Gi        RWX           Delete          Available   2m
```

20. View the current quota settings.

```
student@lfs458-node-1a0a:~$ kubectl describe ns small
.
requests.storage      0      500Mi
```

21. Create the pvc again. Even with no pods running, note the resource usage.

```
student@lfs458-node-1a0a:~$ kubectl create -f pvc.yaml -n small
persistentvolumeclaim "pvc-one" created

student@lfs458-node-1a0a:~$ kubectl describe ns small
.
requests.storage      200Mi      500Mi
```

22. Remove the existing quota from the namespace.

```
student@lfs458-node-1a0a:~$ kubectl get resourcequota -n small
NAME          AGE
storagequota  12m

student@lfs458-node-1a0a:~$ kubectl delete resourcequota \
    storagequota -n small
resourcequota "storagequota" deleted
```

23. Edit the storagequota.yaml file and lower the capacity to 100Mi.

```
student@lfs458-node-1a0a:~$ vim storage-quota.yaml

requests.storage: "100Mi"
```

24. Create and verify the new storage quota. Note the hard limit has already been exceeded.

```
student@lfs458-node-1a0a:~$ kubectl create -f storage-quota.yaml -n small
resourcequota "storagequota" created

student@lfs458-node-1a0a:~$ kubectl describe ns small
.
persistentvolumeclaims      1      10
requests.storage             200Mi   100Mi

No resource limits.
```

25. Create the deployment again. View the deployment. Note there are no errors seen.

```
student@lfs458-node-1a0a:~$ kubectl create -f nfs-pod.yaml \
    --namespace=small
deployment "nginx-nfs" created

student@lfs458-node-1a0a:~$ kubectl describe deploy/nginx-nfs -n small
Name:                nginx-nfs
Namespace:            small
<output_omitted>
```

26. Examine the pods to see if they are actually running.

```
student@lfs458-node-1a0a:~$ kubectl get po -n small
NAME                                READY    STATUS    RESTARTS   AGE
nginx-nfs-2854978848-vb6bh          1/1      Running   0           58s
```

27. As we were able to deploy more pods even with apparent hard quota set, let us test to see if the reclaim of storage takes place. Remove the deployment and the persistent volume claim.

```
student@lfs458-node-1a0a:~$ kubectl delete deploy nginx-nfs -n small
deployment "nginx-nfs" deleted

student@lfs458-node-1a0a:~$ kubectl delete pvc/pvc-one -n small
persistentvolumeclaim "pvc-one" deleted
```

28. View if the persistent volume exists. You will see it attempted a removal, but failed. If you look closer you will find the error has to do with the lack of a deleter volume plugin for NFS. Other storage protocols have a plugin.

```
student@lfs458-node-1a0a:~$ kubectl get pv -n small
NAME          CAPACITY  ACCESSMODES  RECLAIMPOLICY  STATUS  CLAIM
STORAGECLASS  REASON    AGE
pvvol-1       1Gi       RWX          Delete         Failed  small/pvc-one  20m
```

29. Ensure the deployment, pvc and pv are all removed.

```
student@lfs458-node-1a0a:~$ kubectl delete pv/pvvol-1
persistentvolume "pvvol-1" deleted
```

30. Edit the persistent volume YAML file and change the persistentVolumeReclaimPolicy: to Recycle.

```
student@lfs458-node-1a0a:~$ vim PVol.yaml
....
  persistentVolumeReclaimPolicy: Recycle
....
```

31. Add a LimitRange to the namespace and attempt to create the persistent volume and persistent volume claim again. We can use the LimitRange we used earlier.

```
student@lfs458-node-1a0a:~$ kubectl create -f low-resource-range.yaml\
-n small
limitrange "low-resource-range" created
```

32. View the settings for the namespace. Both quotas and resource limits should be seen.

```
student@lfs458-node-1a0a:~$ kubectl describe ns small
<output_omitted>
Resource Limits

```

Type	Resource	Min	Max	Default Request	Default Limit	...
Container	cpu	-	-	500m	1	-
Container	memory	-	-	100Mi	500Mi	-

33. Create the persistent volume again. View the resource. Note the Reclaim Policy is Recycle.

```
student@lfs458-node-1a0a:~$ kubectl create -f PVol.yaml -n small
persistentvolume "pvvol-1" created

student@lfs458-node-1a0a:~$ kubectl get pv

```

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	...
pvvol-1	1Gi	RWX	Recycle	Available	...

34. Attempt to create the persistent volume claim again. The quota only takes effect if there is also a resource limit in effect.

```
student@lfs458-node-1a0a:~$ kubectl create -f pvc.yaml -n small
Error from server (Forbidden): error when creating "pvc.yaml":
  persistentvolumeclaims "pvc-one" is forbidden: exceeded quota:
storagequota, requested: requests.storage=200Mi, used:
requests.storage=0, limited: requests.storage=100Mi
```

35. Edit the resourcequota to increase the requests.storage to 500mi.

```
student@lfs458-node-1a0a:~$ kubectl edit resourcequota -n small
....
spec:
  hard:
    persistentvolumeclaims: "10"
    requests.storage: 500Mi
status:
  hard:
    persistentvolumeclaims: "10"
....
```

36. Create the pvc again. It should work this time. Then create the deployment again.

```
student@lfs458-node-1a0a:~$ kubectl create -f pvc.yaml -n small
persistentvolumeclaim "pvc-one" created
```

```
student@lfs458-node-1a0a:~$ kubectl create -f nfs-pod.yaml -n small
deployment "nginx-nfs" created
```

37. View the namespace settings.

```
student@lfs458-node-1a0a:~$ kubectl describe ns small
<output_omitted>
```

38. Delete the deployment. View the status of the pv and pvc.

```
student@lfs458-node-1a0a:~$ kubectl delete deploy nginx-nfs -n small
deployment "nginx-nfs" deleted
```

```
student@lfs458-node-1a0a:~$ kubectl get pvc -n small
NAME      STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
pvc-one   Bound     pvvol-1   1Gi        RWX              small           7m
```

```
student@lfs458-node-1a0a:~$ kubectl get pv -n small
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM   STORAGECLASS   AGE
pvvol-1   1Gi        RWX             Recycle          Bound    small/pvc-one   ...
```

39. Delete the pvc and check the status of the pv. It should show as Available.

```
student@lfs458-node-1a0a:~$ kubectl delete pvc pvc-one -n small
persistentvolumeclaim "pvc-one" deleted
```

```
student@lfs458-node-1a0a:~$ kubectl get pv -n small
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM   STORAGECLASS   AGE
pvvol-1   1Gi        RWX             Recycle          Available    ...
```

40. Remove the pv and any other resources created during this lab.

```
student@lfs458-node-1a0a:~$ kubectl delete pv pvvol-1
persistentvolume "pvvol-1" deleted
```