Table of Contents

* [Compute Resource Management Lab](https://www.opentlc.com/labs/ocp_advanced_deployment/04_2_Managing_Compute_Resources_Solution_Lab.html#_compute_resource_management_lab)
* [1. Create Limit Range](https://www.opentlc.com/labs/ocp_advanced_deployment/04_2_Managing_Compute_Resources_Solution_Lab.html#labexercises)
* [2. Clean Up Environment](https://www.opentlc.com/labs/ocp_advanced_deployment/04_2_Managing_Compute_Resources_Solution_Lab.html#_clean_up_environment)

Compute Resource Management Lab

**Goals**

* Create a **LimitRange** object
* Create a Horizontal Pod Autoscaler (HPA)
* Test the HPA

|  |  |
| --- | --- |
|  | You must have have metrics working in your cluster to create an HPA. |

1. Create Limit Range

1. Create a new project:

oc new-project test-hpa

1. Deploy **hello-openshift** in the new project:
2. oc new-app openshift/hello-openshift:v3.9 -n test-hpa

oc expose svc hello-openshift

1. Create a **LimitRange** object with the following properties:

|  |  |
| --- | --- |
| **Pod Limits** | |
| Min CPU | 10m |
| Max CPU | 100m |
| Min Memory | 5Mi |
| Max Memory | 750Mi |

|  |  |
| --- | --- |
| **Container Limits** | |
| Min CPU | 10m |
| Max CPU | 100m |
| Min Memory | 5Mi |
| Max Memory | 750Mi |
| Default CPU | 50m |
| Default Memory | 100Mi |

1. Confirm that your code looks similar to this:
2. echo '{
3. "kind": "LimitRange",
4. "apiVersion": "v1",
5. "metadata": {
6. "name": "limits",
7. "creationTimestamp": null
8. },
9. "spec": {
10. "limits": [
11. {
12. "type": "Pod",
13. "max": {
14. "cpu": "100m",
15. "memory": "750Mi"
16. },
17. "min": {
18. "cpu": "10m",
19. "memory": "5Mi"
20. }
21. },
22. {
23. "type": "Container",
24. "max": {
25. "cpu": "100m",
26. "memory": "750Mi"
27. },
28. "min": {
29. "cpu": "10m",
30. "memory": "5Mi"
31. },
32. "default": {
33. "cpu": "50m",
34. "memory": "100Mi"
35. }
36. }
37. ]
38. }

}' | oc create -f - -n test-hpa

1. Create an HPA for the **hello-openshift** deployment to scale between one and five replicas and set it to scale up when the CPU utilization reaches 80%.

oc autoscale dc/hello-openshift --min 1 --max 5 --cpu-percent=80

1. List the status of the autoscaler:

oc get hpa/hello-openshift -n test-hpa

**Sample Output**

NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE

hello-openshift DeploymentConfig/hello-openshift <unknown> / 80% 1 5 1 2h

1. Review the autoscaler information:

oc describe hpa/hello-openshift -n test-hpa

**Sample Output**

Name: hello-openshift

Namespace: test-hpa

Labels: <none>

Annotations: <none>

CreationTimestamp: Thu, 19 Apr 2018 18:12:19 +0000

Reference: DeploymentConfig/hello-openshift

Metrics: ( current / target )

resource cpu on pods (as a percentage of request): <unknown> / 80%

Min replicas: 1

Max replicas: 5

Conditions:

Type Status Reason Message

---- ------ ------ -------

AbleToScale True SucceededGetScale the HPA controller was able to get the target's current scale

ScalingActive False FailedGetResourceMetric the HPA was unable to compute the replica count: missing request for cpu on container hello-openshift in pod test-hpa/hello-openshift-1-fc9l6

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Warning FailedGetResourceMetric 2s (x3 over 1m) horizontal-pod-autoscaler missing request for cpu on container hello-openshift in pod test-hpa/hello-openshift-1-fc9l6

Warning FailedComputeMetricsReplicas 2s (x3 over 1m) horizontal-pod-autoscaler failed to get cpu utilization: missing request for cpu on container hello-openshift in pod test-hpa/hello-openshift-1-fc9l6

1. Fix the error displayed in the output above by rolling out the latest deployment configuration of **hello-openshift**. It will read the defaults in the LimitRange.
   1. Redeploy the application to pick up the default request from the **LimitRange** object:

oc rollout latest hello-openshift -n test-hpa

|  |  |
| --- | --- |
|  | It takes several minutes for the HPA to collect enough metrics to present a current status. Generating load later in this lab will also produce a current Metrics. |

1. In a separate window, create work for the pod and monitor the environment:
2. ROUTE=$(oc get route hello-openshift --template "{{ .spec.host }}")
3. for time in {1..15000}
4. do
5. echo time $time
6. curl ${ROUTE}

done

1. Run this command in a few windows concurrently to produce more work for the pods.
   * You need at least three parallel windows to generate enough load.
2. Examine the HPA information to see the effect of the workload.
3. Use the OpenShift Container Platform web console to see whether new pods are being created.

|  |  |
| --- | --- |
|  | The HPA does not scale up or down instantly. |

2. Clean Up Environment

1. Remove the HPA project:

oc delete project test-hpa

Build Version: c3147ce9f77191e30b447cc423f2f68a0c40fc03 : Last updated 2018-07-31 01:29:02 EDT