

Task

1. login to your cluster and create a new namespace with the name mem-example

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ k top node
NAME                                CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
cka-cluster2-control-plane          326m         4%     827Mi           21%
cka-cluster2-worker                 95m          1%     302Mi           7%
cka-cluster2-worker2                70m          0%     193Mi           5%

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ k create ns mem-example
namespace/mem-example created
```

2. Install metrics server using the yaml provided in this repo

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ k apply -f day-16/metrics-server.yaml
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
service/metrics-server created
deployment.apps/metrics-server created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ k get pods -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
coredns-7db6d8ff4d-rtcck            1/1     Running   1 (32h ago)  2d8h
coredns-7db6d8ff4d-wwjdf            1/1     Running   1 (32h ago)  2d8h
etcd-cka-cluster2-control-plane      1/1     Running   1 (32h ago)  2d8h
kindnet-9t5t1                       1/1     Running   1 (32h ago)  2d8h
kindnet-ftznj                       1/1     Running   1 (32h ago)  2d8h
kindnet-x2fwv                       1/1     Running   1 (32h ago)  2d8h
kube-apiserver-cka-cluster2-control-plane 1/1     Running   1 (32h ago)  2d8h
kube-controller-manager-cka-cluster2-control-plane 1/1     Running   15 (6h35m ago)  2d8h
kube-proxy-dbz65                    1/1     Running   1 (32h ago)  2d8h
kube-proxy-dlkc8                    1/1     Running   1 (32h ago)  2d8h
kube-proxy-m52lb                    1/1     Running   1 (32h ago)  2d8h
kube-scheduler-cka-cluster2-control-plane 1/1     Running   5 (6h33m ago)  25h
metrics-server-55677cdb4c-9mms8      1/1     Running   0           57s
```

3. Perform the steps given in the below doc:

<https://kubernetes.io/docs/tasks/configure-pod-container/assign-memory-resource/#specify-a-memory-request-and-a-memory-limit>

- Specify a memory request and a memory limit (you create a Pod that has one Container. The Container has a memory request of 100 MiB and a memory limit of 200 MiB.)

```
Day-16 > ! mem-request.yaml > {} metadata > namespace
io.k8s.api.core.v1.Pod (v1@pod.json)
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: memory-demo
5    namespace: mem-example
6  spec:
7    containers:
8    - name: memory-demo-ctr
9      image: polinux/stress
10     resources:
11       requests:
12         memory: "100Mi"
13       limits:
14         memory: "200Mi"
15     command: ["stress"]
16     args: ["--vm", "1", "--vm-bytes", "150M", "--vm-hang", "1"]
17
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ k apply -f day-16/mem-request.yaml
pod/memory-demo created
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ k get pod -n mem-example
```

NAME	READY	STATUS	RESTARTS	AGE
memory-demo	0/1	ContainerCreating	0	10s

- Exceed a Container's memory limit (you create a Pod that attempts to allocate more memory than its limit. Here is the configuration YAML file for a Pod that has one Container with a memory request of 50 MiB and a memory limit of 100 MiB)

```
day-16 > ! mem-2.yaml > {} spec > [ ] containers > {} 0
io.k8s.api.core.v1.Pod (v1@pod.json)
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: memory-demo-2
5    namespace: mem-example
6  spec:
7    containers:
8      - name: memory-demo-2-ctr
9        image: polinux/stress
10       resources:
11         requests:
12           memory: "50Mi"
13         limits:
14           memory: "100Mi"
15         command: ["stress"]
16         args: ["--vm", "1", "--vm-bytes", "250M", "--vm-hang", "1"]
17
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ k apply -f day-16/mem-2.yaml
pod/memory-demo-2 created
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ k get pod -n mem-example
```

NAME	READY	STATUS	RESTARTS	AGE
memory-demo	1/1	Running	0	6m34s
memory-demo-2	0/1	Error	1 (5s ago)	9s

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ k get pod -n mem-example
```

NAME	READY	STATUS	RESTARTS	AGE
memory-demo	1/1	Running	0	9m19s
memory-demo-2	0/1	CrashLoopBackOff	4 (69s ago)	2m54s

- Specify a memory request that is too big for your Nodes (create a Pod that has a memory request so big that it exceeds the capacity of any Node in your cluster. Here is the configuration file for a Pod that has one Container with a request for 1000 GiB of memory, which likely exceeds the capacity of any Node in your cluster.)

```
Day-16 > ! mem-3.yaml > {} spec > [ ] containers > {} 0
```

```
io.k8s.api.core.v1.Pod (v1@pod.json)
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: memory-demo-3
5    namespace: mem-example
6  spec:
7    containers:
8      - name: memory-demo-3-ctr
9        image: polinux/stress
10       resources:
11         requests:
12           memory: "1000Gi"
13         limits:
14           memory: "1000Gi"
15       command: ["stress"]
16       args: ["--vm", "1", "--vm-bytes", "150M", "--vm-hang", "1"]
17
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
```

- \$ k apply -f day-16/mem-3.yaml
pod/memory-demo-3 created

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
```

- \$ k get pod -n mem-example
- | NAME | READY | STATUS | RESTARTS | AGE |
|---------------|-------|------------------|-------------|-------|
| memory-demo | 1/1 | Running | 0 | 10m |
| memory-demo-2 | 0/1 | CrashLoopBackOff | 5 (29s ago) | 3m43s |
| memory-demo-3 | 0/1 | Pending | 0 | 6s |

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
```

- \$ k describe pod memory-demo-3 -n mem-example
- ```
Name: memory-demo-3
Namespace: mem-example
Priority: 0
Service Account: default
Node: <none>
Labels: <none>
Annotations: <none>
Status: Pending
IP:
IPs: <none>
Containers:
 memory-demo-3-ctr:
 Image: polinux/stress
 Port: <none>
 Host Port: <none>
 Command:
 stress
 memory: 1000Gi
 Environment: <none>
 Mounts:
 /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-lbscp (ro)
Conditions:
 Type Status
 PodScheduled False
Volumes:
 kube-api-access-lbscp:
 Type: Projected (a volume that contains injected data from multiple sources)
 TokenExpirationSeconds: 3607
QoS Class: Burstable
Node-Selectors: <none>
Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
QoS Class: Burstable
Node-Selectors: <none>
Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
 Type Reason Age From Message
 ---- -
Warning FailedScheduling 60s default-scheduler 0/3 nodes are available: 1 Insufficient memory, 1 node(s) had untolerated taint (gpu: true), 1 node(s) had untolerated taint (node-role.kubernetes.io/control-plane:).preemption: 0/3 nodes are available: 1 No preemption victims found for incoming pod, 2 Preemption is not helpful for scheduling.
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
```