

Define Resource Requirements



Elle Krout

Principal Course Author, Pluralsight

Kubernetes Resource Requirements



Add Requests and Limits

```
limited-pod.yaml

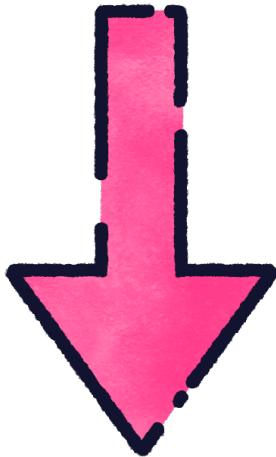
apiVersion: v1
kind: Pod
metadata:
  name: limited-pod
spec:
  containers:
  - name: app
    image: nginx
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
```



How can you best determine sensible resource requirements?

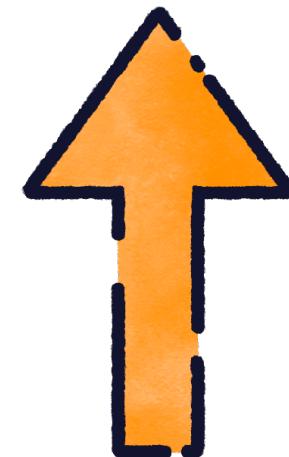


Why Care?



Set Too Low

The pod can be throttled or OOMKilled



Set Too High

You waste cluster resources



Tools For Determining Resource Usage

**Prometheus, Other
Monitoring Tools**

**(Not available on exam
environment)**

Metrics Server

(Available on exam environment)





More Information

**Application Observability and
Maintenance for CKAD**

Elle Krout



**[https://github.com/
kubernetes-sigs/metrics-
server](https://github.com/kubernetes-sigs/metrics-server)**



Setting Resource Requests and Limits



View Pod Data

> `kubectl describe pod <pod_name>`

```
Name:          nginx
Namespace:    default
Priority:     0
Service Account: default
Node:         worker1/172.31.114.162
Start Time:   Fri, 22 Aug 2025 12:03:17 -0400
Labels:        <none>
Annotations:  <none>
Status:       Running
IP:           10.244.1.161
IPs:
  IP: 10.244.1.161
Containers:
  web:
    Container ID:  containerd://a0471ca38205ca5b8e8e78982aa115976336b14ceaf60dd075420d17fe877ef
    Image:         nginx:1.25
    Image ID:     docker.io/library/nginx@sha256:a484819eb60211f5299034ac80f6a681b06f89e65866ce91f356ed7c72af059c
    Port:         <none>
    Host Port:   <none>
    State:       Running
    Started:    Fri, 22 Aug 2025 12:03:18 -0400
    Ready:       True
Events:
  Type  Reason  Age   From            Message
  ----  -----  --   --              --
Normal  Scheduled  106s  default-scheduler  Successfully assigned default/nginx to worker1
```



View Events

> `kubectl get events`

LAST SEEN	TYPE	REASON	OBJECT	MESSAGE
43m	Normal	Starting	node/controller	Starting kubelet.
43m	Warning	InvalidDiskCapacity image filesystem	node/controller	invalid capacity 0 on
43m	Normal	NodeHasSufficientMemory now: NodeHasSufficientMemory	node/controller	Node controller status is
43m	Normal	NodeHasNoDiskPressure now: NodeHasNoDiskPressure	node/controller	Node controller status is
43m	Normal	NodeHasSufficientPID now: NodeHasSufficientPID	node/controller	Node controller status is
43m	Normal	NodeAllocatableEnforced limit across pods	node/controller	Updated Node Allocatable
16m	Normal	Created	pod/load-pod	Created container: stress
16m	Normal	Started	pod/load-pod	Started container stress
12m	Normal	Killing	pod/load-pod	Stopping container stress
11m	Warning	FailedScheduling 2 Insufficient cpu. preemption: 0/2 nodes are available: 2 No preemption victims found for incoming pod	pod/load-pod	0/2 nodes are available: 2 Insufficient cpu. preemption: 0/2 nodes are available: 2 No preemption victims found for incoming pod





What is the pod is running fine?

Use Metrics Server!



Metrics Server



Use for autoscaling

Requests and limits help determine when a pod should scale

Available on the CKAD



Determine Resource Requirements

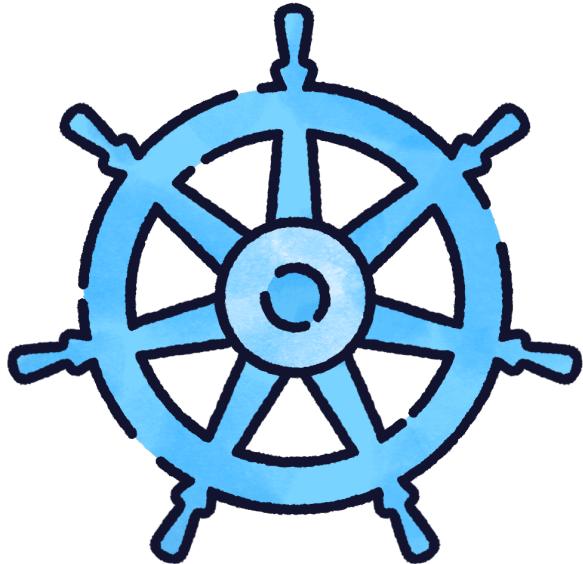
- 1 Run `kubectl top pod <pod_name>`
- 2 Observe point-in-time steady state metrics
- 3 Set requests slightly above observed metrics
- 4 Set limits 1.5 to 2x above observed metrics
- 5 Update the pod and redeploy



Exam Tip!

If the pod is OOMKilled or otherwise throttled, removed preset resource requirements to observe the pod running without restrictions.





During the Exam

When setting requests and limits remember to observe, measure, and adjust.



Exam Scenario



Deploy the `memory-app.yaml` pod. Wait a few minutes, then check its status

Troubleshoot the `memory-app` pod and get it in a running state



Exam Tip!

During the exam, it won't always be obvious it's a resource requirements task, but if it involved an OOMKilled or throttled pod, it's a good place to start.

