Guaranteed
: Pod that has an equal amount of `request` and `limit` resources. These are the highest-priority Pods and guaranteed not to be killed before Best-Effort and Burstable Pods.

So the resource characteristics you define or omit for the containers have a direct impact on its QoS and define the relative importance of the Pod in the event of resource starvation. Define your Pod resource requirements with this consequence in mind.

Pod Priority

We explained how container resource declarations also define Pods' QoS and affect the order in which the Kubelet kills the container in a Pod in case of resource starvation. Another related feature that is still in beta at the time of this writing is Pod Priority and Preemption. Pod priority allows indicating the importance of a Pod relative to other Pods, which affects the order in which Pods are scheduled. Let's see that in action in Example 2-4.

Example 2-4. Pod priority

```
apiVersion: scheduling.k8s.io/v1beta1
kind: PriorityClass
metadata:
  name: high-priority                    ❶
value: 1000                              ❷
globalDefault: false
description: This is a very high priority Pod class
---
apiVersion: v1
kind: Pod
metadata:
  name: random-generator
  labels:
    env: random-generator
spec:
  containers:
  - image: k8spatterns/random-generator:1.0
    name: random-generator
  priorityClassName: high-priority       ❸
```

❶ The name of the priority class object

❷ The priority value of the object

❸ The priority class to use with this Pod, as defined in PriorityClass resource