Pod Quality of Service Classes

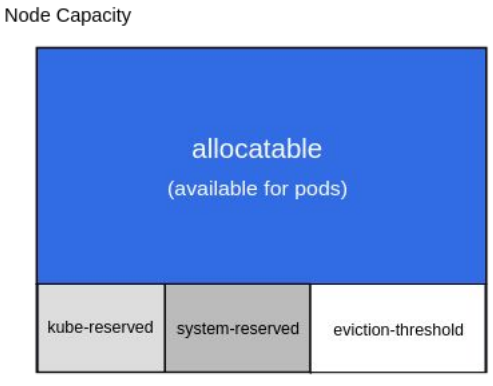
Capacity and Allocatable Resources

The maximum resources available for any container is the maximum resources on a single Kubernetes node.

However, not all Kubernetes node resources is available for the pods.

Part of the node resources are saved for Kubernetes agent essential components, operating system, and eviction threshold.

- **Capacity**: total node resources.
- **Allocatable**: resources available for Pods.



Pod Quality of Service Classes

- Kubernetes classifies the Pods that you run and allocates each Pod into a specific **quality of service (QoS)** class.
- Kubernetes relies on this classification to make decisions about **which Pods to evict** when there are not enough available resources on a Node.
- Kubernetes does this classification **based on the resource requests** of the Containers in that Pod, along with **how those requests relate to resource limits.**

Pod Quality of Service Classes

- The possible QoS classes are **Guaranteed**, **Burstable**, and **BestEffort**.
- When a Node runs out of resources, Kubernetes will **first evict BestEffort Pods** running on that Node, **followed by Burstable and finally Guaranteed Pods**.
- When this eviction is due to resource pressure, only Pods exceeding resource requests are candidates for eviction.

Guaranteed

- Pods that are **Guaranteed** have the **strictest resource limits** and are **least likely to face eviction**.
- They are guaranteed not to be killed **until they exceed their limits** or there are no lower-priority Pods that can be preempted from the Node.

Guaranteed - Criteria

For a Pod to be given a QoS class of Guaranteed:

- Every Container in the Pod must **have a memory limit and a memory request.**
- For every Container in the Pod, **the memory limit must equal the memory request.**
- Every Container in the Pod must **have a CPU limit and a CPU request.**
- For every Container in the Pod, **the CPU limit must equal the CPU request.**

Guaranteed - Criteria

```yaml
apiVersion: v1
kind: Pod
metadata:
 name: qos-demo
spec:
 containers:
 - name: qos-demo-ctr
   image: nginx
   resources:
     limits:
       memory: "200Mi"
       cpu: "700m"
     requests:
       memory: "200Mi"
       cpu: "700m"
```


Burstable

- Pods that are Burstable have some lower-bound resource guarantees **based on the request**, but do not require a specific limit.
- If a limit is not specified, it defaults to a limit equivalent to the capacity of the Node, which allows the Pods to flexibly increase their resources if resources are available.
- In the **event of Pod eviction due to Node resource pressure,** these Pods are evicted o**nly after all BestEffort Pods** are evicted. Because a **Burstable Pod** can include a **Container that has no resource limits or requests**, a Pod that is Burstable can try to use any amount of node resources.

Burstable - Criteria

Criteria

A Pod is given a QoS class of Burstable if:

- The Pod does **not meet the criteria for QoS class Guaranteed**.
- At least **one Container in the Pod** has a **memory or CPU request or limit.**

Burstable - Criteria

```yaml
apiVersion: v1
kind: Pod
metadata:
 name: qos-demo-2
spec:
 containers:
 - name: qos-demo-2-ctr
   image: nginx
   resources:
     limits:
       memory: "200Mi"
     requests:
       memory: "100Mi"
```

Burstable - Criteria

```yaml
apiVersion: v1
kind: Pod
metadata:
 name: qos-demo-4
spec:
 containers:
 - name: qos-demo-4-ctr-1
   image: nginx
   resources:
     requests:
       memory: "200Mi"
 - name: qos-demo-4-ctr-2
   image: redis
```

Notice that this Pod meets the criteria for QoS class Burstable. That is, it does not meet the criteria for QoS class Guaranteed, **and one of its Containers has a memory request.**

BestEffort

- Pods in the **BestEffort** QoS class can use node resources that aren't specifically assigned to Pods in other QoS classes.
- For example, if you have a node with **16 CPU cores available** to the kubelet, and you assign **4 CPU cores to a Guaranteed** Pod, then a Pod in the BestEffort QoS class can try to use any amount of the remaining **12 CPU cores**.
- The kubelet prefers to **evict BestEffort** Pods if the node comes under resource pressure.

BestEffort - Criteria

- A Pod has a QoS class of **BestEffort** if it **doesn't meet the criteria for either Guaranteed or Burstable**.
- In other words, a Pod is BestEffort only if **none of the Containers** in the Pod have **a memory limit or a memory request, and none of the Containers** in the Pod **have a CPU limit or a CPU request.**

BestEffort - Criteria

```
apiVersion: v1
kind: Pod
metadata:
 name: qos-demo-3
spec:
 containers:
 - name: qos-demo-3-ctr
   image: nginx
```

THANKS!

Any questions?