

K8S 19: Kubernetes. Advanced. Quality of Service

Описание:

Для более простого управления ресурсами в kubernetes есть QoS классы.

Изначально есть три класса BestEffort, Burstable, Guaranteed.

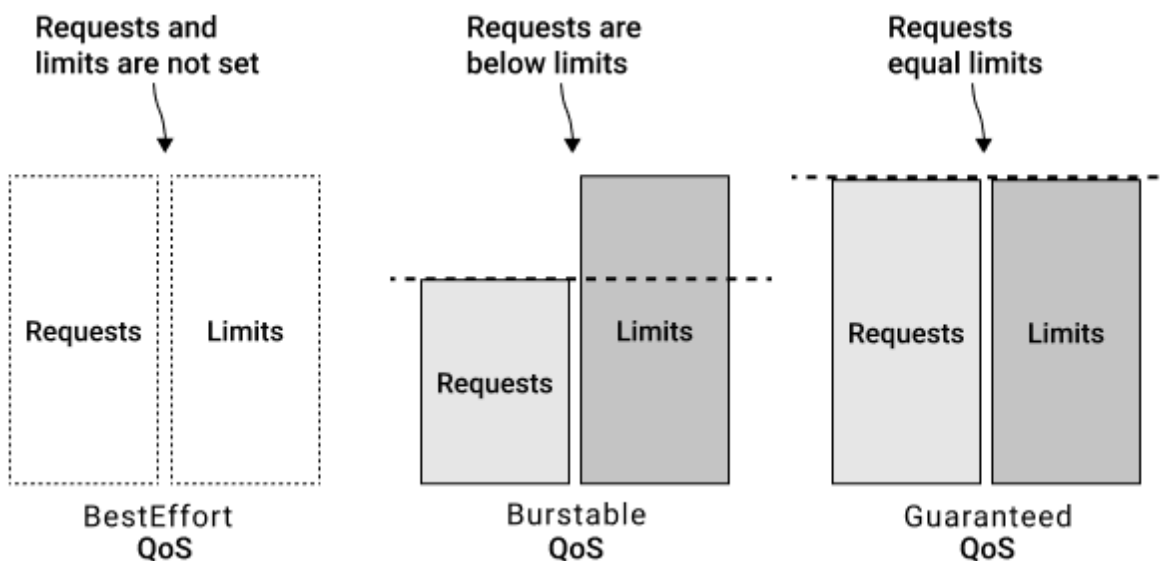
QoS класс не задается нигде в описании объекта, он зависит от комбинации requests и limits.

BestEffort самый низкоприоритетный класс. Он назначается объектам у которых не указано requests и limits. В самом худшем случае такие объекты могут вообще не получить ресурсов. Они же первые кандидаты на перемещение на другой узел при перераспределении ресурсов.

Guaranteed - назначается объектам с заданными requests == limits.

Нужно задать эти значения для каждого контейнера, при этом запрошенное и лимитируемое значение rpi и ram должно быть идентичным.

Burstable - почти как Guaranteed, только requests < limits, или у которых задан один из параметров.

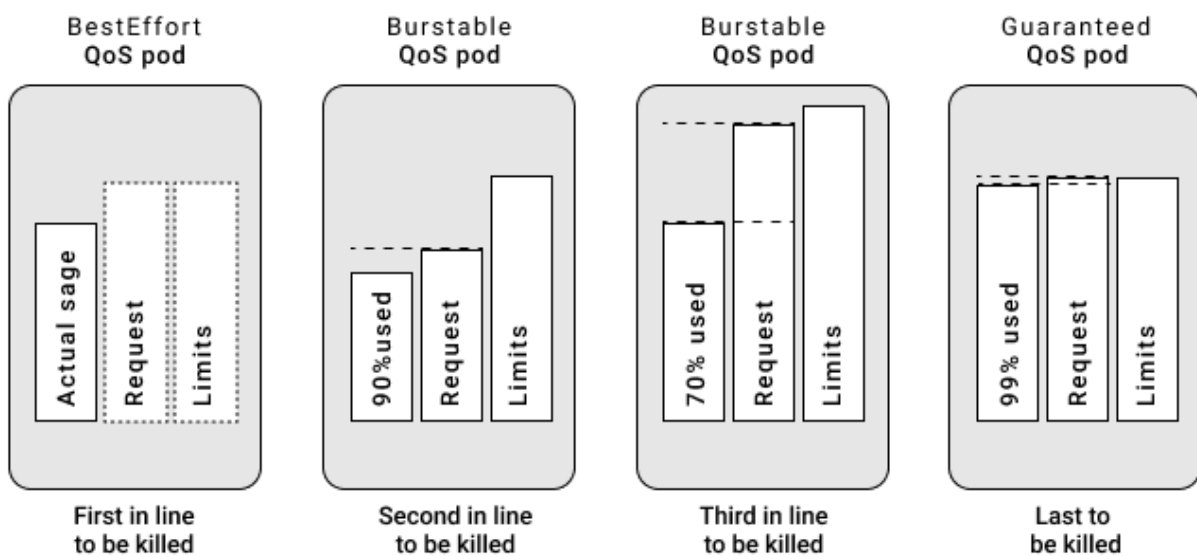


CPU requests vs. limits	Memory requests vs. limits	Container QoS class
None set	None set	BestEffort
None set	Requests < Limits	Burstable
None set	Requests = Limits	Burstable
Requests < Limits	None set	Burstable
Requests < Limits	Requests < Limits	Burstable
Requests < Limits	Requests = Limits	Burstable
Requests = Limits	Requests = Limits	Guaranteed

если в объекте несколько контейнеров

Container 1 QoS class	Container 2 QoS class	Pods QoS class
BestEffort	BestEffort	BestEffort
BestEffort	Burstable	Burstable
BestEffort	Guaranteed	Burstable
Burstable	Burstable	Burstable
Burstable	Guaranteed	Burstable
Guaranteed	Guaranteed	Guaranteed

Рассмотрим ситуацию, когда ресурсов недостаточно. Например недостаток памяти. OOM-Killer первыми завершит процессы BestEffort, затем Burstable, и только потом Guaranteed.



Важно упомянуть, что QoS влияет только на такие ресурсы, как CPU и Memory, но если на сервере не хватает диска для запуска приложения, то несмотря на политики QoS никакой под на указанной ноде не запустится из-за DiskPressure состояния ноды
Информацию о том, какой QoS выставлен для пода, можно узнать, применив `kubectl describe pod ...` на нем

Полезные ссылки:

- [Inside Kubernetes Resource Management \(QoS\) – Mechanics and Lessons from the Field](#)
- [Configure Quality of Service for Pods \(official docs\)](#)
- [Everything you Need to Know about Kubernetes Quality of Service \(QoS\) Classes](#)
- [What are Quality of Service \(QoS\) Classes in Kubernetes](#)

Задание:

1. Напишите 3 deployment, в каждом из которых реализован свой QoS с образом `nginx:stable`:
 - `nginx-be` - с реализованным BestEffort QoS
 - `nginx-burst` - с реализованным Burstable QoS с `requests Memory 100Mi, CPU 100m` и с `limits Memory 200Mi, CPU 200m`
 - `nginx-guar` - с реализованным Guaranteed QoS с `Memory 200Mi, CPU 200m`
2. Выведите список pods с опцией `-o wide` и состояние ресурсов нод при помощи `kubectl top nodes` (команды и вывод сохранить)
3. В ответе пришлите все сохраненные команды и выводы, а так же написанные Deployments манифесты