

Kubernetes Production Architecture & Pod Lifecycle Cheat Sheet

Control plane components, QoS eviction classes, readiness/liveness probes, HPA math, and network policies.

EXECUTIVE SUMMARY

Production SRE guide to Kubernetes scheduling, resource QoS classes, graceful pod termination lifecycle, Horizontal Pod Autoscaler (HPA) equations, and zero-downtime rolling deployments.

KEY EQUATIONS & INVARIANTS

* Horizontal Pod Autoscaler (HPA) Replica Formula:

$$\text{LaTeX: } R_{\text{desired}} = \left\lceil R_{\text{current}} \times \frac{M_{\text{current}}}{M_{\text{target}}} \right\rceil$$

Note: Scales pod replicas proportionally to the ratio of observed metric M_{current} to target M_{target} .

1. Resource Requests, Limits & OOMKill QoS Classes

CPU is a compressible resource (throttled via CFS quota when exceeding `limits.cpu`), whereas Memory is incompressible (exceeding `limits.memory` triggers immediate kernel `OOMKilled` exit code 137).

QoS Class | Configuration Rule | OOM Score Adj | Eviction Priority

Guaranteed | requests == limits for ALL containers (CPU & Mem) | -997 | Evicted LAST (protected)

Burstable | At least one container has requests < limits | 2 to 999 | Evicted second under node pressure

BestEffort | No requests or limits set on any container | 1000 | Evicted FIRST when node memory is low

[YAML] Zero-Downtime Production Deployment Manifest with Probes & PreStop Hook

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: paits-inference-gateway
spec:
  replicas: 4
  strategy:
    type: RollingUpdate
    rollingUpdate: { maxSurge: 1, maxUnavailable: 0 }
  template:
    spec:
      terminationGracePeriodSeconds: 45
      containers:
        - name: api
          resources:
            requests: { cpu: '500m', memory: '1Gi' }
            limits: { memory: '1Gi' }
          readinessProbe:
            httpGet: { path: /health/ready, port: 8000 }
            periodSeconds: 5
          lifecycle:
            preStop:
              exec:
                command: ['/bin/sh', '-c', 'sleep 5']
```

COMMON PRODUCTION & INTERVIEW PITFALLS

- Having a `livenessProbe`` check downstream database dependencies causes cascading pod restarts across the entire cluster during transient DB blips.
- Omitting a `preStop`` sleep hook causes `SIGTERM`` to reach the app before `kube-proxy`` / Ingress controllers finish removing the pod IP from endpoints, resulting in 502 Bad Gateway errors.

FREQUENTLY ASKED QUESTIONS

Q: Why do many SRE teams avoid setting CPU limits while enforcing Memory limits?

A: Strict CPU CFS quotas can cause artificial latency spikes on multi-threaded runtimes even when the host node has idle CPU cores available.