

17-horizontalpodautoscaler

HorizontalPodAutoscaler (HPA) What it is HorizontalPodAutoscaler scales replica count based on metrics (CPU, memory, custom/external metrics). When to use Variable traffic workloads Cost optimization with demand-based scaling Key fields `spec.scaleTargetRef`: target resource (Deployment, StatefulSet) `spec.minReplicas`, `spec.maxReplicas` `spec.metrics[]`: utilization/value targets Common commands `bash` `kubectl get hpa` `kubectl autoscale deployment web --cpu-percent=70 --min=2 --max=10` `kubectl describe hpa web` `kubectl delete hpa web` **YAML example** `yaml` `apiVersion: autoscaling/v2 kind: HorizontalPodAutoscaler metadata: name: web spec: scaleTargetRef: apiVersion: apps/v1 kind: Deployment name: web minReplicas: 2 maxReplicas: 10 metrics: - type: Resource resource: name: cpu target: type: Utilization averageUtilization: 70` **Practical notes** Requires metrics pipeline (typically `metrics-server`) for CPU/memory scaling. Set resource requests; utilization-based scaling depends on them.