Horizontal Pod Autoscaling (HPA) in Kubernetes

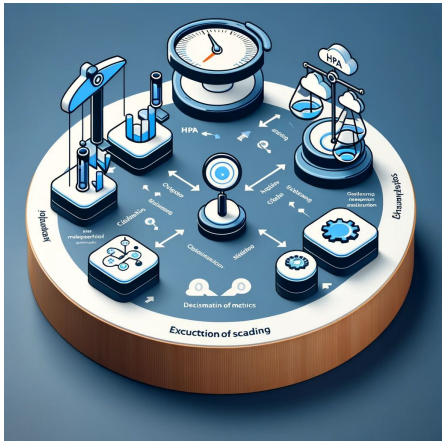
An In-Depth Guide to HPA Parameters

What is Horizontal Pod Autoscaling (HPA)?

- An automated Kubernetes feature for scaling pod replicas
- Adjusts the number based on CPU usage or custom metrics
- Manages load changes dynamically
- Improves resource efficiency and application performance
- Provides elasticity for deployments
- Ensures high availability

How Does HPA Work?

- Metrics Collection: HPA gathers real-time data on resource usage
- Metrics Evaluation: Compares current usage against predefined thresholds
- Decision Making: Determines if scaling out (up) or in (down) is needed
- Scaling Execution: Automatically adjusts the number of pod replicas



```
apiVersion: autoscaling/v2

kind: HorizontalPodAutoscaler

spec:

  behavior:

# behavior configures the scaling behavior of the target in both Up and Down directions: scaleUp and scaleDown fields respectively.
```

```
    scaleDown:
# scaleDown is scaling policy for scaling Down. If not set, the default value is to allow to scale down to minReplicas pods, with a 300 second stabilization window.
      policies:
# policies is a list of potential scaling policies which can be used during scaling. At least one policy must be specified.
```

```
      - periodSeconds: 60
# PeriodSeconds specifies the window of time for which the policy should hold true. PeriodSeconds must be greater than zero and less than or equal to 1800 (30 min).
        type: Percent
# Type is used to specify the scaling policy.
        value: 25
# Value contains the amount of change which is permitted by the policy. It must be greater than zero.
```

```
      selectPolicy: Min
# selectPolicy is used to specify which policy should be used. If not set, the default value Max is used.
      stabilizationWindowSeconds: 300
# StabilizationWindowSeconds is the number of seconds for which past recommendations should be considered while scaling up or scaling down. StabilizationWindowSeconds must be greater than or equal to zero and less than or equal to 3600 (one hour). If not set, use the default values: 300 (i.e. the stabilization window is 300 seconds long).
```

```
    scaleUp:
# scaleUp is scaling policy for scaling Up. If not set, the default value is the higher of: increase no more than 4 pods per 60 seconds, or double the number of pods per 60 seconds. No stabilization is used by default.
      policies:
# policies is a list of potential scaling policies which can be used during scaling. At least one policy must be specified.
```

```
    - periodSeconds: 30
# PeriodSeconds specifies the window of time for which the policy should hold true. PeriodSeconds must be greater than zero and less than or equal to 1800 (30 min).
      type: Percent
# Type is used to specify the scaling policy.
      value: 50
# Value contains the amount of change which is permitted by the policy. It must be greater than zero.
```

```
      selectPolicy: Max
# selectPolicy is used to specify which policy should be used. If not set, the default value Max is used.
      stabilizationWindowSeconds: 0
# StabilizationWindowSeconds is the number of seconds for which past recommendations should be considered while scaling up or scaling down. StabilizationWindowSeconds must be greater than or equal to zero and less than or equal to 3600 (one hour). If not set, use the default values: 0 (i.e. no stabilization is done).
```

```
  maxReplicas: 60

# maxReplicas is the upper limit for the number of replicas to which the autoscaler can scale up. It cannot be less than minReplicas.

  metrics:

# metrics contains the specifications for which to use to calculate the desired replica count (the maximum replica count across all metrics will be used). The desired replica count is calculated multiplying the ratio between the target value and the current value by the current number of pods. If not set, the default metric will be set to 80% average CPU utilization.
```

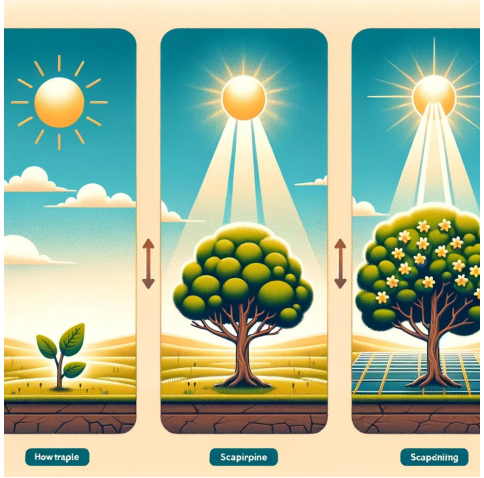
```
  - resource:
# resource refers to a resource metric (such as those specified in requests and limits) known to Kubernetes describing each pod in the current scale target (e.g. CPU or memory). Such metrics are built in to Kubernetes, and have special scaling options on top of those available to normal per-pod metrics using the "pods" source.
      name: cpu
# name is the name of the resource in question.
      target:
# target specifies the target value for the given metric.
```

```
      averageValue: 465m
# averageValue is the target value of the average of the metric
across all relevant pods (as a quantity).
      type: AverageValue
# type represents whether the metric type is Utilization,
Value, or AverageValue.
    type: Resource
# type is the type of metric source. It will be one of
"External", "Object", "Pods" or "Resource", each corresponds to
a matching field in the object.
```

```
  minReplicas: 15

# minReplicas is the lower limit for the number of replicas to which the autoscaler can scale down. It defaults to 1 pod. Scaling is active as long as at least one metric value is available.

  scaleTargetRef:

# scaleTargetRef points to the target resource to scale, and is used to the pods for which metrics should be collected, as well as to actually change the replica count.

    apiVersion: argoproj.io/v1alpha1

# API version of the referent.

    kind: Rollout

# Kind of the referent.
```

HPA in Action

- Low Traffic: HPA maintains minimal pod replicas
- Scaling Trigger: On demand, HPA initiates scaling
- High Traffic: HPA scales up to meet increased load



Best Practices and Tips

- Adjust settings for optimal performance
- Monitor metrics regularly for insights
- Maintain thorough documentation
- Implement security measures to protect your setup

Summary

- HPA automates pod scaling based on observed metrics
- Configure min/max replicas to manage scaling boundaries
- Utilize CPU and custom metrics for precise scaling
- Monitor and adjust HPA settings regularly for efficiency

Q&A

- Ready to answer your queries
- Let's discuss any thoughts or insights you may have

Thank You!

- Please feel free to reach out with any further questions or follow-up