Kubernetes-Cluster Autoscaler

Here's how Cluster Autoscaler works:

1. **Monitoring**: CA continuously monitors the cluster's CPU utilization across all nodes.
2. **Scaling Decision**: When the average CPU utilization of the nodes exceeds a specified threshold (called the "scale-up" threshold), CA adds more nodes to the cluster.
3. **Node Addition**: CA creates new nodes and attaches them to the existing cluster.
4. **Scaling Down**: Conversely, when the average CPU utilization drops below another threshold (the "scale-down" threshold), CA removes excess nodes from the cluster.

Let's consider an example:

Suppose we have a Kubernetes cluster with 3 worker nodes, each with 2 CPUs. Our pod deployment requires resources to run a busy workload, which consumes a significant portion of the available CPU capacity. In this scenario:

1. **CPU Utilization exceeds threshold**: The average CPU utilization across all nodes (60%) exceeds the scale-up threshold (50%).
2. **CA adds new node**: Cluster Autoscaler creates a new node with 2 CPUs and attaches it to the existing cluster.
3. **Pods are redistributed**: CA ensures that the pods running on the original 3 nodes are evenly distributed across the 4 nodes, including the newly added one.
4. **Scaling down happens**: After some time, the workload decreases, and the average CPU utilization drops below the scale-down threshold (30%). In this case:
    - **CA removes excess node**: Cluster Autoscaler deletes the excess node to prevent waste.

Example YAML configuration for Cluster Autoscaler:

```
 apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: hpa-example
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: my-deployment
  minReplicas: 3
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

In this example, the `HorizontalPodAutoscaler` (HPA) configuration specifies a scale-up threshold of 50% and a maximum number of replicas to 10. When the average CPU utilization across all nodes exceeds 50%, CA will add more nodes until it reaches the maximum limit. Conversely, when the utilization drops below 30%, CA will start removing excess nodes.

Keep in mind: This is just an example configuration; you may need to adjust it based on your specific requirements and cluster architecture.

Curated by Brajesh Kumar