Kubernetes Cluster Autoscaling

Thomas Hartland

How many nodes should my cluster have?

"It should have just the right number of nodes, not too many and not too few."

The “right” size for you cluster

- Depends on a lot of factors
- Might change over time

Cluster autoscaling

The cluster autoscaler is an in-cluster component that monitors the pods and nodes in the cluster.

Cluster autoscaling

- Pending pods? Scale up.

Cluster autoscaling

- Pending pods? Scale up.
- Empty nodes? Scale down.

Horizontal pod autoscaling

- HPA is a core feature of Kubernetes.
- It scales a deployment to satisfy a metric.
- (e.g maintain 80% CPU usage in all pods).

Deploying on OpenStack

Enabled in labels passed to `cluster create`[1]:

```
$ openstack coe cluster create test-cluster
    --cluster-template kubernetes-1.15.3-3
    --labels ...
    --labels auto_scaling_enabled=true
    --labels min_node_count=1
    --labels max_node_count=4
```

[1] https://clouddocs.web.cern.ch/containers/tutorials/cluster-autoscaler.html

Demo

- Let's deploy a cluster with autoscaling
- and trigger a scale up.

While we wait...

- How can we get this pod running faster?
- We want to have the right number of nodes *plus one*.

The solution

- Create a “buffer” deployment that reserves some space.
- By giving that deployment a lower priority than assigned by default, other pods will pre-empt pods in the buffer.
- The buffer pods will then trigger a scale up.

Filling the empty space in the cluster without scaling up

- The cluster autoscaler will ignore pods below a certain priority level.
- By default this priority value is -10.
- A low priority deployment can be used to backfill empty spaces in the cluster without causing a scale up.

Kubernetes scheduling

- By default Kubernetes prefers to schedule pods onto the least used node.
- For efficient scaling down, we want the opposite of that.
- Documented in our cloud docs[2].

[2] https://clouddocs.web.cern.ch/containers/tutorials/scheduling.html

Advising the autoscaler

- Ensure a pod can not be moved by the autoscaler

```
metadata:
  annotations:
    cluster-autoscaler.kubernetes.io/safe-to-evict: false
```

- Ensure a node will not be removed

```
$ kubectl annotate node <nodename>
          cluster-autoscaler.kubernetes.io/scale-down-disabled=true
```

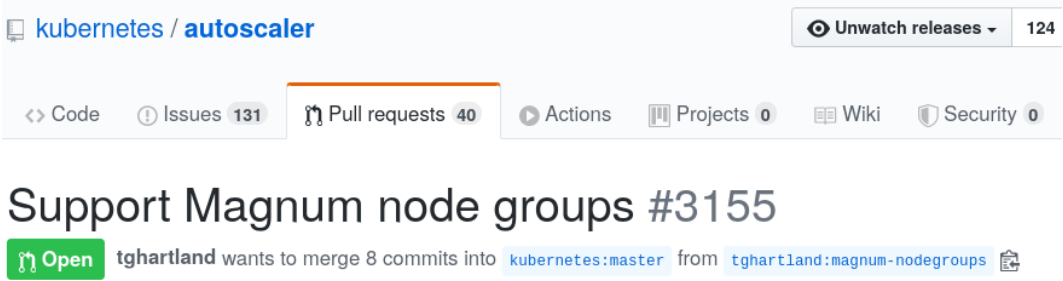
Another use: auto healing

- You have a 5 node cluster
- You enable cluster autoscaling with min=5, max=6.
- If a node breaks the autoscaler will add a new node and remove the broken one.

The future

- Support for Magnum node groups
- Node group auto discovery

The future


kubernetes / autoscaler
Unwatch releases
124
Code
Issues 131
Pull requests 40
Actions
Projects 0
Wiki
Security 0
Support Magnum node groups #3155
Open
tghartland wants to merge 8 commits into kubernetes:master from tghartland:magnum-nodegroups

For further information

https://github.com/kubernetes/autoscaler/blob/master/cluster-autoscaler/FAQ.md

Q&A

Thanks for listening.