Kubernetes Node Autoscaling Operations Guide

Document Level Classification

300

Intended Audience: InfraOps and Substrate developers who have had training on ArgoCD deployments and configurations.

What is Kubernetes Node Autoscaling?

Kubernetes Node Autoscaling is the ability for Substrate clusters to add and remove nodes automatically in response to pods being created and removed. Node autoscaling is a different topic than *pod* autoscaling, but they are closely related because they both need to exist for a fully elastic cluster.

Substrate accomplishes node scale out using Karpenter and performs node removal (compaction) with Compactor (a tool created at Epic). You can watch a demonstration of Karpenter+Compactor here.

Enabling Node Autoscaling

All clusters are enabled for Karpenter and Compactor by default, details

Enabling node autoscaling is relatively simple. You will have to submit a pull request to the ArgoCD Git Repository that enables both Karpenter and Compactor with default configurations, then use the ArgoCD UI to sync the Kubernetes specs into your cluster. Here is a step by step of how to do that:

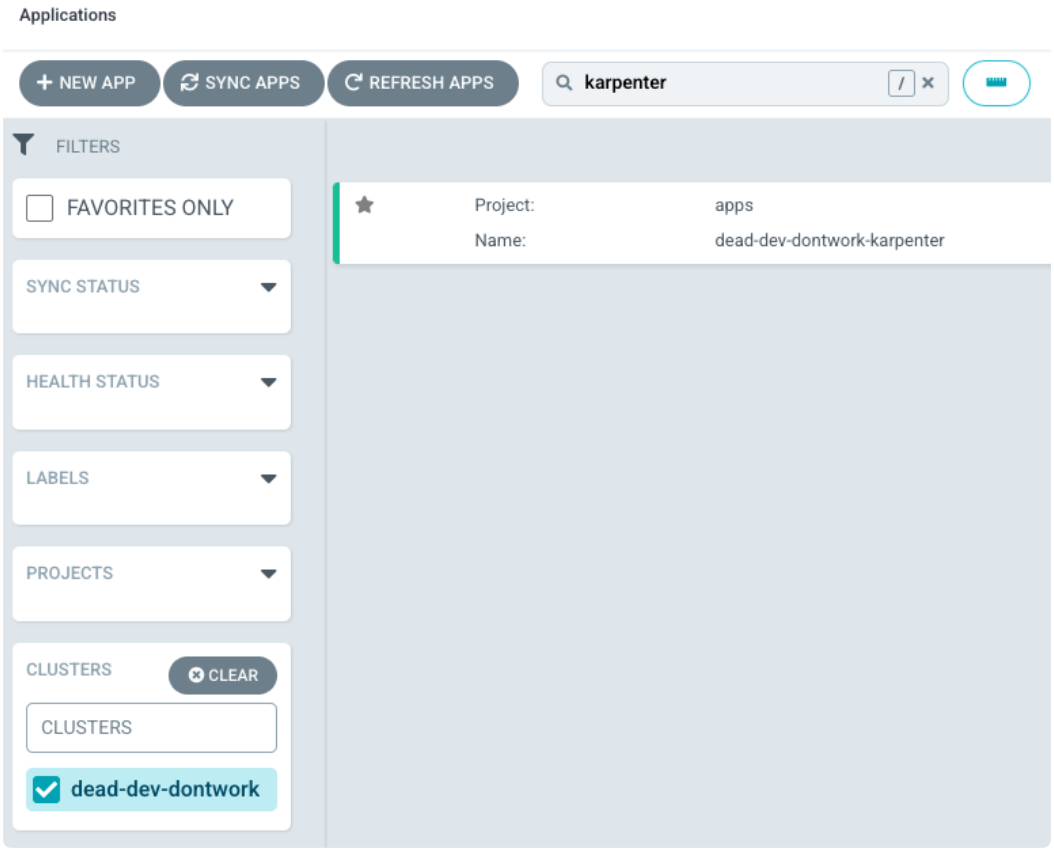
- Find your Substrate cluster in this list.
- Create a pull request to add the following yaml to the file named after your cluster:

karpenter: enabled: true compactor: enabled: true

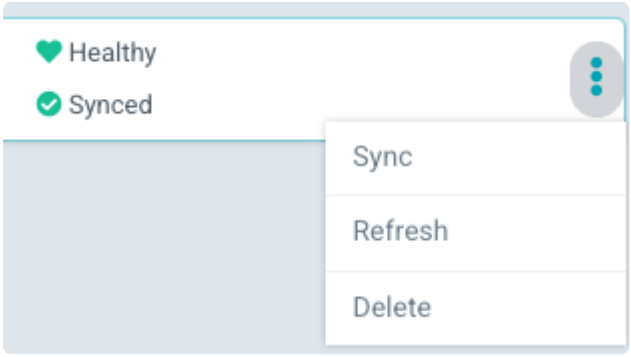
If you are not part of InfraOps, you will need to contact someone to help you with the following steps. You can not yet perform the following steps as a substrate customer. Reach out to #cloud-ops-support-ext for help reviewing and deploying your change. If you need direct attention, please reach out to Eric Greeror Former user (Deleted)

- Once merged, you need to push the specs out via the ArgoCD UI:
  - Find your cluster on the ArgoCD Applications page by filtering for your cluster's name and searching '`karpenter' and then`

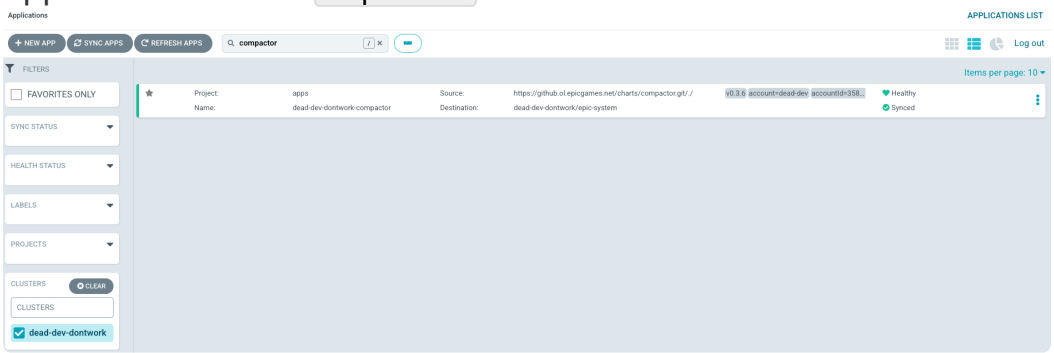
again with 'compactor':



- sync your application until it shows as 'Healthy' and 'Synced':



- Repeat the ArgoCD search and sync process using the application name 'c`ompactor` '



- You now have enabled Karpenter+Compactor in your cluster!

Additional Karpenter and Compactor Configuration

Karpenter is responsible for building out new nodes in your cluster. By default, the configuration we supply will work nicely in Substrate clusters. However, if you find a reason for more advanced configurations, Karpenter can be configured by adding Helm configuration values to your cluster's ArgoCD configuration values. These values in ArgoCD get passed to the various Charts that get applied to your cluster.

All charts must be mirrored to internal repositories before they can be used by ArgoCD, which means **only options in our mirrored charts are available**. You can view the Karpenter and Compactor charts that are currently mirrored under the `charts/` organization on GitHub Enterprise.

Here is an example set of cluster values that configures several options for both Karpenter and Compactor.

```
karpenter:
  enabled: true
  values:
    awsNodeTemplates:
      default:
        instanceProfile: dead-dev-dontwork-worker-profile
    provisioners:
      default:
        requirements:
          instanceType:
            values:
              - c5d.4xlarge
              - c5d.9xlarge
compactor:
  enabled: true
  values:
    compactor:
      debug: true
```

You will notice that because this `values.yaml` file configures multiple charts, the top level of the YAML is the name of the chart followed by some ArgoCD metadata such as `enabled: true` . However, every value under the `values:` section will be passed directly to the chart for that application. If you find a value you want to configure in the mirrored Helm chart for Karpenter or Compactor, you can add it here under the `values:` section as if it were the `values.yaml` file of the nested chart.

Frequently Asked Questions

Placeholder

Page Information:
Page ID: 81068321
Space: Cloud Developer Platform
Downloaded: 2025-07-12 04:09:07