Garbage Collection on EKS

About:
Setting up automatic image cache cleanup on Amazon EKS worker nodes based on disk usage.

We are using the kubelet garbage collection using the arguments to clean up the image cache.

- The --image-gc-high-threshold argument defines the percent of disk usage that initiates image garbage collection. The default is 85%.
- The --image-gc-low-threshold argument defines the percent of disk usage that image garbage collection tries to free. The default is 80%.

NOTE: Please be careful while doing changes in kubelet configuration any error can cause issue in node initialization and fail to join the cluster

Steps:

Adding the kubelet garbage collection arguments to an existing worker node

sudo vi /etc/kubernetes/kubelet/kubelet-config.json

```
{
  "kind": "KubeletConfiguration",
  "apiVersion": "kubelet.config.k8s.io/v1beta1",
  .
  .
  .
  "imageGCHighThresholdPercent": 70,         ==> Add the argument
under the same alignment as the "kind"
  "imageGCLowThresholdPercent": 50,
  "maxPods": ...
}
```

To restart the kubelet service in the worker node, run the following command:

sudo service kubelet restart

Adding the kubelet garbage collection arguments to the node group

- Go to the NodeGroup, and
- click on the ASG associated with it and
- Go to launch template
- Edit the template and create a new version.

In the new version under the advanced settings, click on userdata.
And update the following line:

/etc/eks/bootstrap.sh Fansportiz --kubelet-extra-args '--imageGCHighThresholdPercent=50 --imageGCLowThresholdPercent=30 --node-labels=eks.amazonaws.com/nodegroup-image=ami-0982cb2a877f43c01 ……'

In the above line, we have add the –imageGCHighThresholdPercent and imageGCLowThresholdPercent values to the kubelet args.

Once updated, create the new version and set it to the default. In the ASG also update the template to the default.

Then start instance refresh to update the nodes with the new changes.

Adding the kubelet garbage collection arguments to the Karpenter provisioned nodes:

For Karpenter provisioned node, we have to update the AWSNodeTemplate,

Under the Userdata section in Karpenter AWSNodeTemplate

```
apiVersion: karpenter.k8s.aws/v1alpha1
kind: AWSNodeTemplate
metadata:
  name: al2-example
spec:
  amiFamily: AL2
  instanceProfile: MyInstanceProfile
  subnetSelector:
    karpenter.sh/discovery: my-cluster
  securityGroupSelector:
    karpenter.sh/discovery: my-cluster
  userData: |
    #!/bin/bash
/etc/eks/bootstrap.sh --kubelet-extra-args
'--imageGCHighThresholdPercent=50 --imageGCLowThresholdPercent=30'
```

Update the Userdata script as above,

And apply the updated AWSNodeTemplate.

That’s all,we did it.

Reference: https://repost.aws/knowledge-center/eks-worker-nodes-image-cache