Stackdriver Logging Agent

==========

Stackdriver Logging Agent is a DaemonSet which spawns a pod on each node that reads logs, generated by kubelet, container runtime and containers and sends them to the Stackdriver. When logs are exported to the Stackdriver, they can be searched, viewed, and analyzed.

Learn more at: https://kubernetes.io/docs/tasks/debug-application-cluster/logging-stackdriver

Troubleshooting

In Kubernetes clusters in version 1.10.0 or later, fluentd-gcp DaemonSet can be manually scaled. This is useful e.g. when applications running in the cluster are sending a large volume of logs (i.e. over 100kB/s), causing fluentd-gcp to fail with OutOfMemory errors. Conversely, if the applications aren't generating a lot of logs, it may be useful to reduce the amount of resources consumed by fluentd-gcp, making these resources available to other applications. To learn more about Kubernetes resource requests and limits, see the official documentation (CPU, memory). The amount of resources requested by fluentd-gcp on every node in the cluster can be fetched by running following command:

```
$ kubectl get ds -n kube-system -l k8s-app=fluentd-gcp \
-o custom-columns=NAME:.metadata.name,\
CPU_REQUEST:.spec.template.spec.containers[].resources.requests.cpu,\
MEMORY_REQUEST:.spec.template.spec.containers[].resources.requests.memory,\
MEMORY_LIMIT:.spec.template.spec.containers[].resources.limits.memory
```

This will display an output similar to the following:

```
NAME                  CPU_REQUEST   MEMORY_REQUEST   MEMORY_LIMIT
fluentd-gcp-v2.0.15   100m          200Mi            300Mi
```

In order to change those values, a ScalingPolicy needs to be defined. Currently, only base values are supported (no automatic scaling). The ScalingPolicy can be created using kubectl. E.g. to set cpu request to 101m, memory request to 150Mi and memory limit to 400Mi:

```
$ cat <<EOF | kubectl apply -f -
apiVersion: scalingpolicy.kope.io/v1alpha1
kind: ScalingPolicy
metadata:
  name: fluentd-gcp-scaling-policy
  namespace: kube-system
spec:
  containers:
  - name: fluentd-gcp
    resources:
      requests:
      - resource: cpu
        base: 101m
      - resource: memory
        base: 150Mi
      limits:
      - resource: memory
```

```
        base: 400Mi
EOF
```

To remove the override and go back to GKE-provided defaults, it is enough to just remove the ScalingPolicy:

```
$ kubectl delete -n kube-system scalingpolicies.scalingpolicy.kope.io/fluentd-gcp-scaling-policy
```