Our team has a pod that generates some log output. However, they want to consume the data using an external application, which requires the data to be in a specific format. Our task is to create a pod design that utilizes an adapter running fluentd to format the output from the main container.

* There is a fluentd configuration located on the server at /usr/ckad/fluent.conf. Load the data from this file into a ConfigMap called fluentd-config.
* Create the pod descriptor in /usr/ckad/adapter-pod.yml. An empty file has already been created for us.
* The pod should be named counter.
* Add a container to the pod that runs the busybox image, and name it count.
* Run the count container with the following arguments:
* - /bin/sh
* - -c
* - >
* i=0;
* while true;
* do
* echo "$i: $(date)" >> /var/log/1.log;
* echo "$(date) INFO $i" >> /var/log/2.log;
* i=$((i+1));
* sleep 1;

done

* Add another container called adapter to the pod, and make it run the k8s.gcr.io/fluentd-gcp:1.30 image.
* Add an environment variable to the adapter container called FLUENTD\_ARGS with the value -c /fluentd/etc/fluent.conf.
* Mount the fluentd-config ConfigMap to the adapter container so that the config data is located inside the container in a file at /fluentd/etc/fluent.conf.
* Create a volume for the pod in such a way that the storage will be deleted if the pod is removed from a node. Mount this volume to *both* containers at /var/log. The count container will output log data to this volume, and the adapter container will read the data from the same volume.
* Create a hostPath volume where the adapter container will output the formatted log data. Store the data at the /usr/ckad/log\_output path. Mount the volume to the adapter container at /var/logout.

## Solution

Open a terminal session and log in to Kube Master server via SSH using the credentials listed on the lab page:

ssh cloud\_user@<PUBLIC\_IP>

**Hint:** When copying and pasting code into Vim from the lab guide, first enter :set paste (and then i to enter insert mode) to avoid adding unnecessary spaces and hashes.

### Create a ConfigMap to Store the fluentd Configuration

1. Become root:

sudo -i

1. View the contents of the config file:

cat /usr/ckad/fluent.conf

1. Create a descriptor for the ConfigMap:

vim fluentd-config.yml

1. Enter the following, making sure the indenting is correct (remember the hint at the beginning of the lab guide):
2. apiVersion: v1
3. kind: ConfigMap
4. metadata:
5. name: fluentd-config
6. data:
7. fluent.conf: |
8. <source>
9. type tail
10. format none
11. path /var/log/1.log
12. pos\_file /var/log/1.log.pos
13. tag count.format1
14. </source>
15. <source>
16. type tail
17. format none
18. path /var/log/2.log
19. pos\_file /var/log/2.log.pos
20. tag count.format2
21. </source>
22. <match \*\*>
23. @type file
24. path /var/logout/count
25. time\_slice\_format %Y%m%d%H%M%S
26. flush\_interval 5s
27. log\_level trace

</match>

1. Save and exit the file by pressing **Escape** followed by :wq!.
2. Create the ConfigMap in the cluster:

kubectl apply -f fluentd-config.yml

### Create the Pod Descriptor

1. Open the /usr/ckad/adapter-pod.yml descriptor file:

vim /usr/ckad/adapter-pod.yml

1. Enter the following, making sure the indenting is correct:
2. apiVersion: v1
3. kind: Pod
4. metadata:
5. name: counter
6. spec:
7. containers:
8. - name: count
9. image: busybox
10. args:
11. - /bin/sh
12. - -c
13. - >
14. i=0;
15. while true;
16. do
17. echo "$i: $(date)" >> /var/log/1.log;
18. echo "$(date) INFO $i" >> /var/log/2.log;
19. i=$((i+1));
20. sleep 1;
21. done
22. volumeMounts:
23. - name: varlog
24. mountPath: /var/log
25. - name: adapter
26. image: k8s.gcr.io/fluentd-gcp:1.30
27. env:
28. - name: FLUENTD\_ARGS
29. value: -c /fluentd/etc/fluent.conf
30. volumeMounts:
31. - name: varlog
32. mountPath: /var/log
33. - name: config-volume
34. mountPath: /fluentd/etc
35. - name: logout
36. mountPath: /var/logout
37. volumes:
38. - name: varlog
39. emptyDir: {}
40. - name: config-volume
41. configMap:
42. name: fluentd-config
43. - name: logout
44. hostPath:

path: /usr/ckad/log\_output

### Create the Pod in the Cluster and Make Sure It Is Working

1. Create the pod:

kubectl apply -f /usr/ckad/adapter-pod.yml

1. Verify the pod starts up:

kubectl get pod counter

1. Verify everything is working by checking the output directory on the worker node. Log in to the worker from the master:

ssh cloud\_user@10.0.1.102

The password is the same as the one we used to log in to the Kube Master server.

1. Make sure we see fluentd output files in the output directory:

ls /usr/ckad/log\_output