Lesson 09 Demo 02

Understanding the Kubernetes Cluster Logging Architecture

Objective: To monitor, troubleshoot, and manage applications and services within the Kubernetes clusters

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes cluster should already be set up (refer to the steps provided in Lesson 02, Demo 01 for guidance).

Steps to be followed:

1. Get help with logging
2. Understand log options and switch information

Step 1: Get help with logging

1.1 To invoke the help menu, execute the following command:

kubectl logs --help

```
labsuser@master:~$ kubectl logs --help
Print the logs for a container in a pod or specified resource. If the pod has only one container, the container name is
optional.

Examples:
  # Return snapshot logs from pod nginx with only one container
  kubectl logs nginx

  # Return snapshot logs from pod nginx with multi containers
  kubectl logs nginx --all-containers=true

  # Return snapshot logs from all containers in pods defined by label app=nginx
  kubectl logs -l app=nginx --all-containers=true

  # Return snapshot of previous terminated ruby container logs from pod web-1
  kubectl logs -p -c ruby web-1

  # Begin streaming the logs of the ruby container in pod web-1
  kubectl logs -f -c ruby web-1

  # Begin streaming the logs from all containers in pods defined by label app=nginx
  kubectl logs -f -l app=nginx --all-containers=true

  # Display only the most recent 20 lines of output in pod nginx
  kubectl logs --tail=20 nginx
```

```
    --since-time='':
        Only return logs after a specific date (RFC3339). Defaults to all logs. Only one of since-time / since may be
        used.

    --tail=-1:
        Lines of recent log file to display. Defaults to -1 with no selector, showing all log lines otherwise 10, if a
        selector is provided.

    --timestamps=false:
        Include timestamps on each line in the log output

Usage:
  kubectl logs [-f] [-p] (POD | TYPE/NAME) [-c CONTAINER] [options]

Use "kubectl options" for a list of global command-line options (applies to all commands).
labsuser@master:~$
```

1.2 Create a pod named **busybox** using the following command:

vi busybox.yaml

```
labsuser@master:~$ vi busybox.yaml
```

1.3 Now paste the following code in **busybox.yaml** file:

```
apiVersion: v1
kind: Pod
metadata:
  name: counter
spec:
  containers:
  - name: count
    image: busybox:1.28
    args: [/bin/sh, -c,
          'i=0; while true; do echo "$i: $(date)"; i=$((i+1)); sleep 1; done']
```

```
apiVersion: v1
kind: Pod
metadata:
  name: counter
spec:
  containers:
  - name: count
    image: busybox:1.28
    args: [/bin/sh, -c,
            'i=0; while true; do echo "$i: $(date)"; i=$((i+1)); sleep 1; done']
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

1.4 Deploy the YAML file and check its logs using the following commands:
kubectl apply -f busybox.yaml
kubectl logs counter

```
labsuser@master:~$ vi busybox.yaml
labsuser@master:~$ kubectl apply -f busybox.yaml
pod/counter created
labsuser@master:~$ kubectl logs counter
0: Wed Oct 11 07:03:44 UTC 2023
1: Wed Oct 11 07:03:45 UTC 2023
2: Wed Oct 11 07:03:46 UTC 2023
3: Wed Oct 11 07:03:47 UTC 2023
labsuser@master:~$
```

1.5 To retrieve the last five lines of the log, execute the following command:
kubectl logs counter --tail=5

```
labsuser@master:~$ vi busybox.yaml
labsuser@master:~$ kubectl apply -f busybox.yaml
pod/counter created
labsuser@master:~$ kubectl logs counter
0: Wed Oct 11 07:03:44 UTC 2023
1: Wed Oct 11 07:03:45 UTC 2023
2: Wed Oct 11 07:03:46 UTC 2023
3: Wed Oct 11 07:03:47 UTC 2023
labsuser@master:~$ kubectl logs counter --tail=5
59: Wed Oct 11 07:04:43 UTC 2023
60: Wed Oct 11 07:04:44 UTC 2023
61: Wed Oct 11 07:04:45 UTC 2023
62: Wed Oct 11 07:04:46 UTC 2023
63: Wed Oct 11 07:04:47 UTC 2023
labsuser@master:~$
```

Step 2: Understand log options and switch information

2.1 To obtain logs from all containers in a namespace, execute the following command:

kubectl logs counter --all-containers

```
labsuser@master:~$ kubectl logs counter --all-containers
0: Wed Oct 11 07:03:44 UTC 2023
1: Wed Oct 11 07:03:45 UTC 2023
2: Wed Oct 11 07:03:46 UTC 2023
3: Wed Oct 11 07:03:47 UTC 2023
4: Wed Oct 11 07:03:48 UTC 2023
5: Wed Oct 11 07:03:49 UTC 2023
6: Wed Oct 11 07:03:50 UTC 2023
7: Wed Oct 11 07:03:51 UTC 2023
8: Wed Oct 11 07:03:52 UTC 2023
9: Wed Oct 11 07:03:53 UTC 2023
10: Wed Oct 11 07:03:54 UTC 2023
11: Wed Oct 11 07:03:55 UTC 2023
12: Wed Oct 11 07:03:56 UTC 2023
13: Wed Oct 11 07:03:57 UTC 2023
14: Wed Oct 11 07:03:58 UTC 2023
15: Wed Oct 11 07:03:59 UTC 2023
16: Wed Oct 11 07:04:00 UTC 2023
17: Wed Oct 11 07:04:01 UTC 2023
18: Wed Oct 11 07:04:02 UTC 2023
19: Wed Oct 11 07:04:03 UTC 2023
20: Wed Oct 11 07:04:04 UTC 2023
21: Wed Oct 11 07:04:05 UTC 2023
22: Wed Oct 11 07:04:06 UTC 2023
```

2.2 To get logs from a specific time range, use the following format:

kubectl logs counter --since=<timespan>

For example, to get logs from the past hour:

kubectl logs counter --since=1h

```
labsuser@master:~$ kubectl logs counter --since=1h
0: Wed Oct 11 07:03:44 UTC 2023
1: Wed Oct 11 07:03:45 UTC 2023
2: Wed Oct 11 07:03:46 UTC 2023
3: Wed Oct 11 07:03:47 UTC 2023
4: Wed Oct 11 07:03:48 UTC 2023
5: Wed Oct 11 07:03:49 UTC 2023
6: Wed Oct 11 07:03:50 UTC 2023
7: Wed Oct 11 07:03:51 UTC 2023
8: Wed Oct 11 07:03:52 UTC 2023
9: Wed Oct 11 07:03:53 UTC 2023
10: Wed Oct 11 07:03:54 UTC 2023
11: Wed Oct 11 07:03:55 UTC 2023
12: Wed Oct 11 07:03:56 UTC 2023
13: Wed Oct 11 07:03:57 UTC 2023
14: Wed Oct 11 07:03:58 UTC 2023
15: Wed Oct 11 07:03:59 UTC 2023
16: Wed Oct 11 07:04:00 UTC 2023
17: Wed Oct 11 07:04:01 UTC 2023
18: Wed Oct 11 07:04:02 UTC 2023
19: Wed Oct 11 07:04:03 UTC 2023
20: Wed Oct 11 07:04:04 UTC 2023
21: Wed Oct 11 07:04:05 UTC 2023
22: Wed Oct 11 07:04:06 UTC 2023
23: Wed Oct 11 07:04:07 UTC 2023
24: Wed Oct 11 07:04:08 UTC 2023
```

By following these steps, you have successfully obtained hands-on experience with Kubernetes logging, gaining a deeper understanding of the cluster's logging architecture.