

Lesson 09 Demo 08

Handling Component Failure Threshold

Objective: To view the nodes within a cluster and gather detailed health information for ensuring a proper functioning of the nodes

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. Check the cluster health information

Step 1: Check the cluster health information

- 1.1 Execute the following command to check the nodes in the cluster:

kubectl get nodes

```
labsuser@master:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
master.example.com                 Ready    control-plane   49m   v1.28.2
worker-node-1.example.com         Ready    <none>        47m   v1.28.2
worker-node-2.example.com         Ready    <none>        47m   v1.28.2
labsuser@master:~$
```

1.2 To check the health information of a cluster and verify its content, execute the following commands:

```
kubectl cluster-info dump > dump.json
```

```
vi dump.json
```

```
labsuser@master:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
master.example.com                  Ready    control-plane   4h42m   v1.28.2
worker-node-1.example.com           Ready    <none>         4h41m   v1.28.2
worker-node-2.example.com           Ready    <none>         4h41m   v1.28.2

labsuser@master:~$ kubectl cluster-info dump > dump.json
labsuser@master:~$ vi dump.json
labsuser@master:~$
```

```
"kind": "NodeList",
"apiVersion": "v1",
"metadata": {
  "resourceVersion": "23821"
},
"items": [
  {
    "metadata": {
      "name": "master.example.com",
      "uid": "cf00285b-4007-49e0-ac04-b6b2b7f758d9",
      "resourceVersion": "23512",
      "creationTimestamp": "2023-10-17T10:46:38Z",
      "labels": {
        "beta.kubernetes.io/arch": "amd64",
        "beta.kubernetes.io/os": "linux",
        "kubernetes.io/arch": "amd64",
        "kubernetes.io/hostname": "master.example.com",
        "kubernetes.io/os": "linux",
        "node-role.kubernetes.io/control-plane": "",
        "node.kubernetes.io/exclude-from-external-load-balancers": ""
      },
      "annotations": {
        "kubeadm.alpha.kubernetes.io/cri-socket": "unix:///var/run/containerd/containerd.sock",
        "node.alpha.kubernetes.io/ttl": "0",
        "projectcalico.org/IPv4Address": "172.31.35.149/20",
        "projectcalico.org/IPv4IPTunnelAddr": "172.16.204.64",
        "volumes.kubernetes.io/controller-managed-attach-detach": "true"
      }
    },
    "spec": {
      "bootstrap": {
        "token": "abcdefg",
        "tokenTTL": "0"
      },
      "controlPlane": {
        "extraManifests": {
          "kube-apiserver": {
            "image": "k8s.gcr.io/kube-apiserver:v1.28.2",
            "args": [
              "--etcd-servers=https://127.0.0.1:2379",
              "--kubeletconfig=https://127.0.0.1:10253/config.yaml",
              "--enable-bootstrap"
            ],
            "extraArgs": {
              "v": "6"
            }
          },
          "kube-controller-manager": {
            "image": "k8s.gcr.io/kube-controller-manager:v1.28.2",
            "args": [
              "--cluster-cidr=10.240.0.0/16",
              "--node-cidr-mask-size=24"
            ],
            "extraArgs": {
              "v": "6"
            }
          },
          "kube-scheduler": {
            "image": "k8s.gcr.io/kube-scheduler:v1.28.2",
            "args": [
              "--kubeletconfig=https://127.0.0.1:10253/config.yaml"
            ],
            "extraArgs": {
              "v": "6"
            }
          }
        },
        "extraVolumes": [
          {
            "name": "certs",
            "mountPath": "/etc/kubernetes/pki",
            "storageClassName": "default",
            "volumeMode": "Filesystem"
          }
        ],
        "extraVolumesMount": [
          {
            "name": "certs",
            "mountPath": "/etc/kubernetes/pki"
          }
        ],
        "extraManifestsMount": {
          "kube-apiserver": {
            "mountPath": "/var/lib/kubelet/pki"
          },
          "kube-controller-manager": {
            "mountPath": "/var/lib/kubelet/pki"
          },
          "kube-scheduler": {
            "mountPath": "/var/lib/kubelet/pki"
          }
        },
        "extraManifestsMountSubPath": "certs"
      },
      "etcd": {
        "image": "k8s.gcr.io/etcd:v3.5.9",
        "args": [
          "--advertise-client-urls=https://127.0.0.1:2379",
          "--listen-client-urls=https://127.0.0.1:2379",
          "--listen-metrics-urls=https://127.0.0.1:2381",
          "--metrics-urls=https://127.0.0.1:2381"
        ],
        "extraArgs": {
          "v": "6"
        },
        "extraVolumes": [
          {
            "name": "etcd-data",
            "mountPath": "/var/lib/etcd",
            "storageClassName": "default",
            "volumeMode": "Filesystem"
          }
        ],
        "extraVolumesMount": [
          {
            "name": "etcd-data",
            "mountPath": "/var/lib/etcd"
          }
        ],
        "extraManifestsMount": {
          "etcd": {
            "mountPath": "/var/lib/etcd"
          }
        },
        "extraManifestsMountSubPath": "etcd"
      },
      "kubelet": {
        "image": "k8s.gcr.io/kubelet:v1.28.2",
        "args": [
          "--admission-control=NodeRestriction",
          "--cni=calico",
          "--cni-conf-dir=/etc/cni/net.d",
          "--cni-plugin=/usr/local/bin/cni-plugin",
          "--config=/etc/kubernetes/kubelet.conf",
          "--container-runtime=containerd",
          "--kubeconfig=/etc/kubernetes/kubelet.conf",
          "--node-labels",
          "--pod-infra-container-image=k8s.gcr.io/pause:3.9",
          "--resolv-conf=/etc/resolv.conf",
          "--root-dir=/var/lib/kubelet",
          "--serialize-features=true",
          "--static-dir=/var/lib/kubelet/manifests",
          "--tls-cert-file=/etc/kubernetes/pki/kubelet.pem",
          "--tls-private-key-file=/etc/kubernetes/pki/kubelet-key.pem",
          "--volume-plugin-dir=/var/lib/kubelet/plugins"
        ],
        "extraArgs": {
          "v": "6"
        },
        "extraVolumes": [
          {
            "name": "kubelet-dir",
            "mountPath": "/var/lib/kubelet",
            "storageClassName": "default",
            "volumeMode": "Filesystem"
          }
        ],
        "extraVolumesMount": [
          {
            "name": "kubelet-dir",
            "mountPath": "/var/lib/kubelet"
          }
        ],
        "extraManifestsMount": {
          "kubelet": {
            "mountPath": "/var/lib/kubelet"
          }
        },
        "extraManifestsMountSubPath": "kubelet"
      },
      "proxy": {
        "image": "k8s.gcr.io/kube-proxy:v1.28.2",
        "args": [
          "--config=/etc/kubernetes/proxy.conf",
          "--hostname-override=master.example.com",
          "--iptables-is-legacy=true",
          "--kubernetes-api-qps=10",
          "--metrics-bind-address=:10249",
          "--metrics-urls=https://127.0.0.1:10249",
          "--mode=iptables",
          "--nodeport-address=10.240.0.0/16",
          "--nodeport-range=30000-32767",
          "--oom-score-adj=-1000",
          "--oom-score-adj-filename=/etc/kubernetes/oom-score-adj.conf",
          "--udp-conntrack-minimal=1",
          "--user-space-ipvs=true"
        ],
        "extraArgs": {
          "v": "6"
        },
        "extraVolumes": [
          {
            "name": "proxy-dir",
            "mountPath": "/var/lib/kube-proxy",
            "storageClassName": "default",
            "volumeMode": "Filesystem"
          }
        ],
        "extraVolumesMount": [
          {
            "name": "proxy-dir",
            "mountPath": "/var/lib/kube-proxy"
          }
        ],
        "extraManifestsMount": {
          "proxy": {
            "mountPath": "/var/lib/kube-proxy"
          }
        },
        "extraManifestsMountSubPath": "proxy"
      }
    },
    "status": {
      "conditions": [
        {
          "type": "Ready",
          "status": "True",
          "lastTransitionTime": "2023-10-17T10:46:38Z",
          "reason": "KubeletReady"
        }
      ]
    }
  }
]
"dump.json" 12784L, 1309856B
```

As shown in the screenshot above, **kubectl cluster-info dump > dump.json** generates a cluster information dump and redirects the output to a file named **dump.json**.

Note: Examine the **dump.json** file to get the details of cluster's health

By following these steps, you have successfully enlisted a comprehensive collection of diagnostic information about the Kubernetes cluster, including details about the cluster's configuration, resources, and status.