**Assignment 17 – Kubernetes Test Questions 5**

Date: 17 May 2024

Name: Mohamad Haffiz Bin Mohd Hissham

**Questions:**

1.

**SSH** into the control plane node with **ssh cluster2-controlplane1**. Check how the control plane components **kubelet**, **kube-apiserver**, **kube-scheduler**, **kube-controller-manager**, and **etcd** are started/installed on the control plane node.

Also, find out the **name of the monitoring application** and **how it is started/installed** on the **control plane node**. **Write** your **findings** into the file **/opt/training/10/controlplane-components.txt**. The file should be **structured like**:

#/opt/training/10/controlplane-components.txt

**Kubelet: [Type]**

**Kube-apiserver: [Type]**

**Kube-scheduler: [Type]**

**Kube-controller-manager: [Type]**

**Etcd: [Type]**

**Monitoring: [Type] [Name]**

2.

Create a **namespace** named **techhub**.

Create a new **network policy** named **internal-access** in the techhub namespace.

Requirements:

* Network policy should allow pods within the techhub namespace to connect to each other **only on port 8080**. No other ports should be allowed.
* **No pods from outside** **of the** **techhub namespace** should be able **to connect to any pods inside** the **techhub** **namespace**.

**References:**

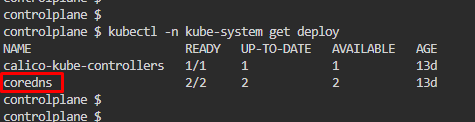
* [Constants and well-known values and paths | Kubernetes Implementation details](https://kubernetes.io/docs/reference/setup-tools/kubeadm/implementation-details/#constants-and-well-known-values-and-paths)
* [Kubernetes Namespace](https://www.aquasec.com/cloud-native-academy/kubernetes-101/kubernetes-namespace/)
* [Network Policies | Kubernetes](https://kubernetes.io/docs/concepts/services-networking/network-policies/#default-deny-all-ingress-traffic)

**Question 1:**

1. Killercoda by default already connected to controlplane node, so doesn’t need to open another SSD connection.
   1. But if applicable, connect to controlplane node, **ssh controlplane**
   2. A screen shot of a computer

      Description automatically generated
2. Get list of components available in controlplane, use **find /etc/kubernetes/manifests/**
   1. Or **ls /etc/kubernetes/manifests/**
   2. A screen shot of a computer

      Description automatically generated
3. Check kube-system pods running on controlplane, **kubectl -n kube-system get pod -o wide | grep controlplane**
   1. kube-system — a default space for Kubernetes system objects
      1. eg, **kube-dns** and **kube-proxy**, and **add-ons** providing **cluster-level features**, such as **web UI** **dashboards**, **ingresses**, and **cluster-level logging**
   2. A screen shot of a computer

      Description automatically generated
4. Check **kube-system deployed**, **kubectl -n kube-system get deploy**
   1. 
5. Create a directory as question specified, **mkdir -p /opt/training/10/**
   1. Create a text file named **controlplane-components.txt**, **nano -p /opt/training/10/controlplane-components.txt**
   2. Paste answer following the schema

**Kubelet: process**

**Kube-apiserver: static-pod**

**Kube-scheduler: static-pod**

**Kube-controller-manager: static-pod**

**Etcd: static-pod**

**Dns: pod coredns**

* 1. A screenshot of a computer program

     Description automatically generated

**Question 2:**

1. Create a **namespace** for **techhub**, **kubectl create ns techhub**
   1. **ns** = namespace, short notation
   2. A screen shot of a computer

      Description automatically generated
2. Check pod running with the namespace created
   1. **kubectl get pods -n techhub**
   2. **kubectl get pods -A**
   3. A screenshot of a computer

      Description automatically generated
3. Create a **new network policy** named **internal-access** in the **techhub namespace**, **nano internal-access.yaml**
   1. Apply the network policy script created, **kubectl apply -f internal-access.yaml**

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

  name: internal-access

  namespace: techhub

spec:

  podSelector: {}

  policyTypes:

  - Ingress

  ingress:

  - from:

    - podSelector: {}

    ports:

    - protocol: TCP

      port: 8080

* 1. A screen shot of a computer

     Description automatically generated

1. Check network policy created
   1. **kubectl get netpol -n techhub** or **kubectl describe netpol internal-access -n techhub**
      1. **netpol** = network policy, short notation
   2. A screenshot of a computer

      Description automatically generated