Lab 7 - Kubernetes

Initialize cluster

1. To free up resources, stop all vagrant VMs in devop-lab environment.
   - Clone the following Git repository :

   https://github.com/brahimhamdi/k8s-lab
   - In k8s-lab directory, execute following command to deploy k8s vagrants VMs :

     vagrant up

2. Kubernetes is already installed on all vagrant VMs. On master VM, initialize the cluster.

```
vagrant@k8s-master:~$ sudo kubeadm init --apiserver-advertise-address 192.168.205.100 --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.26.3
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [k8s-master kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc
.cluster.local] and IPs [10.96.0.1 192.168.205.100]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [k8s-master localhost] and IPs [192.168.205.100 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] etcd/peer serving cert is signed for DNS names [k8s-master localhost] and IPs [192.168.205.100 127.0.0.1 ::1]
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
```

   - Is there any errors ? How to resolve these errors ? *Nbre CPU <2, mém <1.7 Go, swap enable*
   - If no errors, what's the output of the initializing command ? *Join command*
   - Apply flannel yaml file.

```
args:
- --ip-masq
- --kube-subnet-mgr
- --iface=eth1
resources:
```

3. Check the cluster info.

```
vagrant@k8s-master:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.205.100:6443
CoreDNS is running at https://192.168.205.100:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
vagrant@k8s-master:~$
```

- How kubernetes components looks like ?

```
vagrant@k8s-master:~/dockercoins$ kubectl get all -n kube-system -o wide
NAME                                       READY   STATUS    RESTARTS        AGE   IP               NODE          NOMINATED NODE   READINESS GATES
pod/coredns-787d4945fb-k65w9               1/1     Running   3 (13m ago)     48m   10.244.0.50      k8s-master    <none>           <none>
pod/coredns-787d4945fb-kwx7v               1/1     Running   3 (13m ago)     48m   10.244.0.51      k8s-master    <none>           <none>
pod/etcd-k8s-master                        1/1     Running   32 (13m ago)    48m   192.168.205.100  k8s-master    <none>           <none>
pod/kube-apiserver-k8s-master              1/1     Running   11 (13m ago)    48m   192.168.205.100  k8s-master    <none>           <none>
pod/kube-controller-manager-k8s-master     1/1     Running   11 (13m ago)    48m   192.168.205.100  k8s-master    <none>           <none>
pod/kube-proxy-nmcqf                       1/1     Running   4 (13m ago)     48m   192.168.205.100  k8s-master    <none>           <none>
pod/kube-proxy-pw6cg                       1/1     Running   2 (5m21s ago)   26m   192.168.205.101  k8s-worker1   <none>           <none>
pod/kube-proxy-tb4q6                       1/1     Running   1 (2m59s ago)   24m   192.168.205.102  k8s-worker2   <none>           <none>
pod/kube-scheduler-k8s-master              1/1     Running   4 (13m ago)     48m   192.168.205.100  k8s-master    <none>           <none>

NAME               TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)                  AGE   SELECTOR
service/kube-dns   ClusterIP   10.96.0.10   <none>        53/UDP,53/TCP,9153/TCP   48m   k8s-app=kube-dns

NAME                        DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE   CONTAINERS   IMAGES                               SELECTOR
daemonset.apps/kube-proxy   3         3         3       3            3           kubernetes.io/os=linux   48m   kube-proxy   registry.k8s.io/kube-proxy:v1.26.3   k8s-app=kube-proxy

NAME                      READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES                                SELECTOR
deployment.apps/coredns   2/2     2            2           48m   coredns      registry.k8s.io/coredns/coredns:v1.9.3   k8s-app=kube-dns

NAME                                 DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES                                SELECTOR
replicaset.apps/coredns-787d4945fb   2         2         2       48m   coredns      registry.k8s.io/coredns/coredns:v1.9.3   k8s-app=kube-dns,pod-template-hash=787d4945fb
vagrant@k8s-master:~/dockercoins$
```

- What is the IP address of DNS systems ?

```
vagrant@k8s-master:~/dockercoins$ kubectl get all -n kube-system -o wide
NAME                                   READY   STATUS    RESTARTS        AGE   IP               NODE
pod/coredns-787d4945fb-k65w9           1/1     Running   3 (13m ago)     48m   10.244.0.50      k8s-master
pod/coredns-787d4945fb-kwx7v           1/1     Running   3 (13m ago)     48m   10.244.0.51      k8s-master
```

4. Join all nodes to the cluster.

```
vagrant@k8s-master:~$ sudo kubeadm token create --print-join-command
kubeadm join 192.168.205.100:6443 --token k8ppr5.tay9cvsfq2pj81io --discovery-token-ca-cert-hash sha256:5ecd1652d18e0b34ace14af6f1bc303921ce8a
03520040320cad4dbf55a28f47
vagrant@k8s-master:~$

vagrant@k8s-worker1:~$ sudo kubeadm join 192.168.205.100:6443 --token idgap9.r77clmsp3tnikc7w --discovery-token-ca-cert-hash sha256:5ecd1652d1
8e0b34ace14af6f1bc303921ce8a03520040320cad4dbf55a28f47
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

vagrant@k8s-worker1:~$ logout

vagrant@k8s-worker1:~$ sudo kubeadm join 192.168.205.100:6443 --token idgap9.r77clmsp3tnikc7w --discovery-token-ca-cert-hash sha256:5ecd1652d1
8e0b34ace14af6f1bc303921ce8a03520040320cad4dbf55a28f47
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

vagrant@k8s-worker1:~$ logout
brahim@Training:~/k8s-lab$
```

- On the master, check that all nodes are ready on the cluster.

```
vagrant@k8s-master:~/dockercoins$ kubectl get node -o wide
NAME          STATUS   ROLES           AGE   VERSION   INTERNAL-IP       EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION      CONTAINER-RUNTIME
k8s-master    Ready    control-plane   51m   v1.26.2   192.168.205.100   <none>        Ubuntu 20.04.5 LTS   5.4.0-132-generic   containerd://1.6.19
k8s-worker1   Ready    <none>          29m   v1.26.3   192.168.205.101   <none>        Ubuntu 20.04.5 LTS   5.4.0-132-generic   containerd://1.6.19
k8s-worker2   Ready    <none>          27m   v1.26.3   192.168.205.102   <none>        Ubuntu 20.04.5 LTS   5.4.0-132-generic   containerd://1.6.19
vagrant@k8s-master:~/dockercoins$
```

Manage pods

5. Create a yaml file for a *hasher* pod.

```
vagrant@k8s-master:~$ vim hasher.yaml
ivagrant@k8s-master:~$ cat hasher.yaml
apiVersion: v1
kind: Pod
metadata:
  name: hasher
  labels:
    app: dockercoins
spec:
  containers:
    - name: hasher
      image: brahimhamdi/hasher
vagrant@k8s-master:~$
```

6. Apply the yaml file.

```
vagrant@k8s-master:~$ kubectl apply -f hasher.yaml
pod/hasher created
```

```
vagrant@k8s-master:~$ kubectl describe pod hasher
Name:             hasher
Namespace:        default
Priority:         0
Service Account:  default
Node:             k8s-worker2/192.168.205.102
Start Time:       Tue, 28 Mar 2023 11:51:40 +0000
Labels:           app=dockercoins
Annotations:      <none>
Status:           Running
IP:               10.244.2.26
IPs:
  IP:  10.244.2.26
Containers:
  hasher:
    Container ID:   containerd://496fe513411fc325cd93b5524f804dcf9c8d3b28c61f1d84d7575530ef0f67fb
    Image:          brahimhamdi/hasher
    Image ID:       docker.io/brahimhamdi/hasher@sha256:a37377f07840109415eb7df07ae830bc617d0f3ac3c98c904b7a8647868785f5
```

- On which node the pod is created ?
- What is the pod's IP address ?
- What is the container's name and ID ?
- What is the image's name and ID ?

7. Remove the pod from the cluster.

```
vagrant@k8s-master:~$ kubectl delete pod hasher
pod "hasher" deleted
vagrant@k8s-master:~$ kubectl get pod
No resources found in default namespace.
vagrant@k8s-master:~$
```

Manage deployments and services

8. Create yaml file to describe *dockercoins* application deployment.

```
vagrant@k8s-master:~/dockercoins$ kubectl create namespace dockercoins
namespace/dockercoins created
vagrant@k8s-master:~/dockercoins$ kubectl get namespace
NAME              STATUS   AGE
default           Active   29m
dockercoins       Active   29s
kube-flannel      Active   29m
kube-node-lease   Active   29m
kube-public       Active   29m
kube-system       Active   29m
vagrant@k8s-master:~/dockercoins$
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: worker
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dockercoins
      tier: worker
  template:
    metadata:
      name: worker
      labels:
        app: dockercoins
        tier: worker
    spec:
      containers:
        - name: worker
          image: brahimhamdi/worker
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: rng
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dockercoins
      tier: rng
  template:
    metadata:
```

```
    name: rng
    labels:
     app: dockercoins
     tier: rng
   spec:
    containers:
     - name: rng
       image: brahimhamdi/rng
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hasher
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dockercoins
      tier: hasher
  template:
    metadata:
      name: hasher
      labels:
        app: dockercoins
        tier: hasher
    spec:
      containers:
        - name: hasher
          image: brahimhamdi/hasher
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dockercoins
      tier: redis
  template:
    metadata:
      name: redis
      labels:
        app: dockercoins
        tier: redis
    spec:
      containers:
```

```
      - name: redis
        image: redis
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webui
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dockercoins
      tier: front
  template:
    metadata:
      name: webui
      labels:
        app: dockercoins
        tier: front
    spec:
      containers:
      - name: webui
        image: brahimhamdi/webui
---
apiVersion: v1
kind: Service
metadata:
  name: rng
spec:
  selector:
    app: dockercoins
    tier: rng
  type: ClusterIP
  ports:
  - port: 80
    targetPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: hasher
spec:
  selector:
    app: dockercoins
    tier: hasher
  type: ClusterIP
  ports:
  - port: 80
```

```
    targetPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: redis
spec:
  selector:
    app: dockercoins
    tier: redis
  type: ClusterIP
  ports:
  - port: 6379
    targetPort: 6379
---
apiVersion: v1
kind: Service
metadata:
  name: webui
spec:
  selector:
    app: dockercoins
    tier: front
  type: NodePort
  ports:
  - port: 80
    targetPort: 80
    nodePort: 30001
```

9. Apply the yaml file and check the application.

```
vagrant@k8s-master:~/dockercoins$ kubectl apply -f dockercoins.yaml
deployment.apps/worker created
deployment.apps/rng created
deployment.apps/hasher created
deployment.apps/redis created
deployment.apps/webui created
service/rng created
service/hasher created
service/redis created
```

```
vagrant@k8s-master:~/dockercoins$ kubectl get all -o wide
NAME                          READY   STATUS    RESTARTS   AGE     IP            NODE          NOMINATED NODE   READINESS GATES
pod/hasher-77f9b9d56c-mxx8j   1/1     Running   0          5m10s   10.244.2.35   k8s-worker2   <none>           <none>
pod/redis-84b4f5f9c8-gkjpk    1/1     Running   0          5m10s   10.244.1.67   k8s-worker1   <none>           <none>
pod/rng-64cd6f5c47-98f2h      1/1     Running   0          5m10s   10.244.1.66   k8s-worker1   <none>           <none>
pod/webui-599669698c-mb2ls    1/1     Running   0          115s    10.244.2.36   k8s-worker2   <none>           <none>
pod/worker-5f47c9497b-dw9kg   1/1     Running   0          5m10s   10.244.2.34   k8s-worker2   <none>           <none>

NAME                 TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE     SELECTOR
service/hasher       ClusterIP   10.105.67.75     <none>        80/TCP           5m10s   app=dockercoins
service/kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP          59m     <none>
service/redis        ClusterIP   10.109.132.240   <none>        6379/TCP         5m10s   app=dockercoins
service/rng          ClusterIP   10.101.233.9     <none>        80/TCP           5m10s   app=dockercoins
service/webui        NodePort    10.103.175.69    <none>        80:30001/TCP     5m10s   app=dockercoins,tier=front

NAME                     READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS   IMAGES               SELECTOR
deployment.apps/hasher   1/1     1            1           5m10s   hasher       brahimhamdi/hasher   app=dockercoins
deployment.apps/redis    1/1     1            1           5m10s   redis        redis                app=dockercoins
deployment.apps/rng      1/1     1            1           5m10s   rng          brahimhamdi/rng      app=dockercoins
deployment.apps/webui    1/1     1            1           5m10s   webui        brahimhamdi/webui    app=dockercoins
deployment.apps/worker   1/1     1            1           5m10s   worker       brahimhamdi/worker   app=dockercoins

NAME                                DESIRED   CURRENT   READY   AGE     CONTAINERS   IMAGES               SELECTOR
replicaset.apps/hasher-77f9b9d56c   1         1         1       5m10s   hasher       brahimhamdi/hasher   app=dockercoins,pod-template-hash=77f9b9d56c
replicaset.apps/redis-84b4f5f9c8    1         1         1       5m10s   redis        redis                app=dockercoins,pod-template-hash=84b4f5f9c8
replicaset.apps/rng-64cd6f5c47      1         1         1       5m10s   rng          brahimhamdi/rng      app=dockercoins,pod-template-hash=64cd6f5c47
replicaset.apps/webui-599669698c    1         1         1       115s    webui        brahimhamdi/webui    app=dockercoins,pod-template-hash=599669698c
replicaset.apps/webui-8567cbbbb     0         0         0       5m10s   webui        brahimhamdi/webui    app=dockercoins,pod-template-hash=8567cbbbb
replicaset.apps/worker-5f47c9497b   1         1         1       5m10s   worker       brahimhamdi/worker   app=dockercoins,pod-template-hash=5f47c9497b
vagrant@k8s-master:~/dockercoins$
```