

Module 9: Container Orchestration using Kubernetes Part-II

1. Set Up Kubernetes Cluster and verify.

```
root@kube-master:~# kubectl cluster-info
Kubernetes control plane is running at https://10.128.0.17:6443
CoreDNS is running at https://10.128.0.17:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

```
root@kube-master:~#
```

```
root@kube-master:~# kubectl get node
```

NAME	STATUS	ROLES	AGE	VERSION
kube-master	Ready	control-plane	29m	v1.29.12
wnode1	Ready	<none>	3m42s	v1.29.12
wnode2	Ready	<none>	103s	v1.29.12

```
root@kube-master:~#
```

2. Create a Namespace: It's a good practice to use a dedicated namespace for your application.

```
root@kube-master:~# kubectl create namespace multi-tier-app
namespace/multi-tier-app created
root@kube-master:~#
```

3. Created a MongoDB Deployment and Service.

```
root@kube-master:~# cat mongo-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongo
  namespace: multi-tier-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: mongo
  template:
    metadata:
      labels:
        app: mongo
    spec:
      containers:
        - name: mongo
          image: mongo
          ports:
            - containerPort: 27017
```

```
root@kube-master:~# cat mongo-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: mongo
  namespace: multi-tier-app
spec:
  ports:
    - port: 27017
      targetPort: 27017
  selector:
    app: mongo
```

4. Apply the MongoDB configuration.

```
root@kube-master:~# kubectl apply -f mongo-deployment.yaml
deployment.apps/mongo created
root@kube-master:~# kubectl apply -f mongo-service.yaml
service/mongo created
root@kube-master:~#
```

Verify:

```
root@kube-master:~# kubectl get pods -n multi-tier-app
NAME                                READY   STATUS    RESTARTS   AGE
mongo-5cd597596c-8plnb             1/1     Running   0           2m40s
mongo-5cd597596c-fxbzb             1/1     Running   0           2m40s
root@kube-master:~#
```

```
root@kube-master:~# kubectl get svc -n multi-tier-app
NAME      TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE
mongo     ClusterIP   10.106.2.235  <none>        27017/TCP        3m20s
root@kube-master:~#
```

5. Created the NodeJS Application Deployment and Service.

```
root@kube-master:~# cat nodejs-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: employee
  namespace: multi-tier-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: employee
  template:
    metadata:
      labels:
        app: employee
    spec:
      containers:
      - name: employee
        image: devopsedu/employee
        ports:
        - containerPort: 8888
        env:
        - name: DB_HOST
          value: "mongo:27017"
```

```
root@kube-master:~# cat nodejs-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: employee
  namespace: multi-tier-app
spec:
  ports:
  - port: 8888
    targetPort: 8888
  selector:
    app: employee
  type: LoadBalancer
```

Apply the NodeJS configuration:

```
root@kubernetes-master:~# kubectl apply -f nodejs-deployment.yaml
deployment.apps/employee created
root@kubernetes-master:~# kubectl apply -f nodejs-service.yaml
service/employee created
root@kubernetes-master:~#
```

Verify:

```
root@kubernetes-master:~# kubectl get pods -n multi-tier-app
NAME                                READY    STATUS    RESTARTS    AGE
employee-6db66f8bd6-q8wgh          1/1      Running   0            2m54s
employee-6db66f8bd6-tf98j          1/1      Running   0            2m54s
mongo-5cd597596c-8plnb             1/1      Running   0            8m45s
mongo-5cd597596c-fxbzb             1/1      Running   0            8m45s
root@kubernetes-master:~#
```

```
root@kubernetes-master:~# kubectl get svc -n multi-tier-app
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
employee        LoadBalancer  10.105.199.180  <pending>        8888:30352/TCP   3m23s
mongo           ClusterIP      10.106.2.235    <none>           27017/TCP        9m13s
root@kubernetes-master:~#
```

6. Verify deployment and services.

```
Service/nodejs app configured
root@kubernetes-master:~# kubectl get nodes -o wide
NAME            STATUS    ROLES    AGE    VERSION    INTERNAL-IP    EXTERNAL-IP    OS-IMAGE           KERNEL-VERSION    CONTAINER-RUNTIME
kubernetes-master Ready    control-plane  9h    v1.29.12    10.128.0.17    <none>         Ubuntu 20.04.6 LTS  5.15.0-1073-gcp   docker://27.4.1
wnode1          Ready    <none>      9h    v1.29.12    10.128.0.18    <none>         Ubuntu 20.04.6 LTS  5.15.0-1073-gcp   docker://27.4.1
wnode2          Ready    <none>      9h    v1.29.12    10.128.0.19    <none>         Ubuntu 20.04.6 LTS  5.15.0-1073-gcp   docker://27.4.1
root@kubernetes-master:~# kubectl get pods -o wide
NAME                                READY    STATUS    RESTARTS    AGE    IP              NODE    NOMINATED NODE    READINESS GATES
mongo-7d47b5474b-cs758              1/1      Running   0            8m17s  192.168.130.80  wnode2  <none>             <none>
nodejs-app-5cc4cf8b94-9jshj         1/1      Running   0            6m38s  192.168.121.145 wnode1  <none>             <none>
root@kubernetes-master:~# kubectl get svc -o wide
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE    SELECTOR
kubernetes      ClusterIP      10.96.0.1        <none>           443/TCP          13m    <none>
mongo           ClusterIP      10.109.103.146   <none>           27017/TCP        8m31s  app=mongo
nodejs-app      NodePort       10.100.127.91    <none>           8888:30001/TCP   6m52s  app=nodejs-app
root@kubernetes-master:~#
```