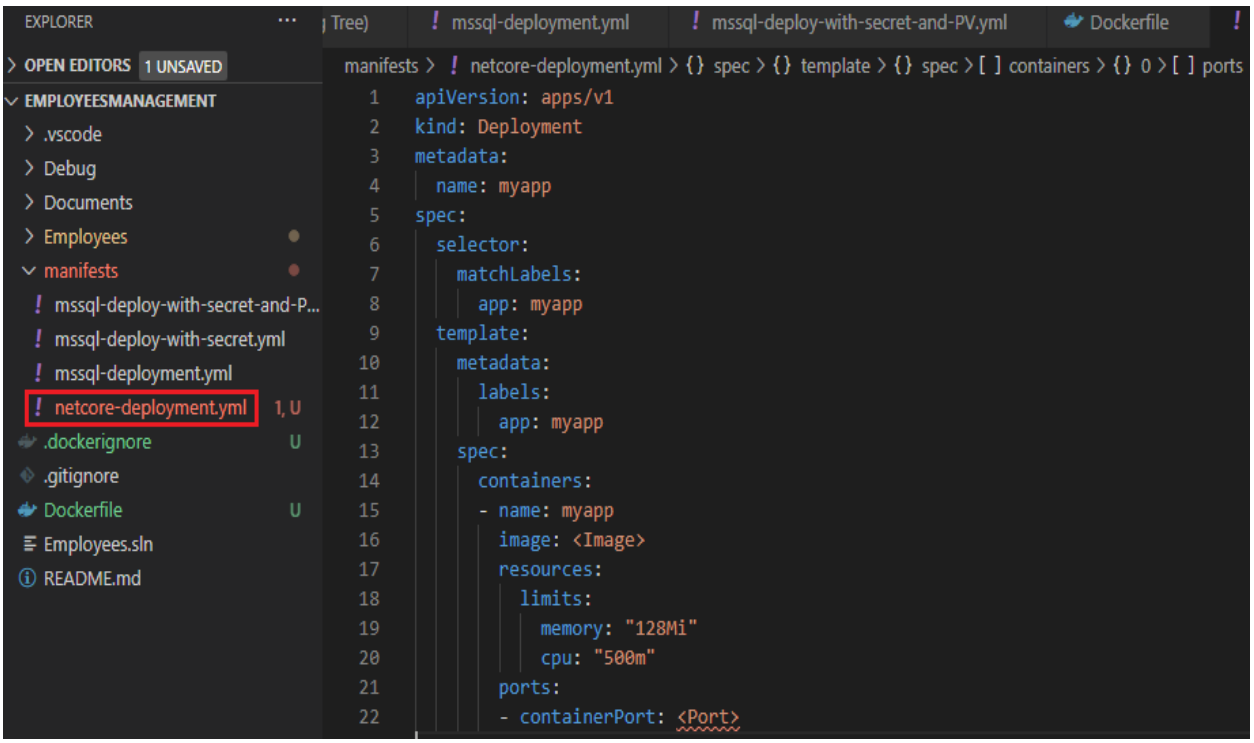
Lesson7 - Deploy netcore web application

In lesson6 we created docker image for the .NET core web application and pushed it to DockerHub. In this lesson we will deploy the docker images to Kuberneties cluster with the MSSQL that already deployed.

Create netcore-deployment.yml:

In lesson2 we saw how to deploy MSSQL with YML file. In this lesson we learn how to create deployment YML from scratch. First we need to create **netcore-deployment.yml** in manifests folder. Next write "deploy" in the empty file and click TAB, VS code will create default deployment template as below.



Update the netcore-deployment.yml as below:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: employee-deployment
  namespace: employee
spec:
  replicas: 1
  selector:
    matchLabels:
      app: employee-pod
  template:
    metadata:
      labels:
        app: employee-pod
    spec:
      containers:
      - name: employee-pod
        image: yaronzlotolov/employees:v1
        resources:
          limits:
            memory: "128Mi"
            cpu: "500m"
        env:
        - name: ConnectionStrings__ConnectionString
          valueFrom:
              secretKeyRef:
                name: mssql-secret
                key: ConnectionString
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: employee-service
  namespace: employee
spec:
  selector:
    app: employee-pod
  ports:
  - port: 8080
    targetPort: 80
  type: LoadBalancer
```

Indicates that the Pod run one container, employee-pod. This is the desired state.

Defines how the Deployment finds which Pods to manage, employees-pod is the name of the pod.

Pull employees docker image (the container name in this example is also employee-pod) from docker hub at version v1 and run it in pod that match the label employee-pod.

Get the connection string to the database in mssql-pod from mssql-secret that was deployed with MSSQL deployment.

ConnectionStrings__ConnectionString is related to appsettings.json:

```
"ConnectionStrings": {
    "ConnectionString":
```

Connection between the Internal ports in the service <>in the pod.

The service exposes external port

The service provides external IP which is Localhost in Docker Desktop

Check the mssql-deployment and its secret already deployed to employee namespace:

kubectl get all -n employee

kubectl get secret -n employee

```
PS C:\Kuberneties\EmployeesManagement\manifests> kubectl get all -n employee
NAME                                     READY   STATUS    RESTARTS   AGE
pod/mssql-deployment-6bcb97764c-m675k    1/1     Running   0          4m24s

NAME                    TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/mssql-service   LoadBalancer   10.98.233.140   localhost     1433:30878/TCP   4m24s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mssql-deployment    1/1     1            1           4m24s

NAME                                           DESIRED   CURRENT   READY   AGE
replicaset.apps/mssql-deployment-6bcb97764c    1         1         1       4m24s
PS C:\Kuberneties\EmployeesManagement\manifests> kubectl get secret -n employee
NAME                  TYPE                                  DATA   AGE
default-token-25bfc   kubernetes.io/service-account-token   3      4m50s
mssql-secret          Opaque                                2      69s
PS C:\Kuberneties\EmployeesManagement\manifests>
```

Apply the netcore-deployment.yml:

cd .\manifests

kubectl apply -f netcore-deployment.yml

```
PS C:\Kuberneties\EmployeesManagement> cd .\manifests\
PS C:\Kuberneties\EmployeesManagement\manifests> kubectl apply -f netcore-deployment.yml
deployment.apps/employee-deployment created
service/employee-services created
PS C:\Kuberneties\EmployeesManagement\manifests>
```

kubectl get all -n employee

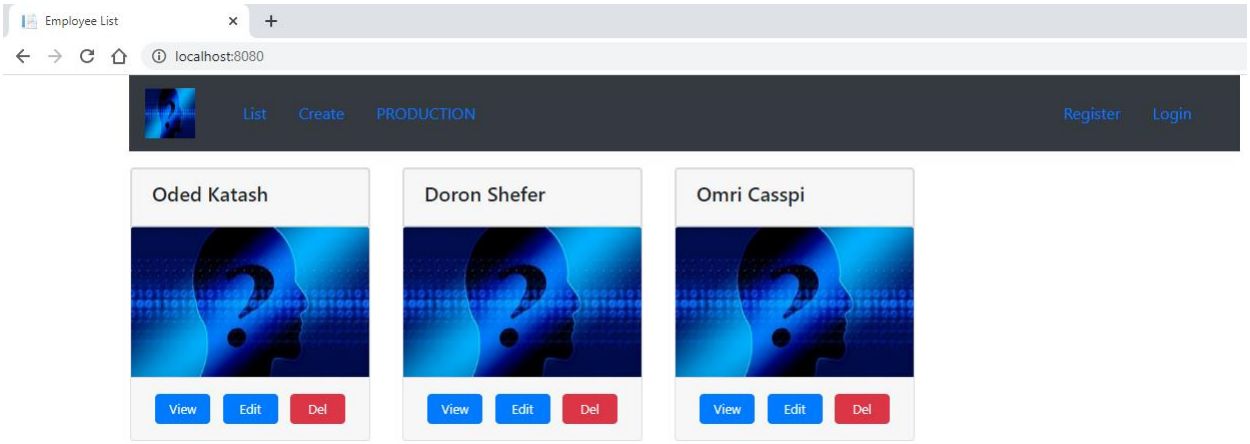
```
PS C:\Kuberneties\EmployeesManagement\manifests> kubectl get all -n employee
NAME                                       READY   STATUS    RESTARTS   AGE
pod/employee-deployment-58684b99cd-qp9tb   1/1     Running   0          21s
pod/mssql-deployment-6bcb97764c-m675k      1/1     Running   0          11m

NAME                        TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/employee-services   LoadBalancer   10.108.117.129   localhost     8080:30993/TCP   21s
service/mssql-service       LoadBalancer   10.98.233.140    localhost     1433:30878/TCP   11m

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/employee-deployment   1/1     1            1           22s
deployment.apps/mssql-deployment      1/1     1            1           11m

NAME                                             DESIRED   CURRENT   READY   AGE
replicaset.apps/employee-deployment-58684b99cd   1         1         1       22s
replicaset.apps/mssql-deployment-6bcb97764c      1         1         1       11m
PS C:\Kuberneties\EmployeesManagement\manifests>
```

chrome > localhost:8080



Done!