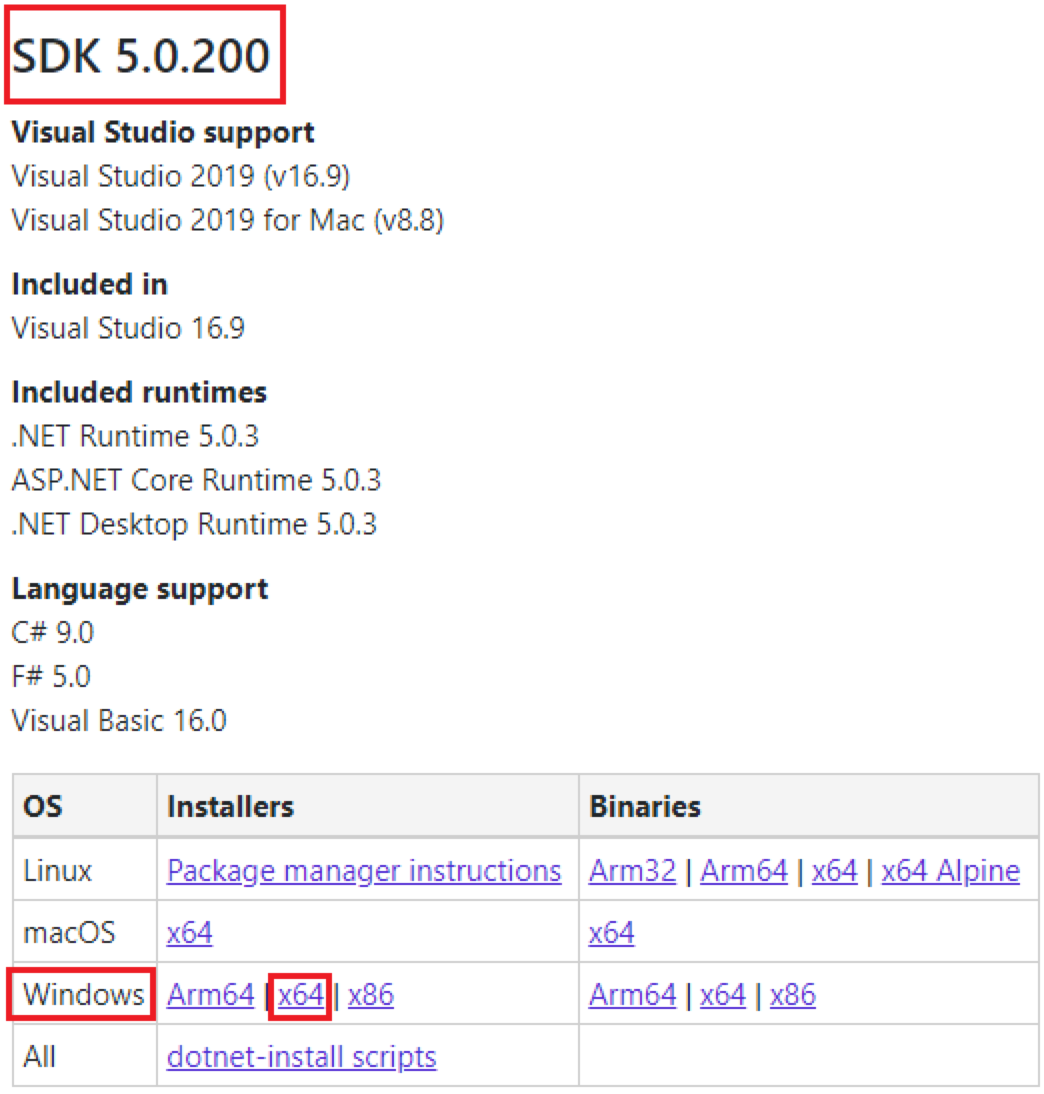
Lesson12 – Upgrade to NET 5.0 with Green Deployment

In this lesson we will learn how to upgrade .NET core 3.1 we b application to NET-5 and deploy it to Kubernetes cluster.

Install .NET 5.0 SDK from: https://dotnet.microsoft.com/download/dotnet/5.0



SDK 5.0.200

Visual Studio support
Visual Studio 2019 (v16.9)
Visual Studio 2019 for Mac (v8.8)

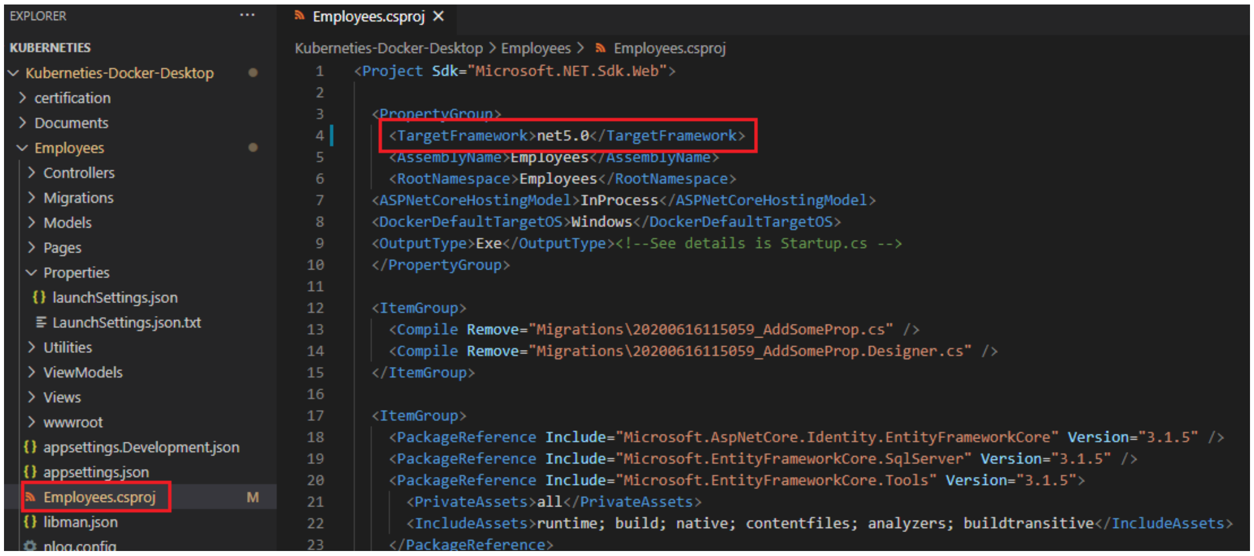
Included in
Visual Studio 16.9

Included runtimes
.NET Runtime 5.0.3
ASP.NET Core Runtime 5.0.3
.NET Desktop Runtime 5.0.3

Language support
C# 9.0
F# 5.0
Visual Basic 16.0

OS	Installers	Binaries
Linux	Package manager instructions	Arm32 | Arm64 | x64 | x64 Alpine
macOS	x64	x64
Windows	Arm64 | x64 | x86	Arm64 | x64 | x86
All	dotnet-install scripts	

Open VS Code and change the Target Framework to .net5.0:



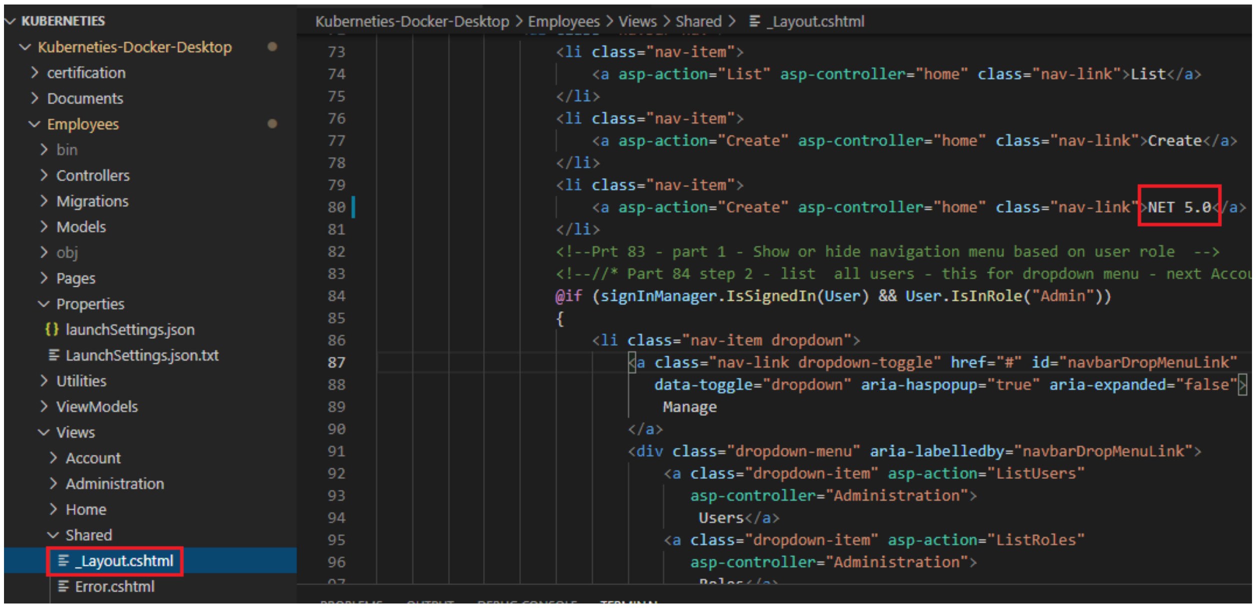
```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net5.0</TargetFramework>
    <AssemblyName>Employees</AssemblyName>
    <RootNamespace>Employees</RootNamespace>
  <ASPNetCoreHostingModel>InProcess</ASPNetCoreHostingModel>
  <DockerDefaultTargetOS>Windows</DockerDefaultTargetOS>
  <OutputType>Exe</OutputType><!--See details is Startup.cs -->
  </PropertyGroup>

  <ItemGroup>
    <Compile Remove="Migrations\20200616115059_AddSomeProp.cs" />
    <Compile Remove="Migrations\20200616115059_AddSomeProp.Designer.cs" />
  </ItemGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.Identity.EntityFrameworkCore" Version="3.1.5" />
    <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="3.1.5" />
    <PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="3.1.5">
      <PrivateAssets>all</PrivateAssets>
      <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>
    </PackageReference>
```

Open Views>Shared> _Layout.cshtml and change the label to NET 5.0 to show the change:



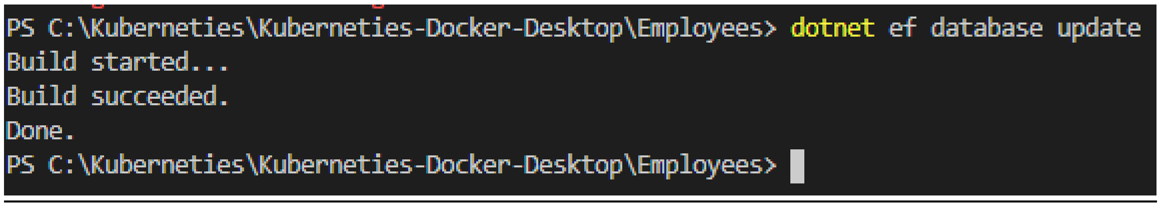
```
                <li class="nav-item">
                    <a asp-action="List" asp-controller="home" class="nav-link">List</a>
                </li>
                <li class="nav-item">
                    <a asp-action="Create" asp-controller="home" class="nav-link">Create</a>
                </li>
                <li class="nav-item">
                    <a asp-action="Create" asp-controller="home" class="nav-link">NET 5.0</a>
                </li>
                <!--Prt 83 - part 1 - Show or hide navigation menu based on user role  -->
                <!--//* Part 84 step 2 - list  all users - this for dropdown menu - next Accou
                @if (signInManager.IsSignedIn(User) && User.IsInRole("Admin"))
                {
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#" id="navbarDropMenuLink"
                           data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                            Manage
                        </a>
                        <div class="dropdown-menu" aria-labelledby="navbarDropMenuLink">
                            <a class="dropdown-item" asp-action="ListUsers"
                               asp-controller="Administration">
                                Users</a>
                            <a class="dropdown-item" asp-action="ListRoles"
                               asp-controller="Administration">
```

Update the connection string in appsettings.json to your local database:

"server=(localdb)\\MSSQLLocalDB;database=EmployeeDB;Trusted_Connection=true;MultipleActiveResultSets=true"

```
"ConnectionStrings": {
  "ConnectionString": "server=(localdb)\\MSSQLLocalDB;database=EmployeeDB;Trusted_Connection=true;MultipleActiveResultSets=true"
}
```

dotnet ef database update


```
PS C:\Kuberneties\Kuberneties-Docker-Desktop\Employees> dotnet ef database update
Build started...
Build succeeded.
Done.
PS C:\Kuberneties\Kuberneties-Docker-Desktop\Employees>
```

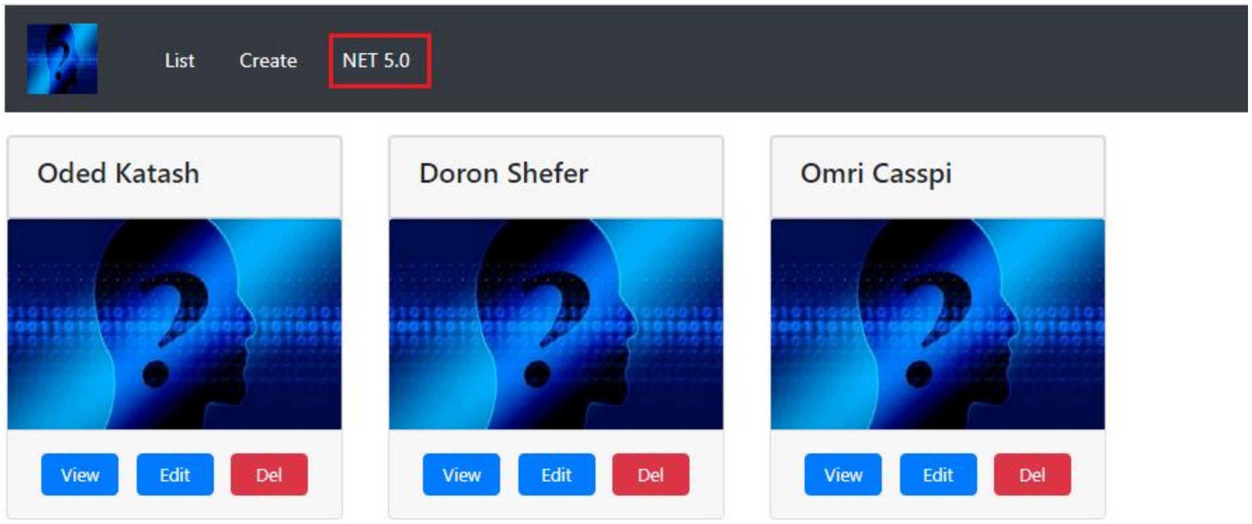
Build and test the application:

cd .\Employees

dotnet build

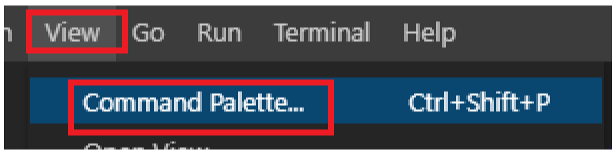
dotnet run

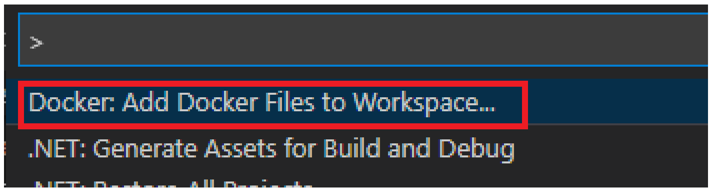
Chrome: Localhost:5000



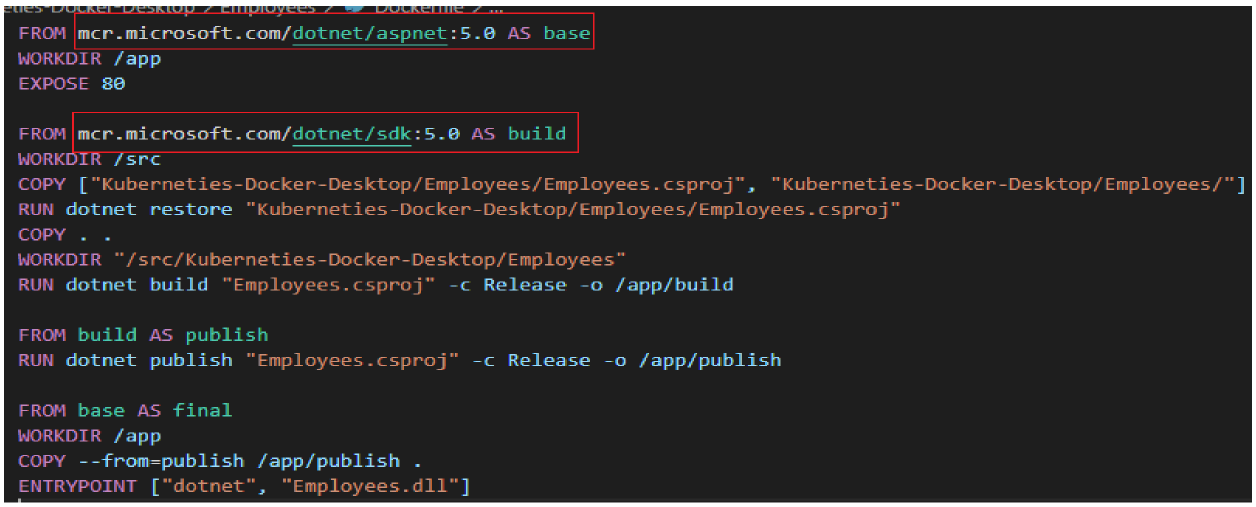
Commit and push to GitHub.

Create Dockerfile with .NET 5.0:




>
Docker: Add Docker Files to Workspace...
.NET: Generate Assets for Build and Debug

The dotnet that will be pulled from Docker Hub is NET 5.0:



```
FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
WORKDIR /app
EXPOSE 80

FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
WORKDIR /src
COPY ["Kuberneties-Docker-Desktop/Employees/Employees.csproj", "Kuberneties-Docker-Desktop/Employees/"]
RUN dotnet restore "Kuberneties-Docker-Desktop/Employees/Employees.csproj"
COPY . .
WORKDIR "/src/Kuberneties-Docker-Desktop/Employees"
RUN dotnet build "Employees.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "Employees.csproj" -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "Employees.dll"]
```

Create docker image and push to Docker Hub (Lesson6)

Create Docker image:

Move the Dockerfile two folders up.

docker build -t employees:v2 .

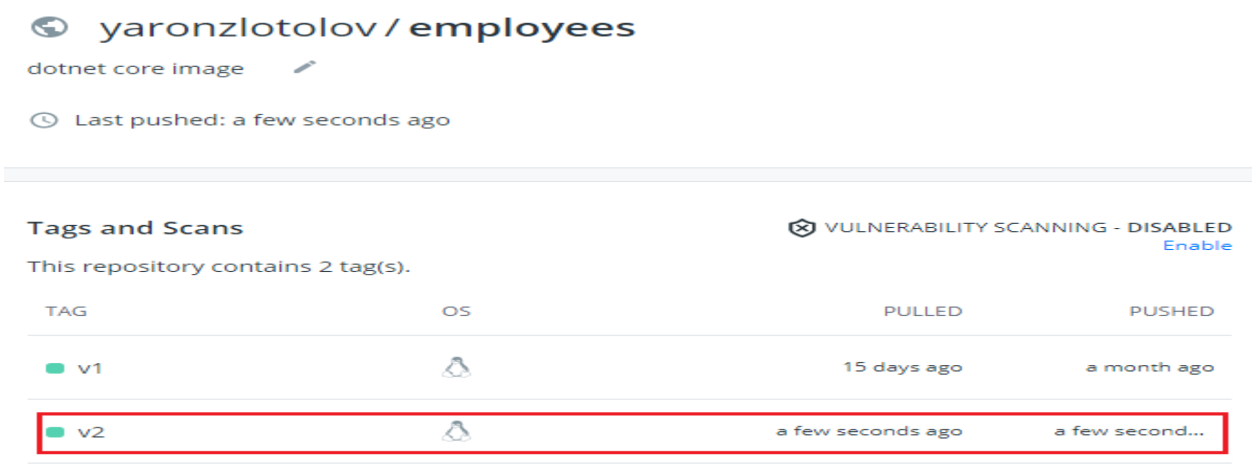
docker images

Push the image to docker hub:

docker login

docker tag employees:v2 yaronzlotolov/employees:v2

docker push yaronzlotolov/employees:v2



yaronzlotolov / employees

dotnet core image

Last pushed: a few seconds ago

Tags and Scans

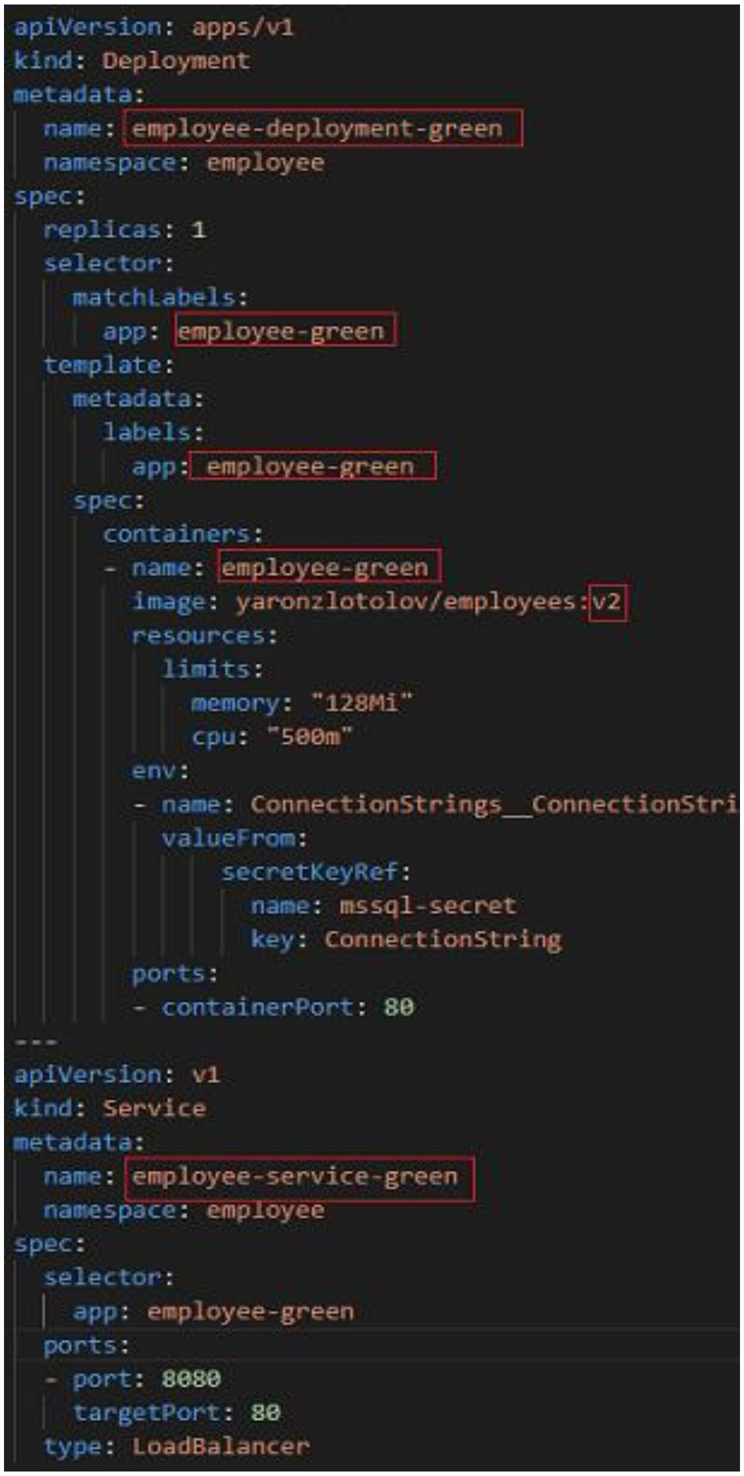
VULNERABILITY SCANNING - DISABLED
Enable

This repository contains 2 tag(s).

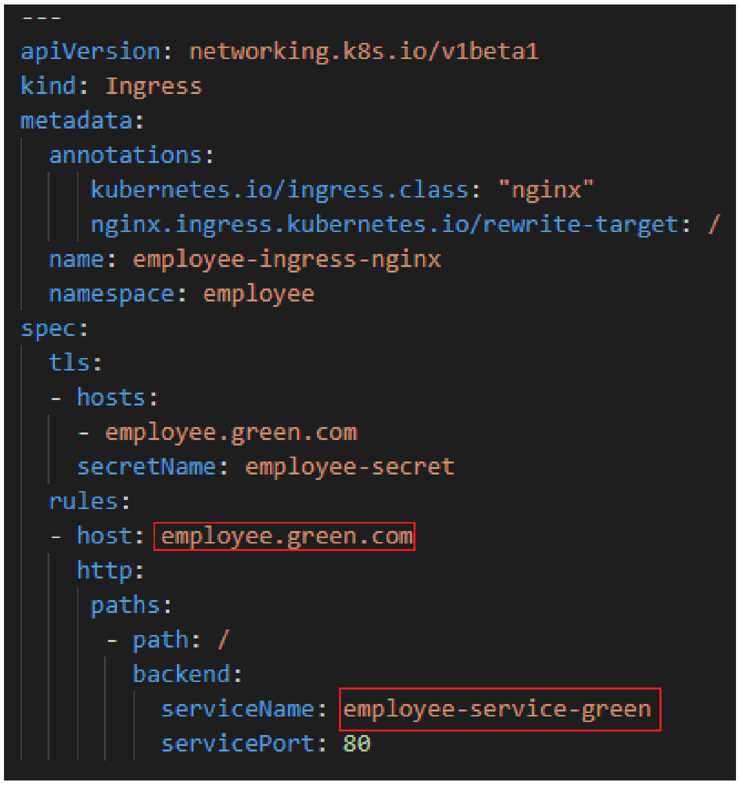
TAG	OS	PULLED	PUSHED
v1		15 days ago	a month ago
v2		a few seconds ago	a few second...

Green Deployment with .NET 5.0

Create netcore-deploy-green.yml for the NET 5.0:



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: employee-deployment-green
  namespace: employee
spec:
  replicas: 1
  selector:
    matchLabels:
      app: employee-green
  template:
    metadata:
      labels:
        app: employee-green
    spec:
      containers:
      - name: employee-green
        image: yaronzlotolov/employees:v2
        resources:
          limits:
            memory: "128Mi"
            cpu: "500m"
        env:
        - name: ConnectionStrings__ConnectionStri
          valueFrom:
              secretKeyRef:
                name: mssql-secret
                key: ConnectionString
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: employee-service-green
  namespace: employee
spec:
  selector:
    app: employee-green
  ports:
  - port: 8080
    targetPort: 80
  type: LoadBalancer
```



```yaml
---
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: "nginx"
    nginx.ingress.kubernetes.io/rewrite-target: /
  name: employee-ingress-nginx
  namespace: employee
spec:
  tls:
  - hosts:
    - employee.green.com
    secretName: employee-secret
  rules:
  - host: employee.green.com
    http:
     paths:
      - path: /
        backend:
          serviceName: employee-service-green
          servicePort: 80
```

<u>In case this is the first deployment to employee namespace run the commands below:</u>

cd .\certification\

kubectl create ns employee

kubectl create secret tls employee-secret --key privkey.pem --cert cert.pem -n employee

kubectl create secret generic mssql-secret --namespace=employee --from-literal='ConnectionString="server=mssql-service;Initial Catalog=EmployeeDB;Persist Security Info=False;User ID=sa;Password=MyDemoPwd2021!;MultipleActiveResultSets=true"' --from-literal='SA_PASSWORD=MyDemoPwd2021!'

cd .\manifests\

kubectl apply -f .\ingress-nginx-deployment.yml

kubectl apply -f .\mssql-deploy-with-secret-and-pv.yml

cd .\Employees\

Environment Variables: server=localhost,1433;Initial Catalog=EmployeeDB;Persist Security Info=False;User ID=sa;Password=MyDemoPwd2021!;MultipleActiveResultSets=true

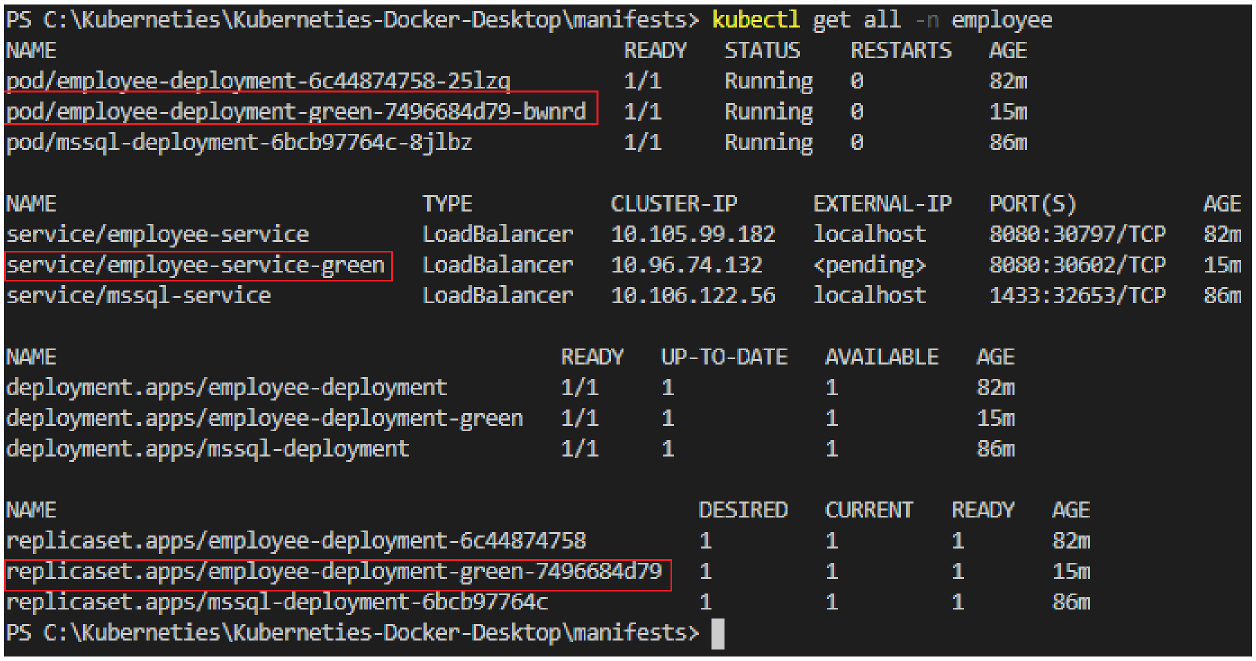
dotnet ef database update

Deploy netcore-deploy-green.yml:

Add employee.green.com -> C:\Windows\System32\drivers\etc\hosts

cd .\manifests

kubectl apply -f netcore-deploy-green.yml



```
PS C:\Kuberneties\Kuberneties-Docker-Desktop\manifests> kubectl get all -n employee
NAME                                              READY   STATUS    RESTARTS   AGE
pod/employee-deployment-6c44874758-25lzq          1/1     Running   0          82m
pod/employee-deployment-green-7496684d79-bwnrd    1/1     Running   0          15m
pod/mssql-deployment-6bcb97764c-8jlbz             1/1     Running   0          86m

NAME                               TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/employee-service           LoadBalancer   10.105.99.182   localhost     8080:30797/TCP   82m
service/employee-service-green     LoadBalancer   10.96.74.132    <pending>     8080:30602/TCP   15m
service/mssql-service              LoadBalancer   10.106.122.56   localhost     1433:32653/TCP   86m

NAME                                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/employee-deployment         1/1     1            1           82m
deployment.apps/employee-deployment-green   1/1     1            1           15m
deployment.apps/mssql-deployment            1/1     1            1           86m

NAME                                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/employee-deployment-6c44874758         1         1         1       82m
replicaset.apps/employee-deployment-green-7496684d79   1         1         1       15m
replicaset.apps/mssql-deployment-6bcb97764c            1         1         1       86m
PS C:\Kuberneties\Kuberneties-Docker-Desktop\manifests>
```

Troubleshoot for DB creation (sometimes the connection string in environment variable is wrong):

kubectl -n employee exec -it pod/mssql-deployment-6bcb97764c-8jlbz -- /bin/sh

ls -ltr /var/opt/mssql/data

Done!