Project:

Deploying .NET app using Kubernetes Manifest Files

🔧 Prerequisites

- Docker installed and running
- Docker Hub account
- Kubernetes cluster (e.g., AKS)
- kubectl configured
- GitHub repo with your .NET app

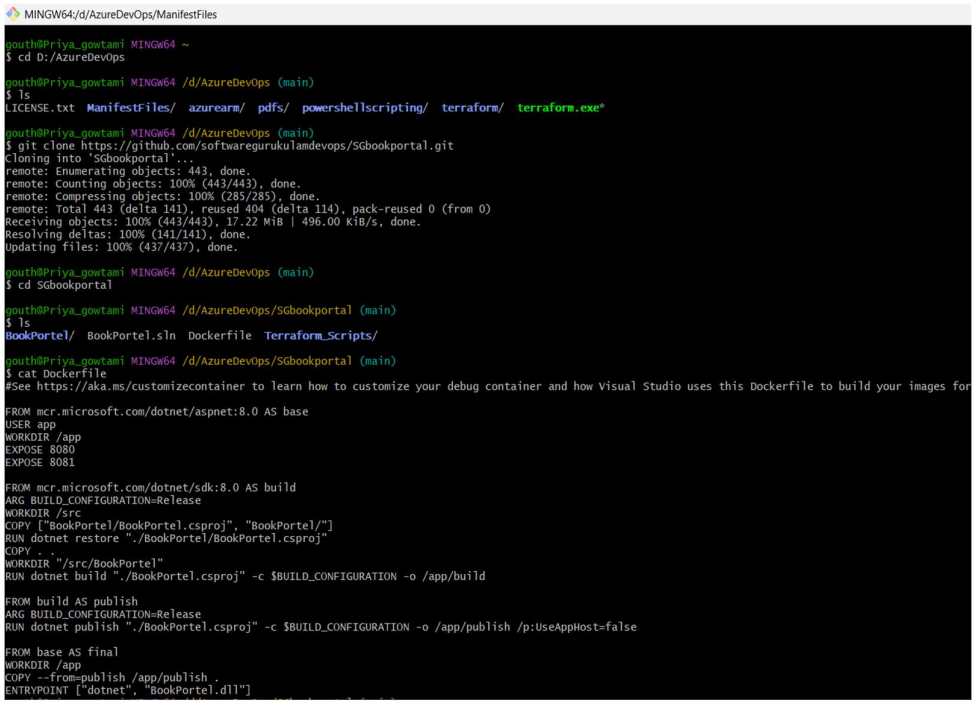
🚀 Part 1: Build and Push .NET App Docker Image

1. Clone the .NET App Repo

- git clone https://github.com/softwaregurukulamdevops/SGjobportal.git

- cd SGjobportal

2. Use the given Dockerfile as it is or create new one in root directory.



3. Build the Docker Image

```
docker build -t kubedotnet:latest .
```

- This command will build the image with the tag latest using the current directory (.) and the Dockerfile.

4. Tag the Docker Image for Docker Hub.

- Create a repo in docker hub and tag that name to the image.

```
docker tag kubedotnet:latest priya1624/priya-kube-dotnet:latest
```

- This command will create an image with the tagged repo name.

5. Login to Docker Hub

```
docker login
```

- Provide your Docker Hub username and password when prompted.

6. Push the Image to Docker Hub

```
docker push priya1624/priya-kube-dotnet:latest
```

✓ After the push completes, the image will be available in your Docker Hub repository.

MINGW64:/d/AzureDevOps/ManifestFiles

```
gouth@Priya_gowtami MINGW64 /d/AzureDevOps/SGbookportal (main)
$ docker build -t dotnetapp:latest .
[+] Building 109.8s (18/18) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 904B
 => [internal] load metadata for mcr.microsoft.com/dotnet/sdk:8.0
 => [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:8.0
 => [internal] load .dockerignore
 => => transferring context: 464B
 => [base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:8.0@sha256:d5c0d91bc8fe887684b97d5409056270ed78cd23a5123436e842a8114a64d584
 => => resolve mcr.microsoft.com/dotnet/aspnet:8.0@sha256:d5c0d91bc8fe887684b97d5409056270ed78cd23a5123436e842a8114a64d584
 => [build 1/7] FROM mcr.microsoft.com/dotnet/sdk:8.0@sha256:e6a5a8d884609907fa8d468b927df765967f6b22f890ce92bd3ae614ca4ae87e
 => => resolve mcr.microsoft.com/dotnet/sdk:8.0@sha256:e6a5a8d884609907fa8d468b927df765967f6b22f890ce92bd3ae614ca4ae87e
 => [internal] load build context
 => => transferring context: 4.39MB
 => CACHED [build 2/7] WORKDIR /src
 => [build 3/7] COPY [BookPortel/BookPortel.csproj, BookPortel/]
 => [build 4/7] RUN dotnet restore "./BookPortel/BookPortel.csproj"
 => [build 5/7] COPY . .
 => [build 6/7] WORKDIR /src/BookPortel
 => [build 7/7] RUN dotnet build "./BookPortel.csproj" -c Release -o /app/build
 => [publish 1/1] RUN dotnet publish "./BookPortel.csproj" -c Release -o /app/publish /p:UseAppHost=false
 => CACHED [base 2/2] WORKDIR /app
 => CACHED [final 1/2] WORKDIR /app
 => [final 2/2] COPY --from=publish /app/publish .
 => exporting to image
 => => exporting layers
 => => exporting manifest sha256:d174a58b8b4202adba7c8243635171227e64ccd10dc7e4f955fa3b945a15c57b
 => => exporting config sha256:ec65e892fc386c9fb0bc493bb47c75bfb94d22d0785714812cc9f2eb351bccc5
 => => exporting attestation manifest sha256:ca47481c01bac74bc880e37f79cfae2d5bf5f30eecc342f8ae8a00cf0f55d9c1
 => => exporting manifest list sha256:b4bb33e135aae187b25ed748ecad2c7b9b5e05f167384f9ae1dcd60449b3c4a3
 => => naming to docker.io/library/dotnetapp:latest
 => => unpacking to docker.io/library/dotnetapp:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/io8kvgl7yOoh7cl3mg3jdi9ej

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/SGbookportal (main)
$ docker tag dotnetapp:latest priya1624/kube-dotnet:latest

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/SGbookportal (main)
$ docker login
Authenticating with existing credentials... [Username: priya1624]

i Info → To login with a different account, run 'docker logout' followed by 'docker login'


Login Succeeded

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/SGbookportal (main)
$ docker images
REPOSITORY              TAG       IMAGE ID       CREATED              SIZE
priya1624/kube-dotnet   latest    b4bb33e135aa   About a minute ago   351MB
dotnetapp               latest    b4bb33e135aa   About a minute ago   351MB
dotnet                  latest    a4bc901f1724   19 hours ago         351MB
priya1624/todo          latest    21a0ab2880fa   3 days ago           346MB
todo                    latest    21a0ab2880fa   3 days ago           346MB
```

MINGW64:/d/AzureDevOps/ManifestFiles

```
gouth@Priya_gowtami MINGW64 /d/AzureDevOps/SGbookportal (main)
$ docker images
REPOSITORY              TAG       IMAGE ID       CREATED              SIZE
priya1624/kube-dotnet   latest    b4bb33e135aa   About a minute ago   351MB
dotnetapp               latest    b4bb33e135aa   About a minute ago   351MB
dotnet                  latest    a4bc901f1724   19 hours ago         351MB
priya1624/todo          latest    21a0ab2880fa   3 days ago           346MB
todo                    latest    21a0ab2880fa   3 days ago           346MB

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/SGbookportal (main)
$ docker push priya1624/kube-dotnet:latest
The push refers to repository [docker.io/priya1624/kube-dotnet]
9a34ec034c5e: Pushed
f8a78fa15ffa: Mounted from priya1624/priya-kube-dotnet
ef919660cf49: Mounted from priya1624/priya-kube-dotnet
4f4fb700ef54: Mounted from priya1624/priya-kube-dotnet
c7b77b706c55: Mounted from priya1624/priya-kube-dotnet
b69e1e22fe06: Mounted from priya1624/priya-kube-dotnet
fad4bc5e686a: Pushed
4e39187bfda9: Mounted from priya1624/priya-kube-dotnet
dad67da3f26b: Mounted from priya1624/priya-kube-dotnet
f20f72d364d3: Mounted from priya1624/priya-kube-dotnet
latest: digest: sha256:b4bb33e135aae187b25ed748ecad2c7b9b5e05f167384f9ae1dcd60449b3c4a3 size: 856

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/SGbookportal (main)
$ cd ..

gouth@Priya_gowtami MINGW64 /d/AzureDevOps (main)
$ cd ManifestFiles

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ kubectl create namespace gowtami-ns
Unable to connect to the server: dial tcp: lookup pocks-dns-8xt19otx.hcp.centralindia.azmk8s.io: no such host

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ kubectl create namespace gowtami-ns
namespace/gowtami-ns created

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ kubectl get namespaces
NAME                STATUS   AGE
default             Active   2d21h
gani-kube           Active   26h
gowtami-ns          Active   9s
jyoshna-meghadri    Active   20h
kube-node-lease     Active   2d21h
kube-public         Active   2d21h
kube-system         Active   2d21h
madhusrivamisett    Active   19h
poc-o1              Active   8h
priya-ns            Active   39m
reshma-node         Active   20h
testns              Active   42h
```

Part 2: Create Kubernetes Manifest Files

deploy.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dotnet-deploy
spec:
 replicas: 1
 selector:
  matchLabels:
   app: dotnetimage
 template:
  metadata:
   labels:
    app: dotnetimage
  spec:
   containers:
   - name: dotnet-deploy01
     image: priya1624/kube-dotnet:latest
     ports:
     - containerPort: 8080
```

service.yaml

```
apiVersion: v1
kind: Service
metadata:
```

```
  name: dotnet-svc
spec:
  type: LoadBalancer
  selector:
    app: dotnetimage
  ports:
    - port: 80
      targetPort: 8080
```

pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: mf-dotnet-pod
  labels:
    app: dotnetimage
spec:
  containers:
    - name: dotnet-pod01
      image: priya1624/kube-dotnet:latest
```

```
gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ vi dotnet-pod.yaml

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ vi dotnet-service.yaml

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ vi dotnet-deploy.yaml

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ cat dotnet-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: mf-dotnet-pod01
  labels:
    app: dotnetimage
spec:
  containers:
    - name: dotnet-pod01
      image: priya1624/kube-dotnet


gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ cat dotnet-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: dotnet-svc
spec:
  type: LoadBalancer
  selector:
    app: dotnetimage
  ports:
    - port: 8080
      targetPort: 8080

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ cat dotnet-deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dotnet-deploy01
spec:
  replicas: 2
  selector:
    matchLabels:
      app: dotnetimage
  template:
    metadata:
      labels:
        app: dotnetimage
    spec:
      containers:
      - name: dotnet-deploy01
        image: priya1624/kube-dotnet
        ports:
        - containerPort: 8080
```

⚙️ Part 3: Deploy to Kubernetes

1. Apply Deployment and Service

- ✓ Kubectl apply -f pod.yaml --namespace=gowtami-ns
- ✓ kubectl apply -f deploy.yaml --namespace=gowtami-ns
- ✓ kubectl apply -f service.yaml --namespace=gowtami-ns

```
gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ kubectl create namespace gowtami-ns
namespace/gowtami-ns created

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ kubectl apply -f dotnet-pod.yaml --namespace=gowtami-ns
pod/mf-dotnet-pod created

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ kubectl apply -f dotnet-service.yaml --namespace=gowtami-ns
service/dotnet-svc created

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ kubectl apply -f dotnet-deploy.yaml --namespace=gowtami-ns
deployment.apps/dotnet-deploy created
```

2. Verify Pods and Services

- ✓ kubectl get pods --namespace=gowtami-ns
- ✓ kubectl get services --namespace=gowtami-ns
- ✓ kubectl get deployments –namespace=gowtami-ns

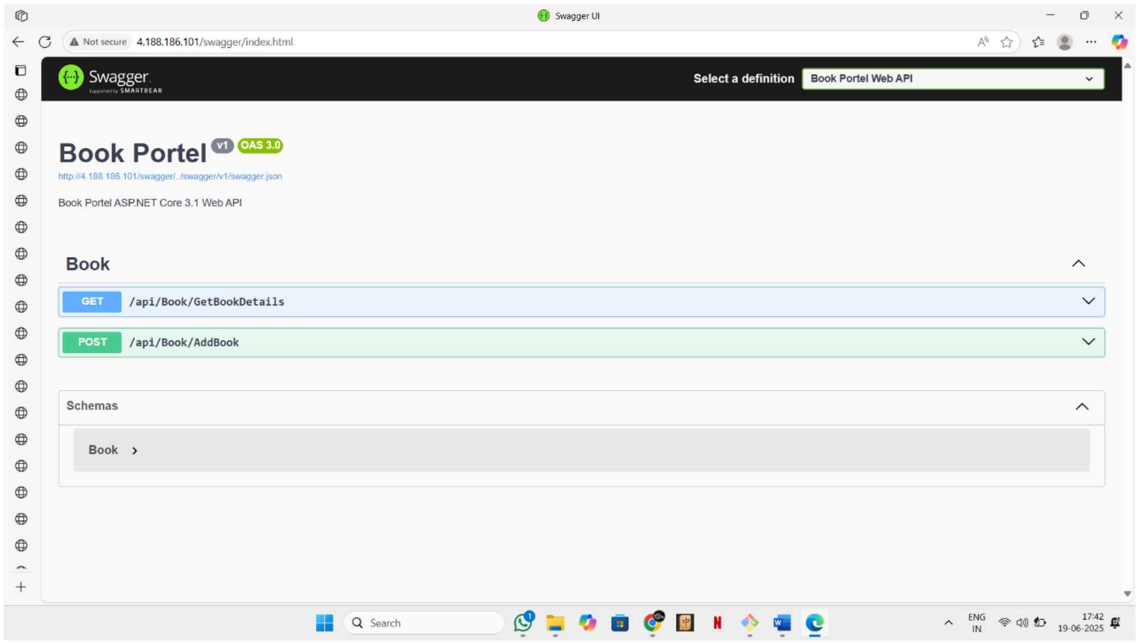
```
gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ kubectl get pod --namespace=gowtami-ns
NAME                              READY   STATUS    RESTARTS   AGE
dotnet-deploy-7747bc9686-x5txt    1/1     Running   0          21s
mf-dotnet-pod                     1/1     Running   0          46s

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ kubectl get service --namespace=gowtami-ns
NAME         TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)        AGE
dotnet-svc   LoadBalancer   10.0.132.93    4.188.186.101   80:31363/TCP   48s

gouth@Priya_gowtami MINGW64 /d/AzureDevOps/ManifestFiles (main)
$ kubectl get deployments --namespace=gowtami-ns
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
dotnet-deploy   1/1     1            1           51s
```

3. Access the Application

- ✓ If using **AKS/Cloud**, copy external IP:
- ✓ Check if the external IP is working using a web browser (i.e., .NET application is successfully deployed)



- ✓ The .NET application is successfully deployed