Process

1) Create a .net core project with the WebAPI template in VS - This creates all the required folder/file structure for the creation of the API operations logic

2) Use the DockerFile (auto generated) with the compoenents required for the basic webAPI

3) Build the docker image from the Docker File in #2 which gives the image of the WebAPI as per our requirements. We call this WebAPI-Image

Command : docker build -f .\DockerFile

4) Tag the image as per the standards. For example v1

Command : docker tag <ImageID> <TargetACR/TargetRepo:v1>

Help: ImageID can be obtained by running the command "docker image ls" - which lists the images on the local machine

Target ACR is the URL of the target azure container registry where the image is intended to be uploaded

Target Repo is the repository where the image is intended to be uploaded in the Target ACR

5) Login to Azure

Command : az login (uses the Azure CLI)

6) Login to the Target ACR

Command : az acr login -n <Target ACR>

7) Push the WebAPI-Image to ACR

Command : docker image push <TargetACR/TargetRepo:v1>

8) Use the file "webapi-deployment.yml" to create the webAPI container

Command : kubectl apply -f "webapi-deployment.yml"

9) Use the file "db-deployment.yml" to create the SQL DB container

Command : kubectl apply -f "db-deployment.yml"

10) Use the file "webapi-service.yml" and "db-service.yml" to create the webapi and DB as services that can be accessed.

Command : kubectl apply -f "webapi-service.yml"

kubectl apply -f "db-service.yml"

11) Get a list of all the pods that are currently running on the server

Command : kubectl get pods -A (lists all the pods across all the namespaces)

12) Open a bash command prompt on the webapi pod and run a few dotnet commands to compile and download the dependencies(get the pod name from the output of #11)

Similarly login to the database pod and run a few commands to build the database schema and other dependencies.

Command : kubectl exec -it <webapi pod name> --bash

On the bash command prompt (webapi) : dotnet build

On the bash command prompt (db) : sqlcmd -S <SQL Hostname (in this case- defined in the db-deployment.yml as "mymssql"> Use the password defined in the same file if prompted

On the bash command prompt (webapi) : dotnet run

13) The API is accessible at "http//localhost:80" (as defined in the webapi-deployment.yml file to send the POST request to the SQL DB