Application Deployment on local machine

**Source code**

Clone the application code. Please ignore if its already cloned.

$ git clone https://dev.azure.com/cloudiq/K8S-Dev-Workshop/\_git/k8s-dev-workshop

$ cd k8s-dev-workshop

Source code Location: src/dotnet-core/demo.cosmosdb.api/demo.cosmosdb.api

$ cd src/dotnet-core/demo.cosmosdb.api/demo.cosmosdb.api

**Build Image**

Docker is a platform for developers and sysadmins to develop, deploy, and run applications with containers. In this section we will build a docker image using a Dockerfile

* Open the terminal and change directory to source code location.
* The Dockerfile in this directory has 3 stages:

1. First stage (build) uses dotnet:2.2-sdk as base image, copies the dotnet source into the image and builds the dotnet project

FROM microsoft/dotnet:2.2-sdk AS build

WORKDIR /src

COPY . .

RUN dotnet restore "demo.cosmosdb.api.csproj"

RUN dotnet build "demo.cosmosdb.api.csproj" -c Release --no-restore

1. Second stage (publish) packs the application and its dependencies into the /app folder

FROM build AS publish

RUN dotnet publish "demo.cosmosdb.api.csproj" -c Release -o /app

1. Third stage uses dotnet:2.2-aspnetcore-runtime as base image and deploys the code by copying the /app folder from publish stage. It also exposes ports 80 and 443 for http and https connectivity.

FROM microsoft/dotnet:2.2-aspnetcore-runtime AS final

WORKDIR /app

EXPOSE 80

EXPOSE 443

COPY --from=publish /app .

ENTRYPOINT ["dotnet", "demo.cosmosdb.api.dll"]

1. Run the container using docker

Build docker image

$ docker build -t aksworkshop/dev/productsapi:v1 .

**Update Environment Variables**

* Open file src/dotnet-core/demo.cosmosdb.api/demo.cosmosdb.api/env/env.dev
* CosmosDb\_\_Account=<cosmos\_uri>
* CosmosDb\_\_Key=<cosmos\_primary\_or\_secondary\_key>
* CosmosDb\_\_DatabaseName=Training
* CosmosDb\_\_ContainerName=Products
* ApplicationInsights\_\_InstrumentationKey=<app\_insights\_key>
* Update appropriate values.

**Run the container**

$ docker run -d --name productsapi -p 3001:80 --env-file ./env/env.dev aksworkshop/dev/productsapi:v1

In the above command:

* --name productsapi --> gives name productsapi to the container
* -p 3001:80 --> binds port 80 of the container to port 3001 on the host machine
* --env-file ./env/env.dev --> get the environment variables from file .\env\env.dev
* aksworkshop/dev/productsapi:v1 --> name and tag of the image to be run

**Testing**

* Browse <http://localhost:3001/swagger>  in your favorite browser to verify the api
* Create one or two products and verify.
* Check application health in <http://localhost:3001/healthcheck/live>
* Check application dependency health (cosmosdb health) in <http://localhost:3001/healthcheck/ready>

**Exec into the running container**

$ docker exec -it productsapi /bin/bash

Run below command inside the container

$ curl http://localhost/api/products