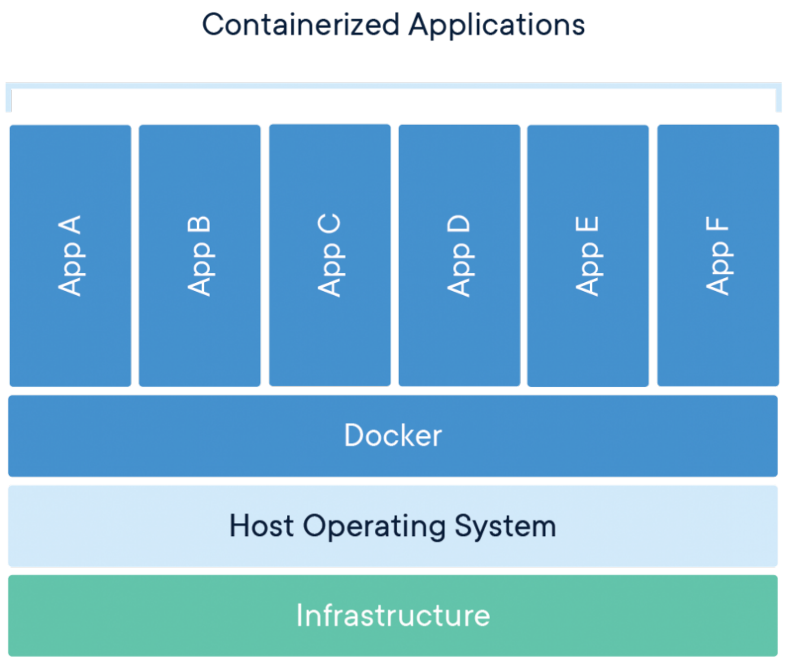
REST Web Services (Part 2)

2019/2020

Build and deploy Web services with Docker

Introduction

Containers are a technology for packaging and deploying a Web service (or other type of application), its dependencies and configuration in a portable, easily distributable image file. A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings. Containers are an abstraction at the app layer that packages code and dependencies together.



Install the Docker extension

T1 Docker support for VS Code is provided by an extension. To install the Docker extension, open the Extensions view and search for docker to filter the results. Select the Microsoft Docker extension.

Dockerfiles

With Docker, you can build images by specifying the step by step commands needed to build the image in a *Dockerfile*. A *Dockerfile* is just a text file that contains the build instructions.

T2 To generate the *Dockerfile* for your project using Visual Studio Code invoke the **Command Palette** (shift+command+p), run the **Docker: Add Docker files to Workspace** command to generate *Dockerfile* for your project type (Asp.NET Core).

The content of the generated Dockerfile is as followed:

```
FROM mcr.microsoft.com/dotnet/core/aspnet:3.0 AS base // Use Microsoft ASP.NET Core container from the Docker hub
WORKDIR /app
EXPOSE 80 // Expose ports 5000 e 5001 for the Web API traffic

FROM mcr.microsoft.com/dotnet/core/sdk:3.0 AS build
WORKDIR /src
COPY ["presencesRESTService.csproj", "./"]
RUN dotnet restore "./presencesRESTService.csproj"
COPY . .
WORKDIR /src/.
RUN dotnet build "presencesRESTService.csproj" -c Release -o /app/build

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

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish . // Copy the published code from the "/app/publish" folder to the root folder in the container
ENTRYPOINT ["dotnet", "presencesRESTService.dll"] // Run the dotnet application from within the container
```

T3 Build and run your project docker image. Open the integrated terminal of VS Code and navigate to your project folder. Use the following commands to build and run your Docker image:

First to build the container:

>> docker build -t presencesrestservice . (builds the image with the context of the current directory)

>> docker run -d -p 8080:80 presencesrestservice

T4 Diagnose Docker using the following commands:

>> docker ps

The docker ps command only shows running containers by default. To see all containers, use the -a (or --all) flag.

>> docker logs [container ID]

The docker logs command fetches the logs of a container present at the time of execution.