Build an ASP.NET Core Service, and App with Core 2.2 Two-Day Hand-On Lab

Lab 0

Welcome to the Build an ASP.NET Core Service and App with EF Core Two-Day Hands-On Lab. Prior to starting the rest of the workshop, you must have .NET Core 2.2+ SDK, .NET Core 2.2+ runtime, Docker Community (or SQL Server 2017 Developer), and an appropriate IDE installed. Supported IDEs includes:

- Visual Studio 2017 15.9+,
- Visual Studio 2019
- Visual Studio for the Mac 7.7+
- Visual Studio Code 1.31.1+

SQL Server Management Studio is recommended.

Part 0: Permissions

You must have admin permissions on your machine to complete this hands-on lab.

Part 1: Installing the Prerequisites

Step 1: Install/Confirm .NET Core Runtime and SDK

1) Download and install the latest 2.2 .NET Core SDK and Runtime from http://dot.net.
    a) For Visual Studio 2017 (15.9+) download SDK 2.2.107
    b) For Visual Studio 2019 (16.0.X) download SDK 2.2.204
    c) For Visual Studio 2019 (16.1.X) download SDK 2.2.300
    d) For Visual Studio for the Mac download SDK 2.2.107
    e) For Visual Studio Code (or any other IDE) download SDK 2.2.300
2) Open a command window and type:

```
where dotnet
```

3) After unpacking some files, it should respond with:

```
C:\Program Files\dotnet\dotnet.exe
```

4) You might also see the following (not required for this course):

```
C:\Program Files (x86)\dotnet\dotnet.exe
```

5) Check the version of the .NET Core Runtime by entering:

```
dotnet --list-runtimes
```

6) The response will be (at the time of this writing):

```
Microsoft.AspNetCore.All 2.2.5 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.All]
Microsoft.AspNetCore.App 2.2.5 [C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App]
Microsoft.NETCore.App 2.2.5 [C:\Program Files\dotnet\shared\Microsoft.NETCore.App]
```

   a) The Microsoft.AspNetCore.All metapackage leverages the .NET Runtime store to trim down the portable deployment size of applications. It contains most everything needed for ASP.NET Core 2.0x applications (including Entity Framework Core). Note: It is for versions 2.0 (not recommended for 2.1+, which uses Microsoft.AspNetCore.App).

   b) Microsoft.AspNetCore.App leverages the ASP.NET Core shared framework. Any assets in the ASP.NET Core shared framework will not be deployed with your app, and are pre-compiled for better application startup time. Microsoft.AspNetCore.App also uses version roll-forward to work with later versions of the 2.x framework installed on the target machine.

7) Check the version of the .NET Core SDK by entering:

```
dotnet --list-sdks
```

8) The response will be (at the time of this writing):

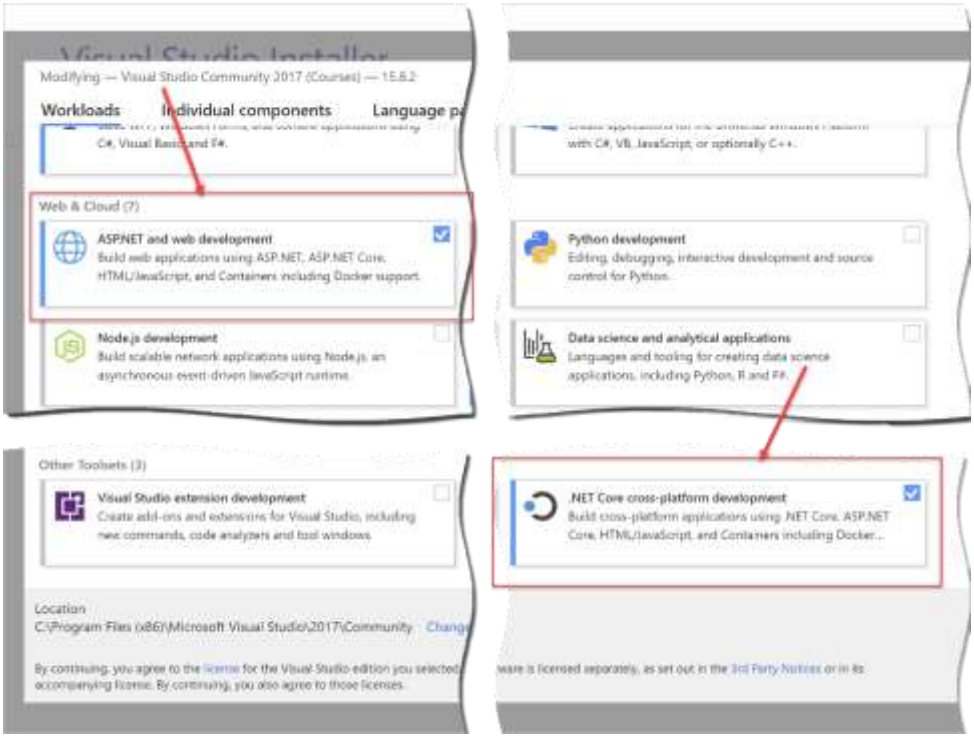
```
2.2.300 [C:\Program Files\dotnet\sdk]
```

Step 2: Install an IDE

The HOL will work with Visual Studio 2017, Visual Studio 2019 (preview), Visual Studio for the Mac, or Visual Studio Code. JetBrains Rider can also be used.

Option 1: Download and install Visual Studio 2017 15.9+

1) Download Visual Studio 2017 15.9+ (any edition) from the Visual Studio home page: https://www.visualstudio.com/en-us/visual-studio-homepage-vs.aspx
   a) The Community Edition is free, and has everything you need to complete this Hands-On Lab
2) Start the installer
   a) The new installation experience has separate workloads based on what type of work you intend to do. For this lab, select the "**ASP.NET and web development**" workload as well as the "**.NET Core cross-platform development**" workloads.



Option 2: Download and install Visual Studio 2019

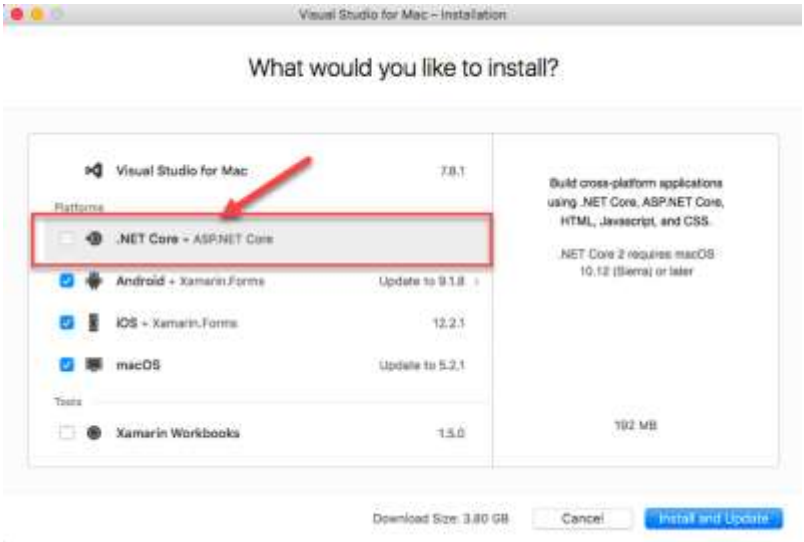
1) Download Visual Studio 2019 Preview from https://visualstudio.microsoft.com/. Just like VS 2017, the features are broken into workloads. Select the same workloads for 2019 as shown in Figure 1-1: "ASP.NET and web development" and the ".NET Core cross-platform development".

Option 3: Download and install Visual Studio Code 1.31.1+

1) Download Visual Studio Code from https://visualstudio.microsoft.com/.
2) Install the "Microsoft C# extension (powered by OmniSharp)" extension.
3) Install the "C# IDE Extension for Visual Studio Code" extension https://marketplace.visualstudio.com/items?itemName=jchannon.csharpextensions

Option 4: Download and install Visual Studio for the Mac

1) Download Visual Studio for the Mac from https://visualstudio.microsoft.com/
2) Select .NET Core from the install screen (image from 7.8.1)



Step 2: Download/Install SQL Server Management Studio (SSMS)

This is not required for the workshop, but makes it easier to work with the database.

1) Download and install SQL Server Management Studio 17.9.1+ from https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017
    a) This is a free tool from Microsoft

Step 3: Install Docker Community

Docker is a containerization platform that runs on Windows, MacOS, and Linux.

1) Download and install Docker Community from https://www.docker.com/get-started
    a) When installing, select Linux containers (and not Windows containers)

b) This is a free tool but requires you to have a Docker user id and password

2) [Windows] Download and Install Kitematic for Docker by right clicking on the Docker ship in the system tray, click Kitematic, and follow the instructions. Kitematic is a GUI for managing Docker images and containers.

Step 4: Pull the SQL Server Image and Create Local Container

A Docker image is like a class definition, while a Docker Container is like an instance of that class. To run SQL Server in Docker, you must first pull the image from Docker Hub, and then create a container using that image.

1) Pull the SQL Server 2017 for Linux (Ubuntu 16.04) image. Enter the following command:

```
docker pull mcr.microsoft.com/mssql/server:2017-latest
```

2) When creating an image, there are two required environment variables, "ACCEPT_EULA" and "SA_PASSWORD". An optional environment variable "MSSQL_PID" sets the product version. The host port mapping to the image port needs to be set, and a friendly name added. Create the container using the following command:

a) **NOTE:** On Windows, use double quotes ("). On Mac and Linux, use single quotes (').

```
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=P@ssw0rd" -e "MSSQL_PID=Express" -p 6433:1433 --name
SpyStoreHol -d mcr.microsoft.com/mssql/server:2017-latest
```

Step 5: [Optional] Download and install SQL Server 2017 Developer

1) Download the SQL Server 2017 Developer Edition from https://go.microsoft.com/fwlink/?linkid=853016

2) NOTE: If you choose to use SQL Server 2017 Express, you will need to update the connection strings in the subsequent labs (this is covered later) and install the latest Cumulative Update located here:

```
https://www.microsoft.com/en-us/download/details.aspx?id=56128
```

Part 2: Clone (or Download) the Lab Repo

All of the lab documents and asset files are located here:
https://github.com/skimedic/dotnetcore_hol/tree/master/TwoDay/2.2

Summary

These are all of the tools you need to complete this Hands-on Lab.