**Dockerize an ASP.NET Core application**

This example demonstrates how to dockerize an ASP.NET Core application.

Prerequisites

This example assumes you already have an ASP.NET Core app on your machine. And need windows docker desktop application.

Create a Dockerfile for an ASP.NET Core application:

1. Go to your application, Project > Add > Docker Support (Fig: 1)
2. Then Select Linux. (Because your docker desktop is run as like it will be on linux machine)
3. Your created docker file is look like as Fig: 2
4. Now you can run your project as “Docker” by click run button.
5. Project will be run on docker. Docker Desktop automatically run by system and you will see the container on it.

**Why we use docker in ASP.net project?**

1. Pre-Made Runtime Environment
2. Version controlled infrastructure.
3. Runtime Consistency
4. Securable like VM
5. Lighter then VM
6. Cloud Friendly
7. Continuous Integration/Deploy
8. Docker in particular.

In details please read the blog <https://stackify.com/10-reasons-to-use-docker-with-asp-net/>

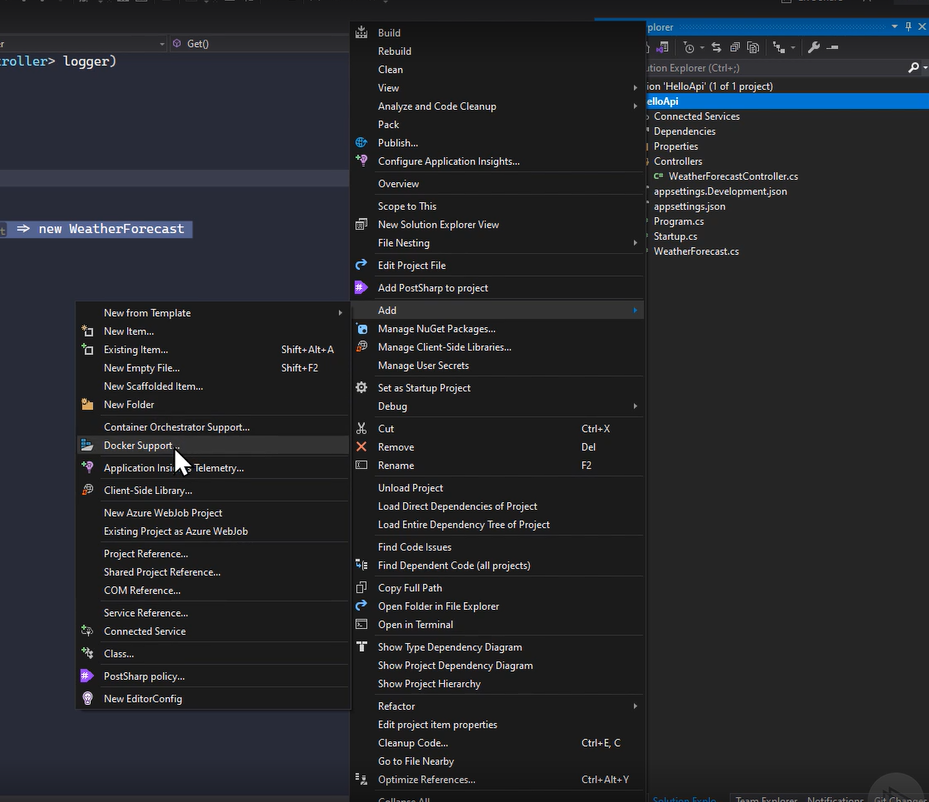


Fig: 1

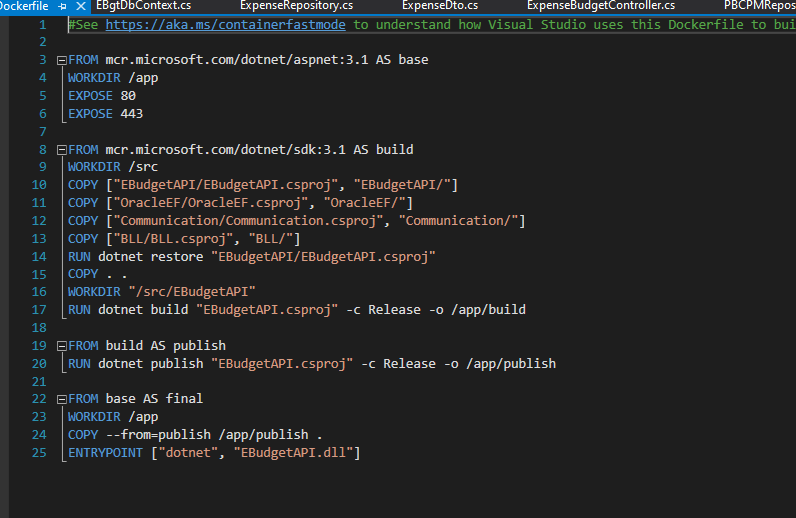


Fig: 2

For more: <https://www.c-sharpcorner.com/article/building-and-running-asp-net-core-application-in-docker-container/>

**Build a Secure Node.js App with SQL Server**

## What is Node.js?

Node.js is neither a framework nor a library but an **open-source server environment**. Being based on Chrome’s V8 JavaScript engine, Node.js is designed to build scalable network applications.

Node.js was launched in 2009 and has quickly gained popularity as the most widely used non-language, non-database development tool due to its simplicity, speed, and multitasking abilities.

JavaScript developers who want to take their coding to the next level love Node.js and the people who are hiring these developers take advantage of fast, real-time, event-driven servers.

## What Node.js is Used For?

The benefits of Node.js make it a good option for different types of applications. Here are the most common:

* Real-time Web Applications
* Single-page Application
* Streaming Applications
* Internet of Things

**Pros of Using Node.js :**

- Node.js has been regarded as a full-stack JavaScript for serving both the client and the server-side applications.

- Node.js interprets the JavaScript code via Google’s V8 JavaScript engine. It compiles JavaScript code into the machine code(directly). This makes it easier and faster to implement the code.

- Speed of the code execution also enhanced by the runtime environment as it supports the non-blocking I/O operations.

- Latest versions of Node.js are strongly overlapping with the development of V8. With V8, JavaScript code is converted to byte code for use in a VMs.

### Pros of Using .NET Core

- The most important benefit of the .NET Core is its high performance. With recent updates, the code gets much more optimized which improves performance at the very end.

- .NET Core demands less coding now, developers can easily optimize the code structure by means of writing much lesser statements. That ends up with less time spent on development and smaller budgets which puts .NET Core very close to Node.js in terms of development speed.

- Maintaining a large .Net Core application is much more easier comparing to Node.js.  
- Platform agnostic nature of the .NET Core makes it easy to create autonomous, self-sufficient, and microservice applications.

- .NET Core is easier to get it to work and work WELL. It puts you, as a developer, into certain limits which are very precise, yet very flexible when you approach things in a right way.

## Advantages of Node.js - Why Use Node.js For Web App Development:

