

# Creating and Deploying Core Web App in Docker

## Introduction:

In this project, we will create an ASP.NET Core Web App MVC project with Docker support, enabling easier deployment and management of your application.

### Step 1: Create a Visual Studio Core Web App MVC Project

1. Open Visual Studio.
2. Click on "Create a new project."
3. In the "Create a new project" dialog, search for "ASP.NET Core Web App (Model-View-Controller)".
4. Select the template and click "Next."
5. Configure your project settings and click "Create."
6. Choose your preferred authentication method, or select "No Authentication" for simplicity.
7. Click "Create."

### Step 2: Configure Docker

1. During project creation, enable Docker support by checking the "Configure for Docker" checkbox.
2. Choose the operating system you want to target (e.g., Windows or Linux). The choice depends on your environment and requirements.

### Step 3: Create a User Class

In the "Models" folder of your project, create a "User" class with properties that represent user data.

### Step 4: Create Controllers

Create two controllers: one for the "Account" functionality and one for the "Dashboard."

- Right-click on the "Controllers" folder.
- Select "Add" > "Controller."
- Choose the "MVC Controller - Empty" template.
- Name the controllers (e.g., "AccountController" and "DashboardController").

### **Step 5: Create Views**

For each controller, create views that represent the user interface. You can use Visual Studio's scaffolding feature to generate the views automatically based on your controller actions.

### **Step 6: Build the Docker Image**

Right click on the project and add docker support

### **Step 8: Access the Application**

Open a web browser and navigate to `http://localhost:32769/` to access your ASP.NET Core MVC application running inside a Docker container.

### **Conclusion:**

By configuring Docker, building a Docker image, and running your application in a container, you can streamline the deployment process and make your ASP.NET Core MVC application more portable and scalable. This approach empowers you to develop and deploy web applications efficiently in various environments.

GitHub: <https://github.com/NakkaAnilkumar/section7-1.20>