Containerize .NET core web API using Docker

# Prerequisites

* [Docker Desktop](https://hub.docker.com/editions/community/docker-ce-desktop-windows) for windows

# Visual Studio Code (Create Web API)

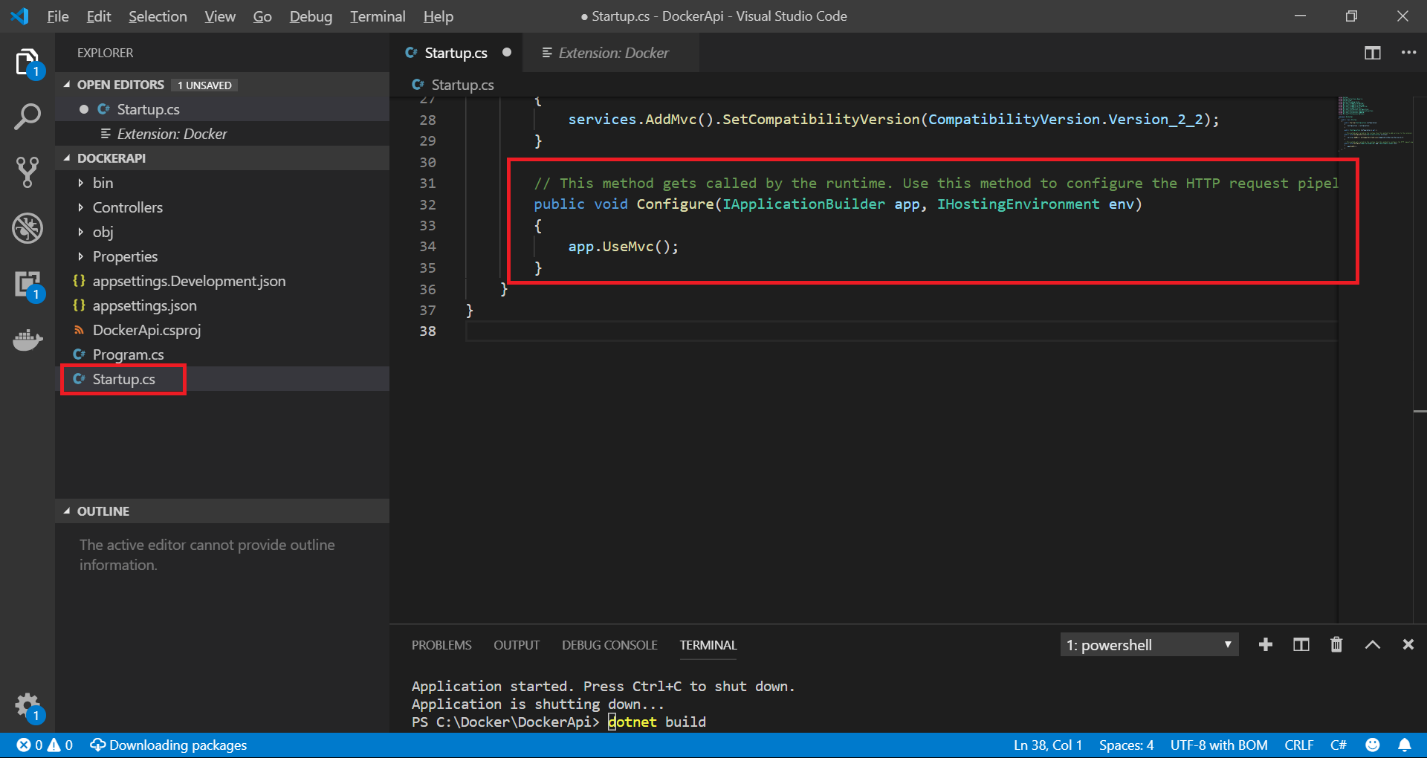
1. Open Visual Studio Code - Terminal - Create Web API Template

**dotnet new webapi -n DockerApi**

1. Open web API Solution

**code -r DockerApi**

***Note****: Remove code from Startup.cs - Configure method*

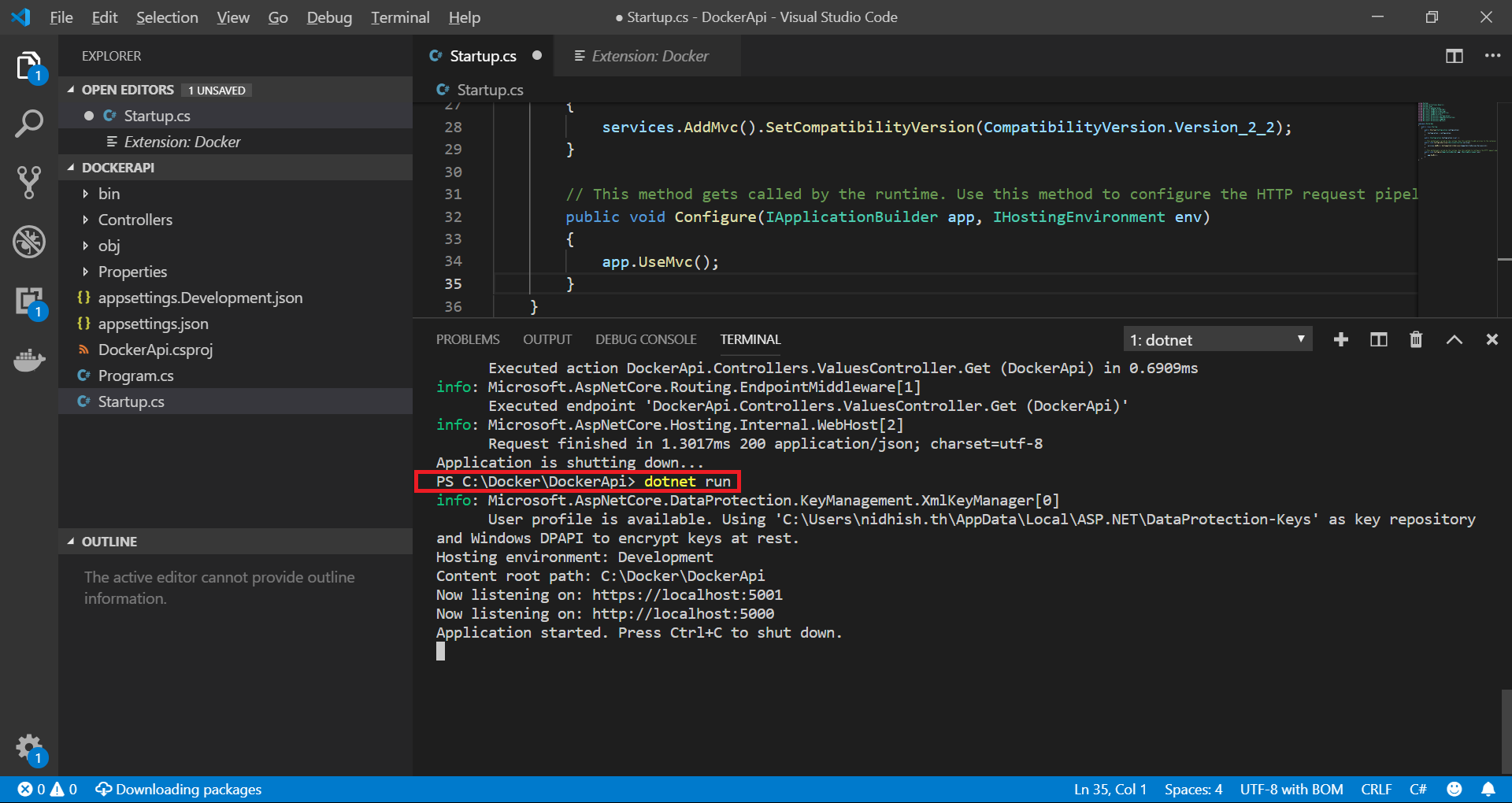


1. Build Web API Solution

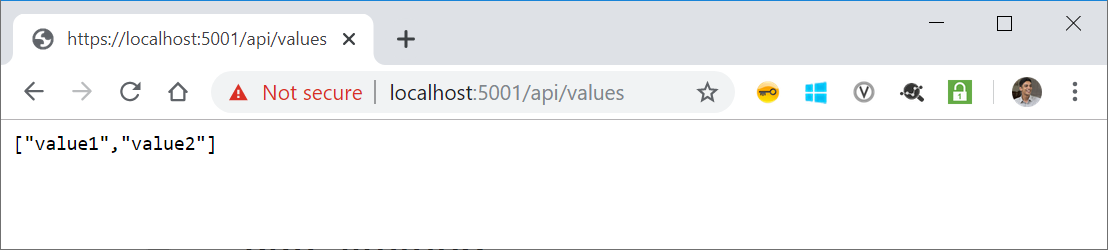
**dotnet build**

1. Run Web API Solution

dotnet run

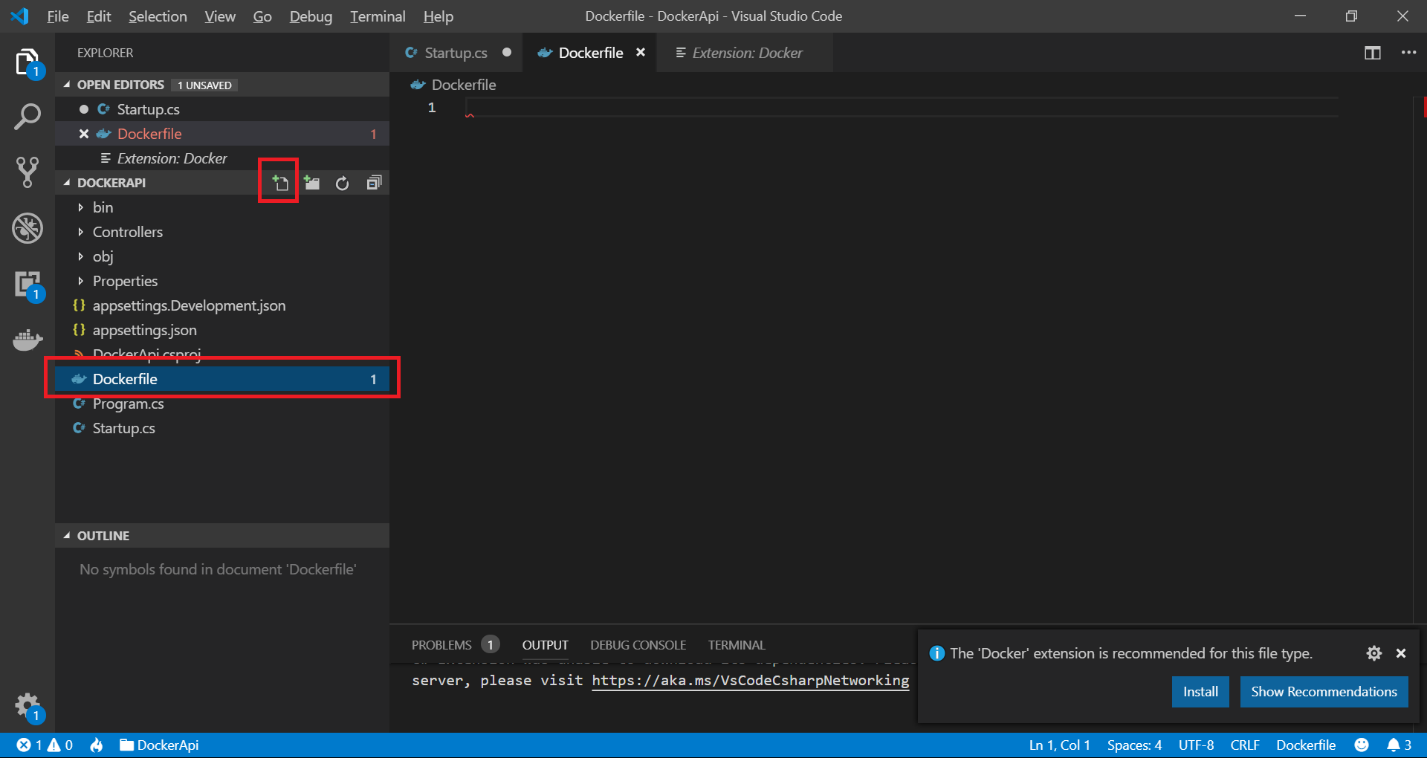


1. Test Web API



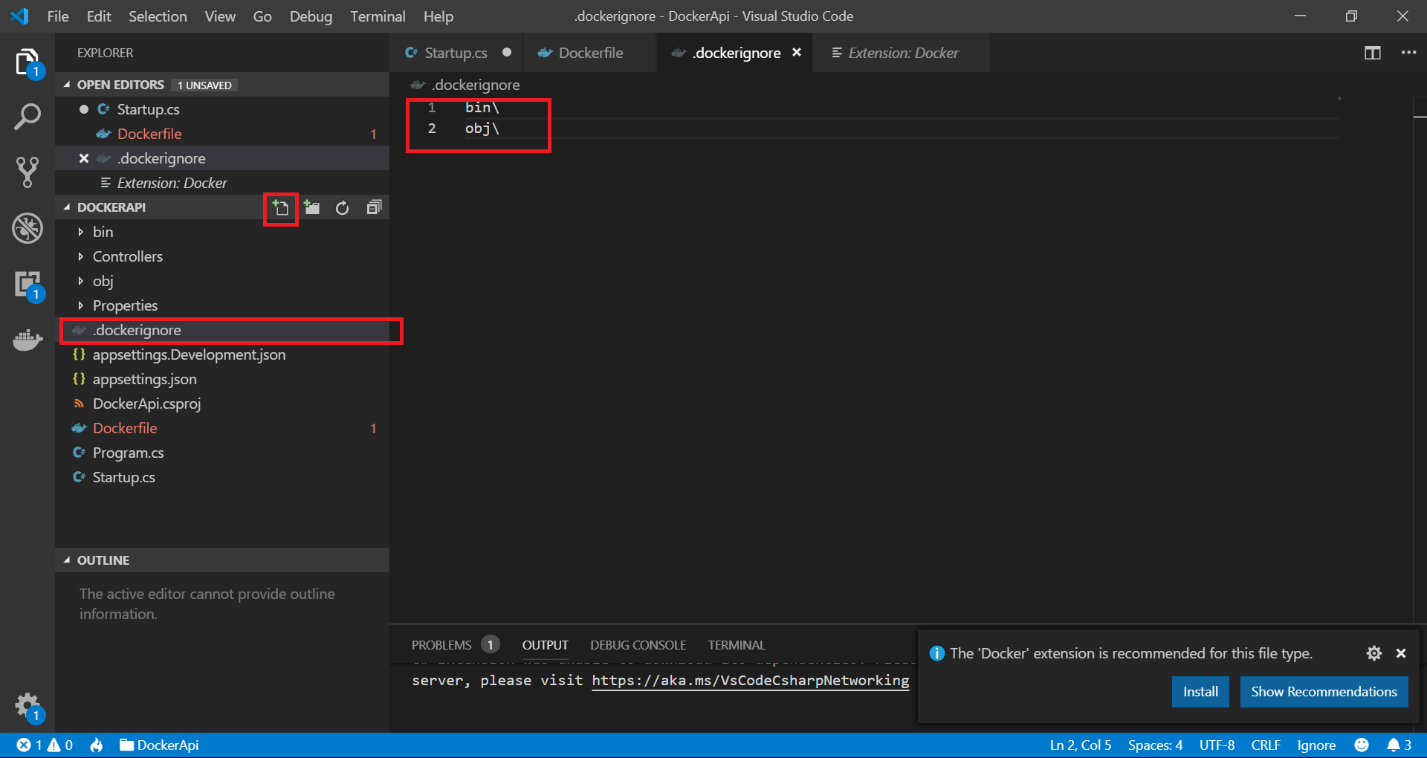
# Add Docker file to Web API Project

1. Add DockerFile to the Visual Studio Code solution



**Note**: Adding Docker file will prompt to install Docker extension

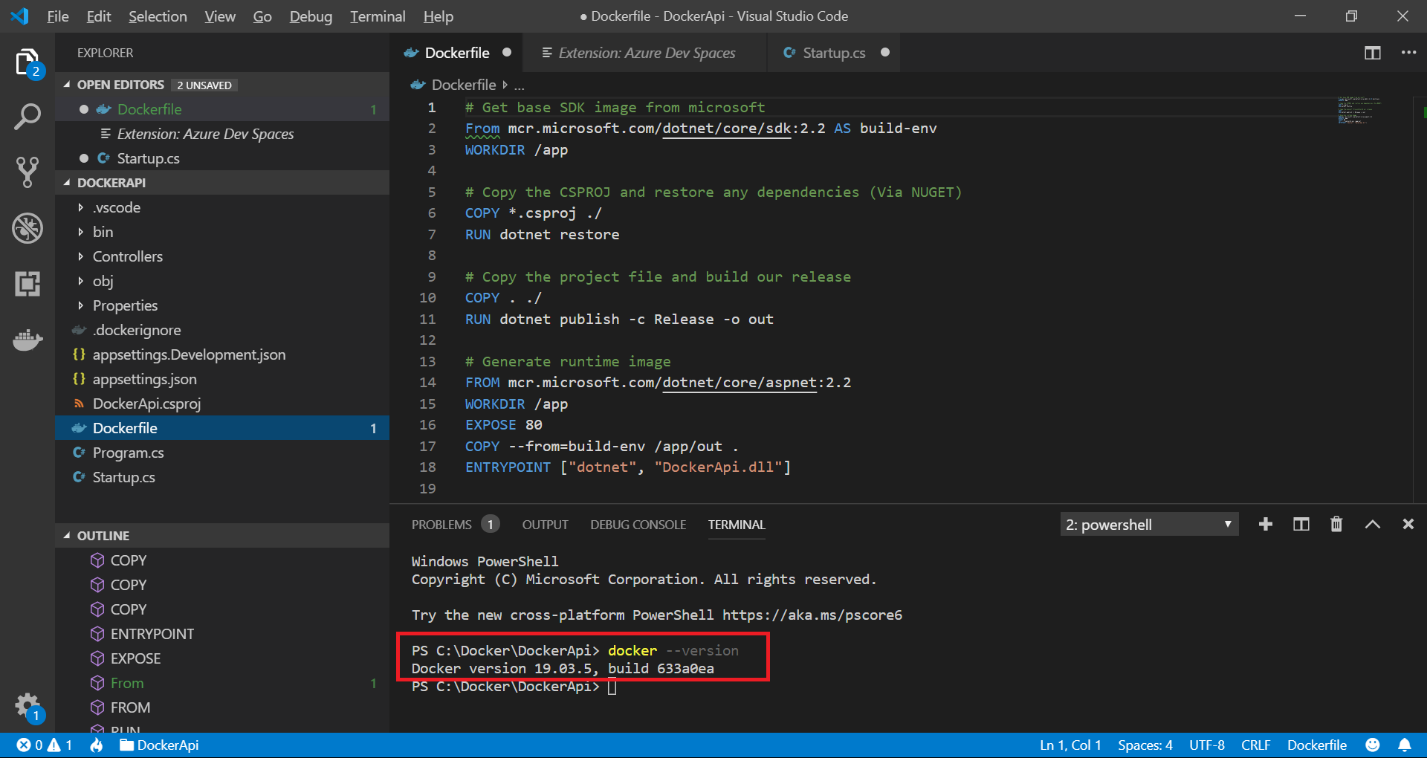
1. Add .dockerignore file to the Visual Studio Code Solution



# Run through Docker Engine

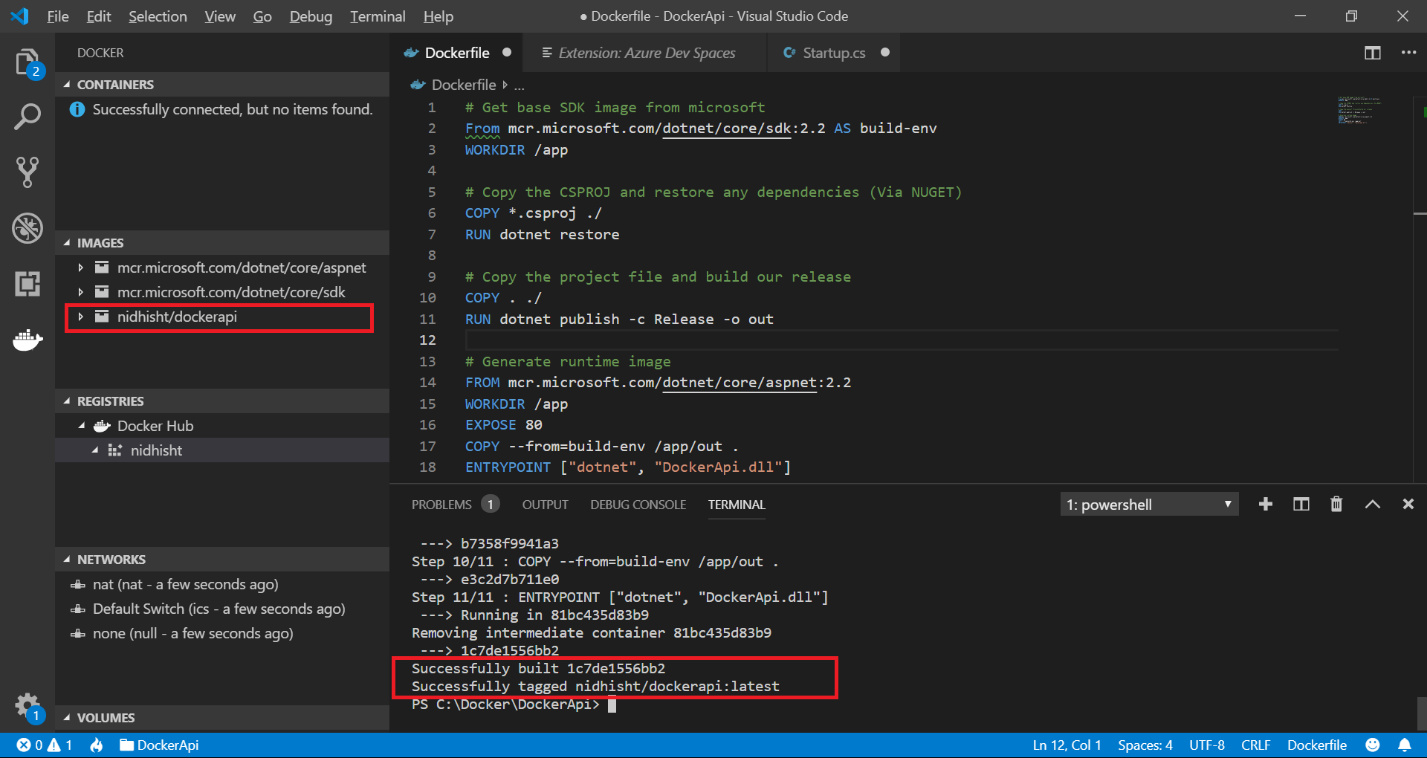
1. Check if docker is installed & running

docker –version



1. Initiate docker build & tag

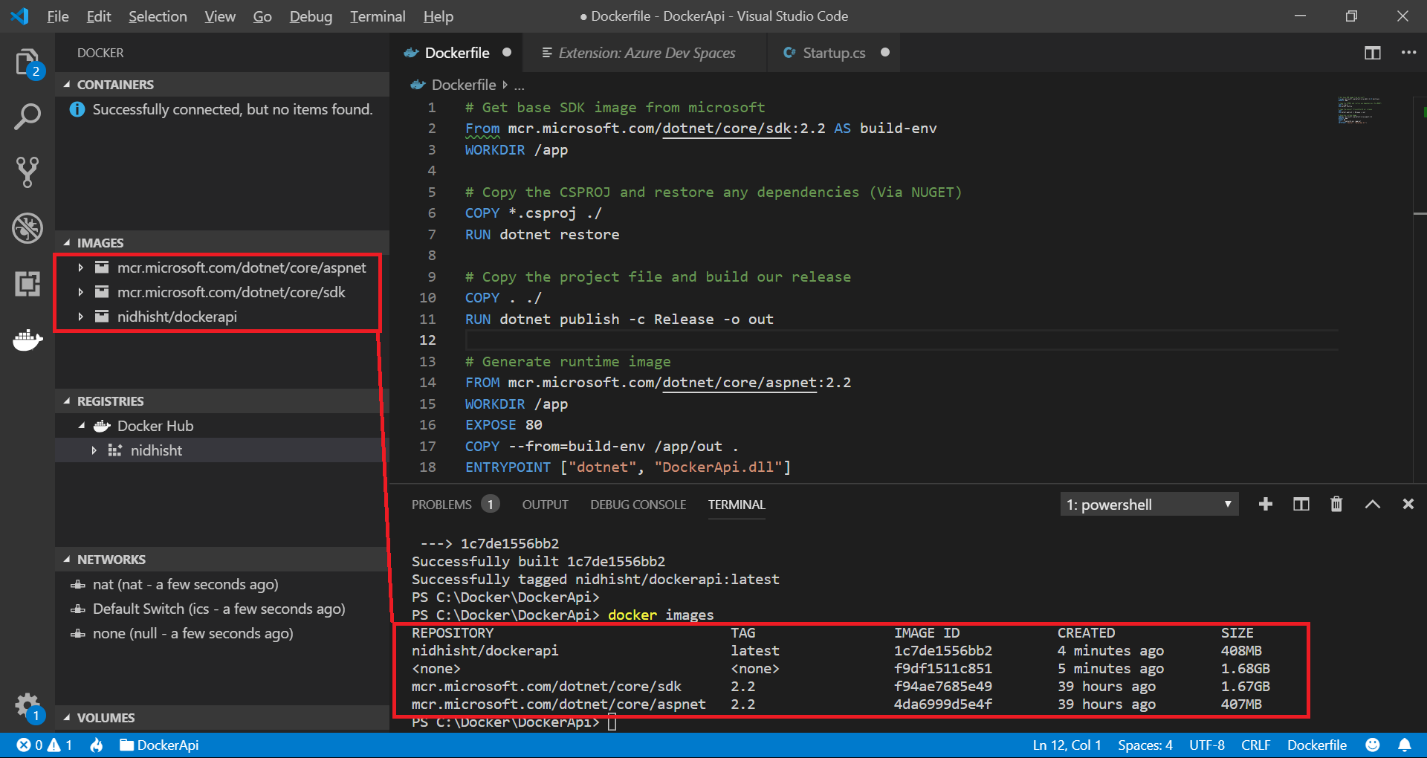
docker build -t nidhisht/dockerapi .



**Note**: repository name should be in smaller case

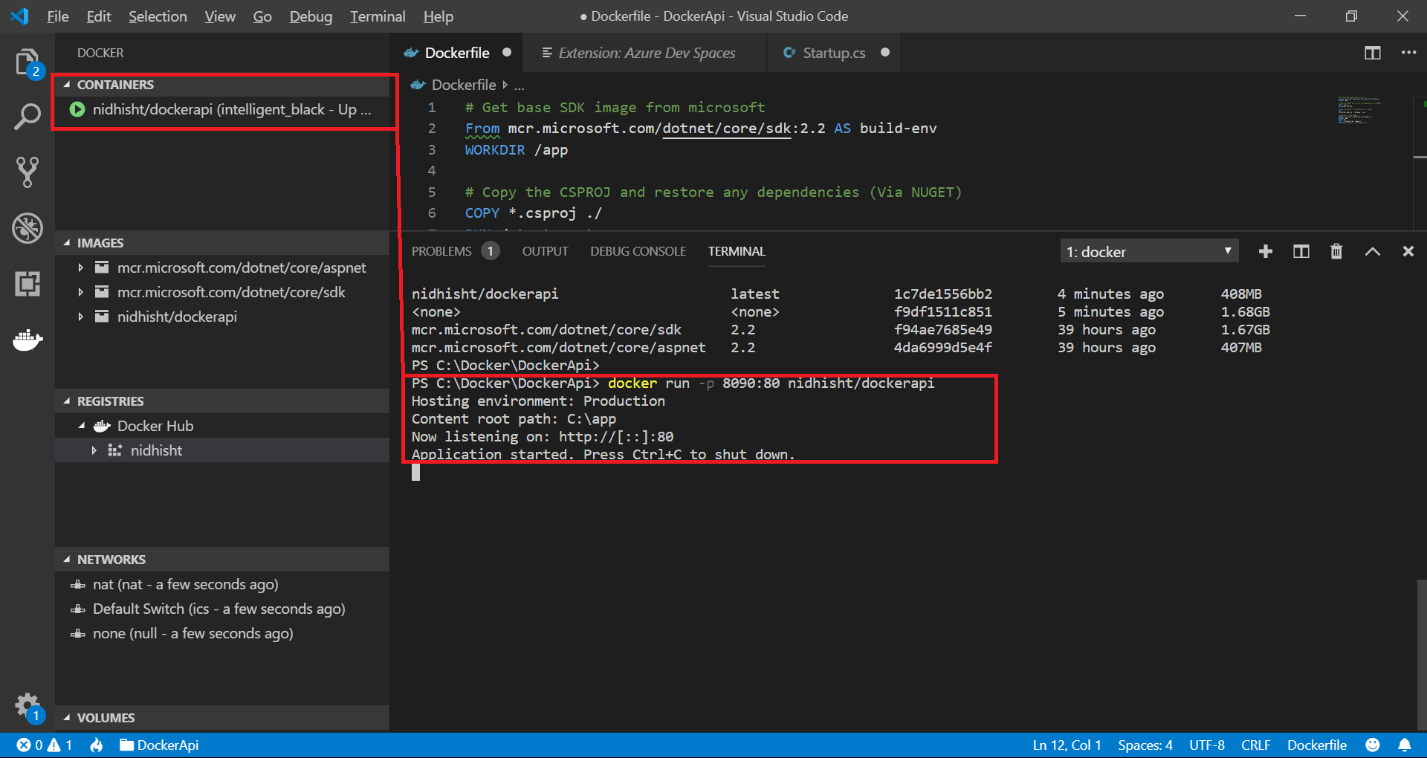
1. Fetch list of images

docker images

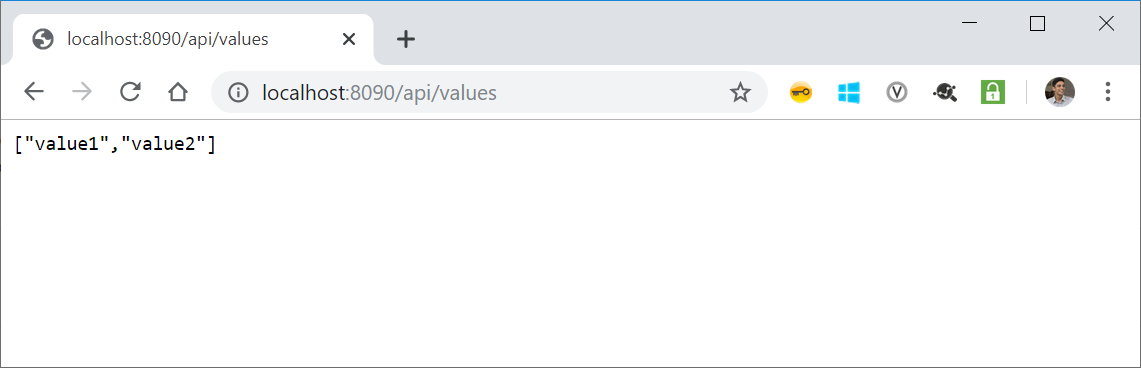


1. Run docker

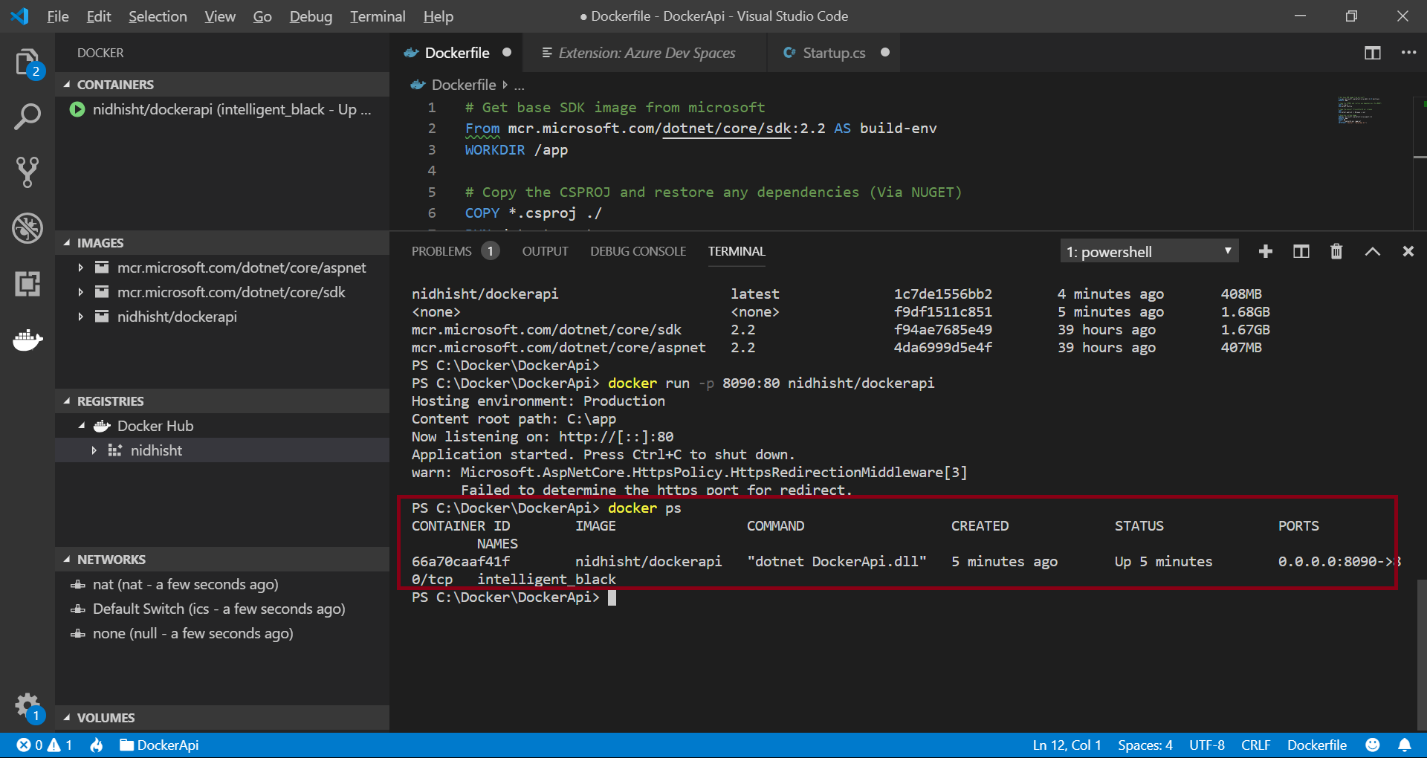
docker run -p 8090:80 nidhisht/dockerapi



http://localhost:8090/api/values



1. Get list of running containers

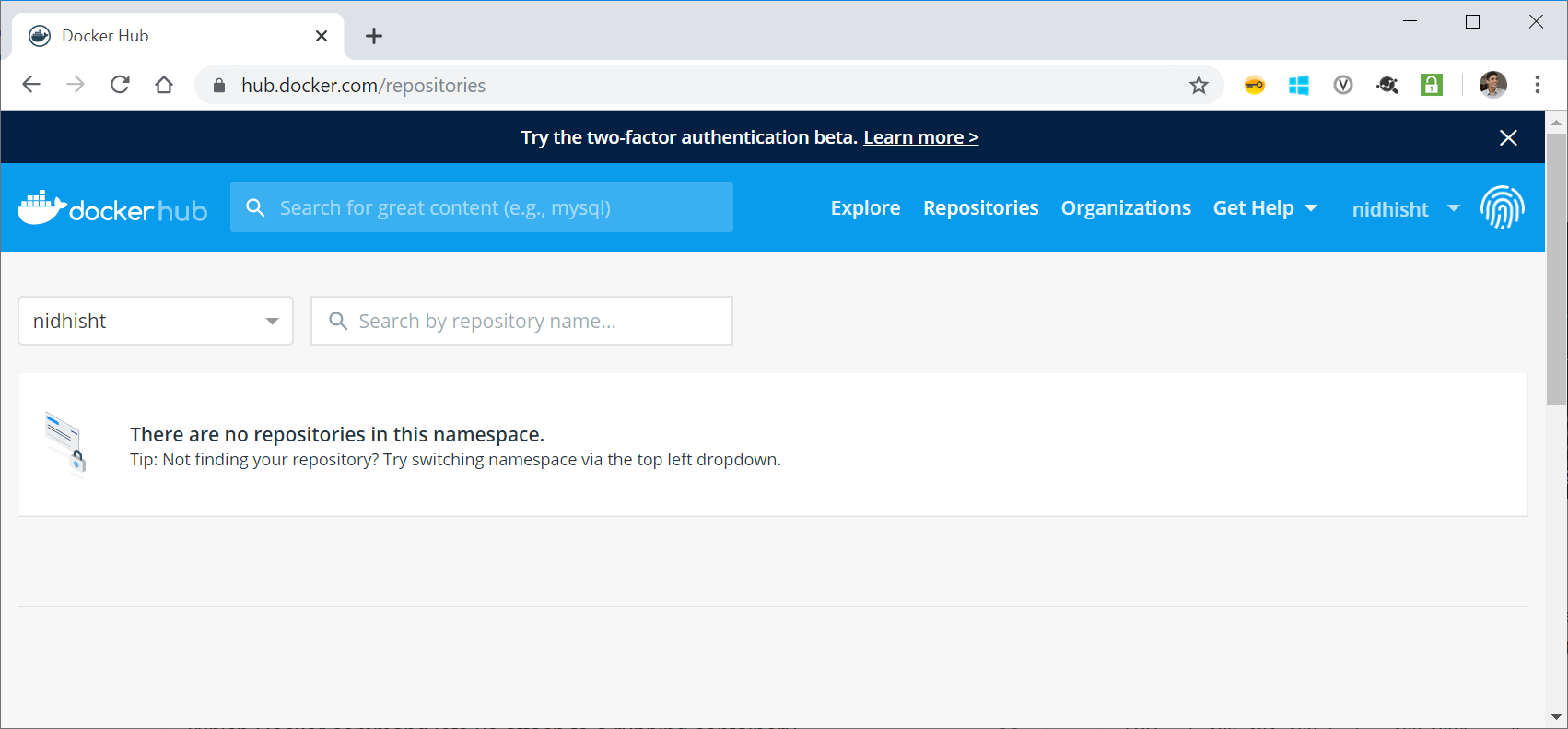


# Push image from localhost to Docker Hub

1. Login to docker

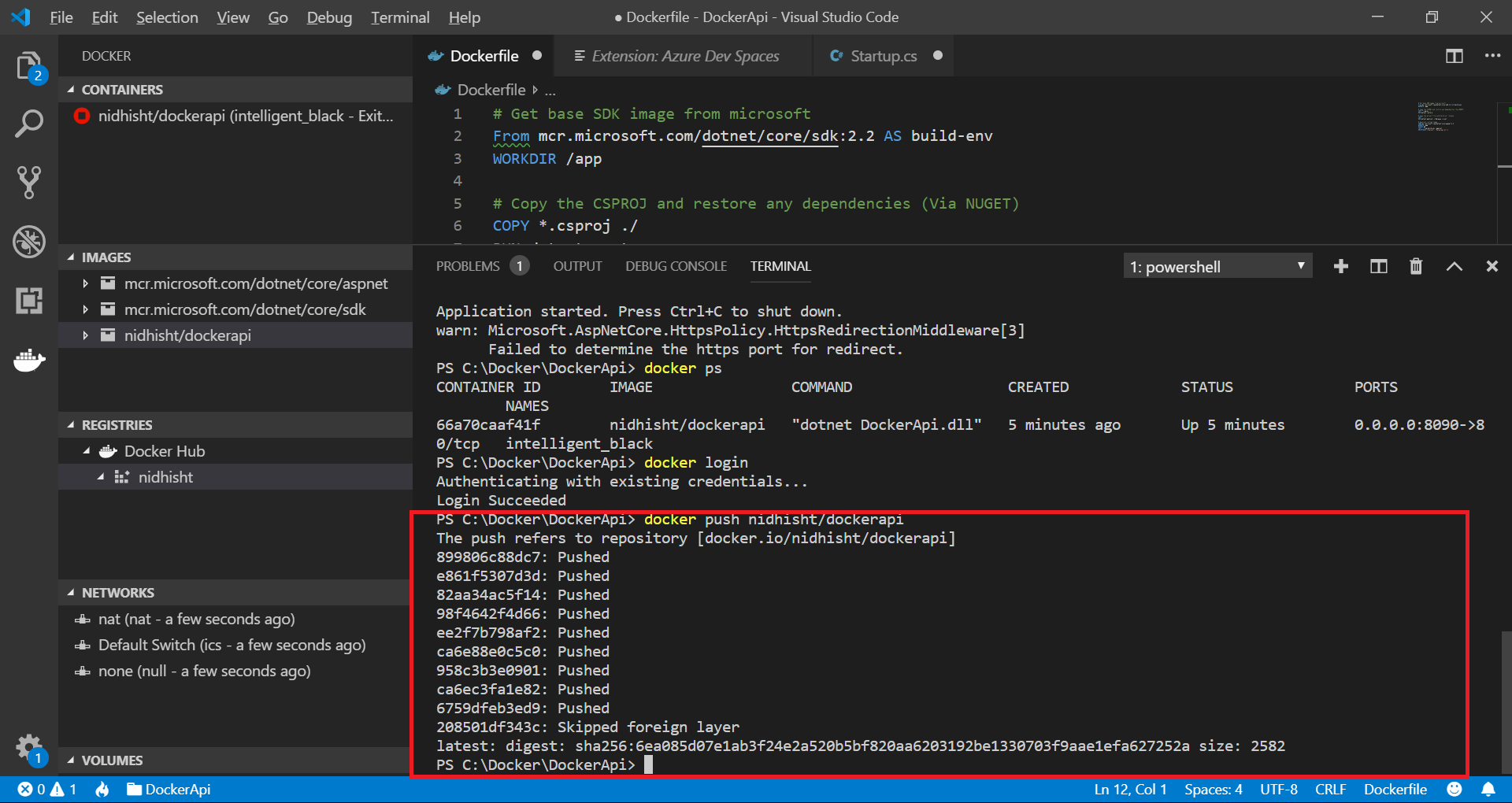
docker login

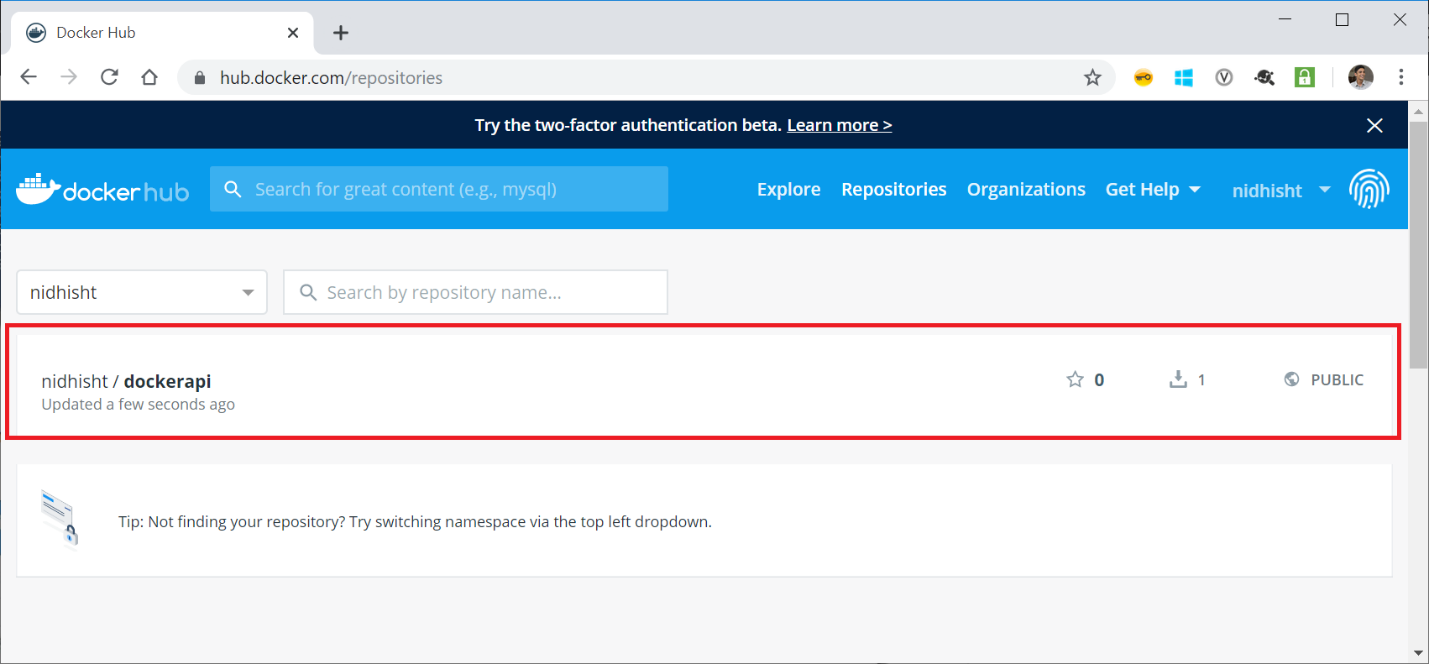
1. Check for existing repositories in Docker Hub



1. Push images to Docker Hub (NOTE: This takes time to complete. Depends on network speed)

docker push nidhisht/dockerapi





# Reference:

<https://dotnetplaybook.com/deploy-a-net-core-api-with-docker/>

Docker file

