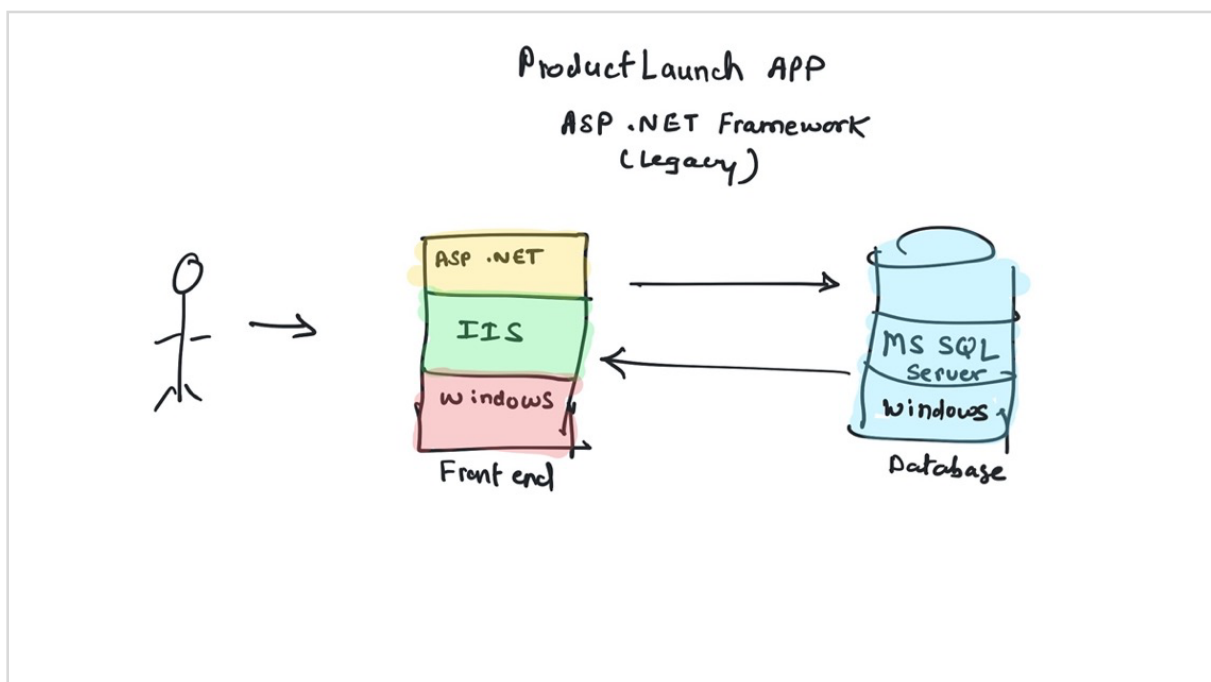


Nano Project 301 - (Docker image Build) - Dockerize a .NET Framework Application

You have learnt how to build docker images, how to write a Dockerfile and also publishing it to the registry such as Docker Hub. Its now time for you to get into a Azure Devops Engineer's hat and start containerising any .NET application like a ninja. Here is your task for today,

Scenario

School of Devops is launching a new Devops Mastery course. They have a legacy ASP .NET framework application which they have been using for all course launches. You as a Devops Consultant are in charge of containerising this application and prepare it to be deployed to Azure Kubernetes Service later. As the preparatory stage, you need to write a Dockerfile, build and test the application locally, and then publish the image to Docker Hub .



Here goes the application code [GitHub - initcron/productlaunch: ASP.net Framework \(Legacy\) Sample App](#)

Steps you would need to perform

1. In the first stage, pull an image which contains the build tools for .NET Framework

2. Launch a dev environment using the image pulled above
3. Copy over the source code to this dev/build environment
4. Build the application using `nuget restore` and then with `msbuild`. This will build the release artifact.
5. In the second stage, launch a container with only ASP .NET runtime. This would contain IIS server packed in.
6. Copy over the artifact generated above to `/inetpub/wwwroot` path. It would automatically be picked up by IIS and served when launched.
7. Build the image using `docker build` command.
8. To test the application completely, you would have to launch a SQL Server container as well, the steps for which are given below. Launch it and find out the ip address of it.
9. Update connection string in `ProductLaunch.web\Web.config` with the correct db hostname/ip and password.
10. Launch container using the image that you just built, map container's port 80 to the host, and see if the application loads up in the browser. If you see the Product Launch page, don't forget to dance with joy.
11. Publish the image to the registry.

Launching SQL Server in a Container

You need not build an image for SQL Server as its a COTS (Commercial Off The Shelf) application and an image for it is already available in the registry. All you need to do is launch it and then connect to it from your application using correct connection string.

Here is the command to launch SQL Server,

```
docker run --detach `
  --publish 1433:1433 `
  --env sa_password=super!secRet `
  --env ACCEPT_EULA=Y `
  --name mssql `
  microsoft/mssql-server-windows-express
```

You could base your work on the Dockerfile provided here as a sample
[dotnet-framework-docker/Dockerfile at master · microsoft/dotnet-framework-docker · GitHub](#)

#docker/windows/labs

#docker/windows/project/dockerfile