[**1. SECTION – 1 : What is Kubernetes** 1](#_Toc172561679)

[**2. SECTION – 2 : Kubernetes Setup** 3](#_Toc172561680)

[**3. SECTION – 3 : Create Kubernetes Pods** 5](#_Toc172561681)

[**4. SECTION – 4 : Create Kubernetes Service** 6](#_Toc172561682)

[**5. SECTION – 5 : Create Kubernetes Namespace** 8](#_Toc172561683)

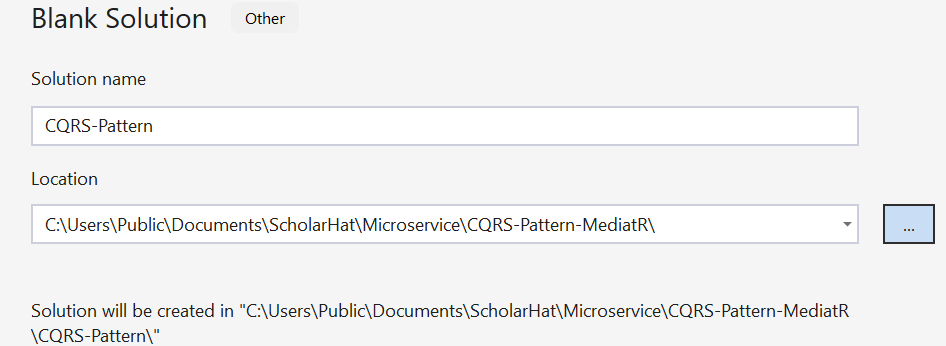
[**6. SECTION – 6 : Create Kubernetes Deployment** 11](#_Toc172561684)

[**7. SECTION – 7 : Create Docker Webapp and SQL Server for Kubernetes** 12](#_Toc172561685)

[**8. SECTION – 8 : Create Replica of Pods and Container (Self Healing and Load Balancing)** 17](#_Toc172561686)

# **1. SECTION – 1 : CQRS Project Setup**

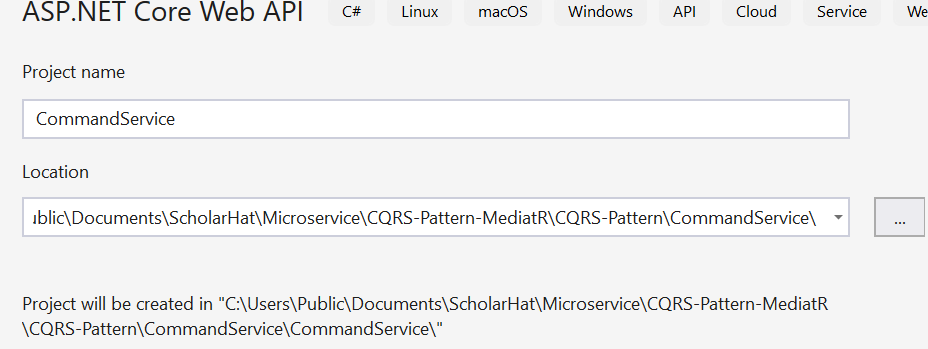
1. Create new empty solution named as CQRS-Pattern. (Create New Project > Blank Solution Template) and select the folder path for solution.



1. Create New project of Asp.Net Core Web API name as CommandService and QueryService under the same created blank solution.
2. Right Click on Empty Solution > Add > New Project > Select ASP.NET Core Web API Template > Next >

Give Project Name and select the folder path for project.

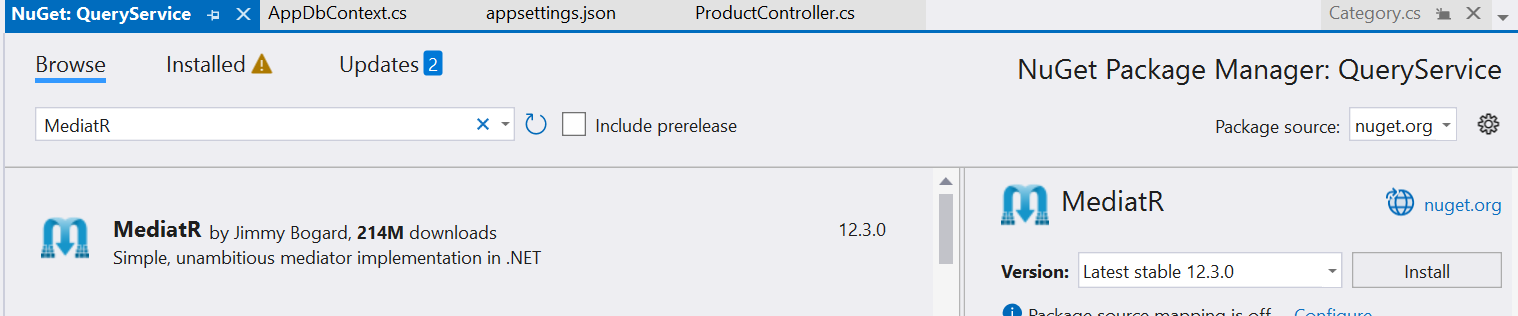
1. Same repeat step 3 for QueryService



A screenshot of a computer

Description automatically generated

1. Add Product controller in QueryService and install the MediatR library from NuGet package.



1. Add Services folder. Add Queries and Handlers sub-folder under Services folder.

A screenshot of a computer

Description automatically generated

* 1. Define a query name as GetProductQuery and GetProductsQuery inside Services > Queries sub-folder of QueryService project for getting Products.
  2. Define the Handler name as GetProductQueryHandler and GetProductsQueryHandler inside Services > Handler sub-folder of QueryService project for handling query.

1. Add the connection string in appsetting.json file in CommandService and QueryService Projects respectively.

"ConnectionStrings": {

"DefaultConnection": "data source=IBM-PF3CYCH3\\SQLEXPRESS; initial catalog=CatalogService06Jul;persist security info=True;user id=Vishwajit;password=November@1417;TrustServerCertificate=True"

}

1. Add the DB First Approach. Right Click on QueryService > EF Core Power Tools > Reverse Engineer > Select EF Core 8 and Clear the existing connection and click Add menu and select Add Database Connection from the Add dropdown menu.

A screenshot of a computer error

Description automatically generated

1. Add Server name and Test Connection. Select CatalogService06Jul ( We created earlier) from the database dropdown menu.

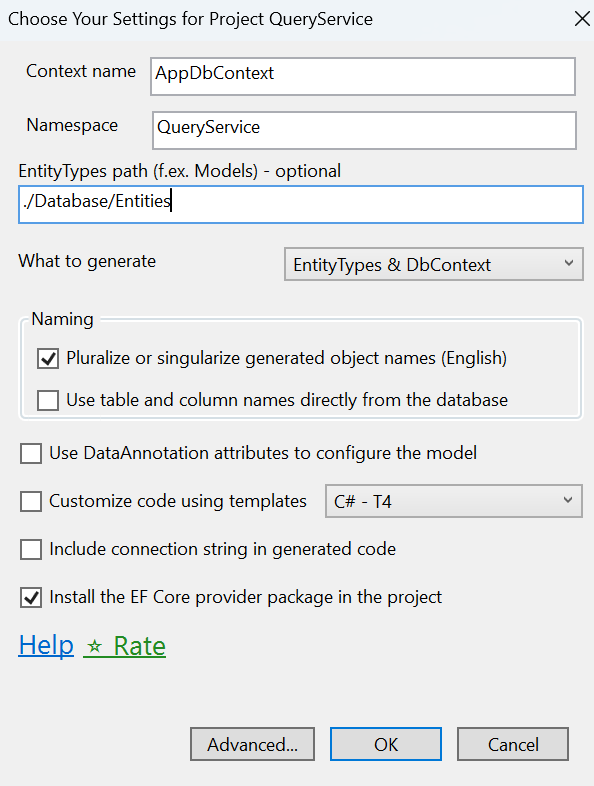
A screenshot of a computer

Description automatically generated

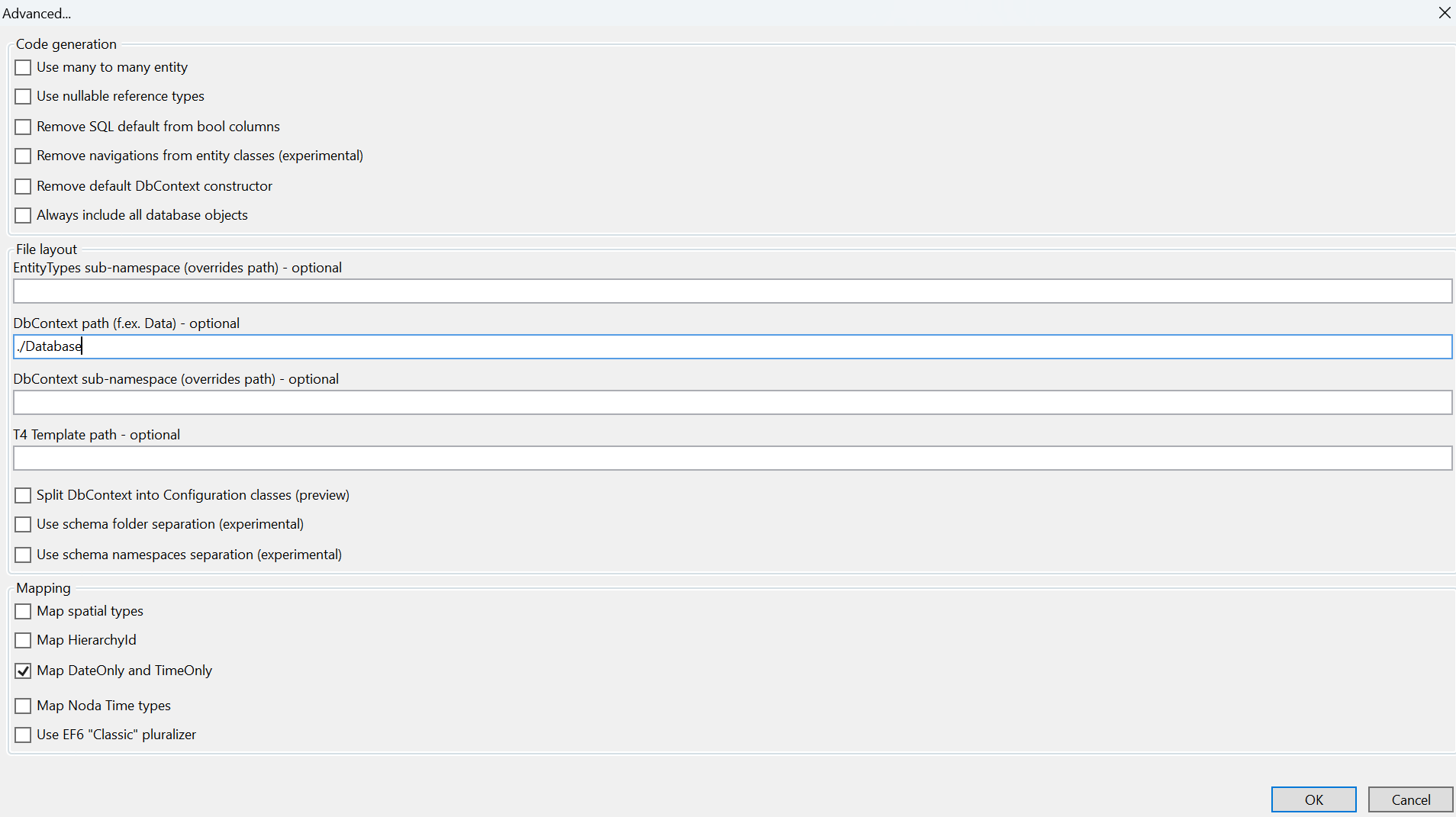
1. Now import the tables.

A screenshot of a computer

Description automatically generated



Click on Advnanced.. and set the DbContext path (./Database)

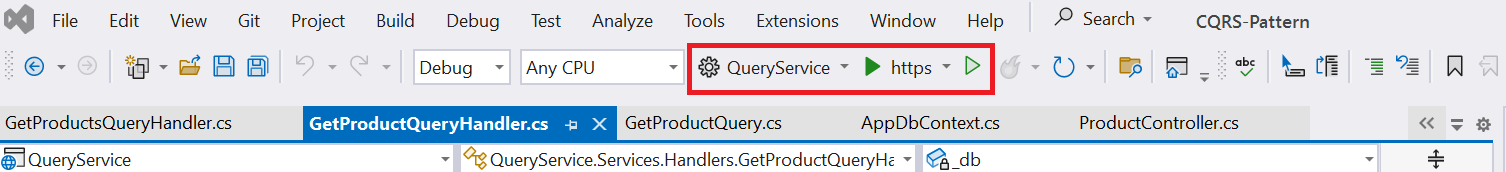


1. Add the below service to the container for Default Connection and MediatR in Program.cs file.

A screenshot of a computer program

Description automatically generated

1. Run the QueryService project using **https** profile and test through QueryService.http file.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A black and white screen

Description automatically generated

1. Same do for the CommandService pattern project.

Eg. Create the Query and Handler Folder under Services folder, Define Controller, Define the connection string in appsetting.json file, Install the MediatR NuGet Package and other package same as QueryService Project , Add services to the container for MediatR and Default connection in Program.cs file. Create DbContext and Entities under Database folder as we did in the QueryService Project.

1. Please refer the code for the same.
2. Execute the CommandService using **https** profile.

