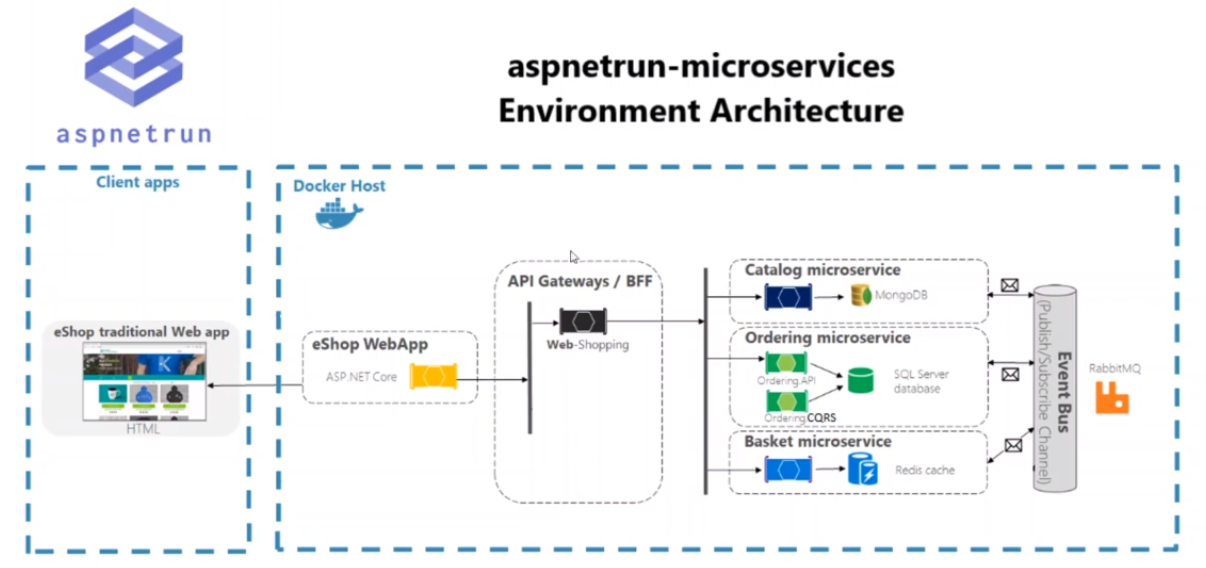
**Section07 Building Ocelot API Gateway Microservices**

**Notes: -**





**Lesson02 API Gateway Design Pattern**

**Notes: -**

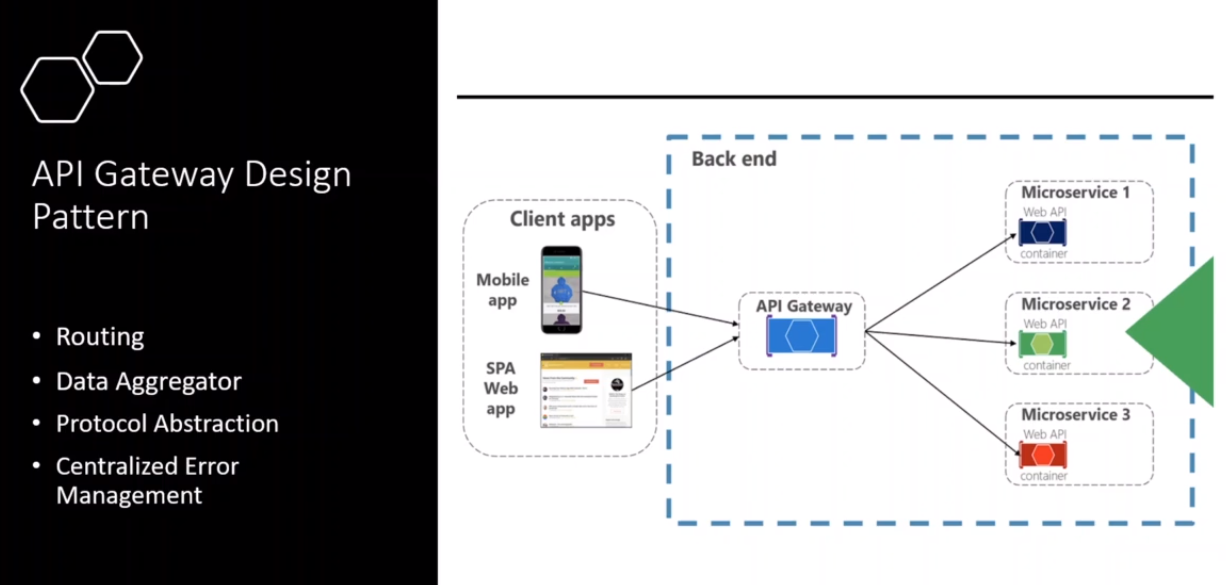
**1-as we see we have unified single entry port that analyze the request to which micro service belong to**

**2-you can make authentication and authorization in the API Gateway**

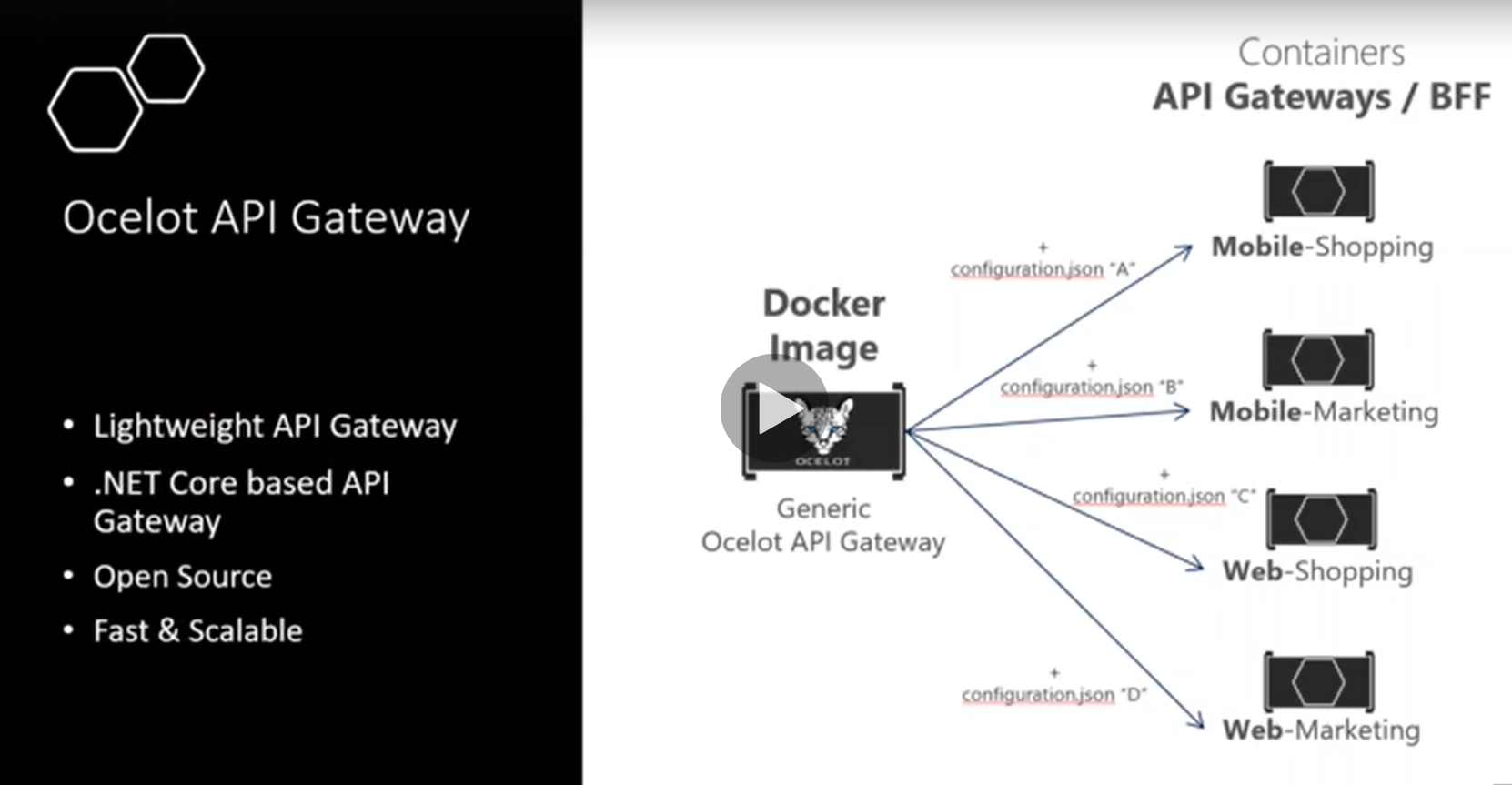
**3-API Gateway provide routing which provide service discovery to which target micro service belong to**

**4-data aggregator, with API Gateway it ensures that the data send from the request and receive should be occur successfully**

**5-Protocol abstraction: with API Gateway it can handle different protocol abstractions**



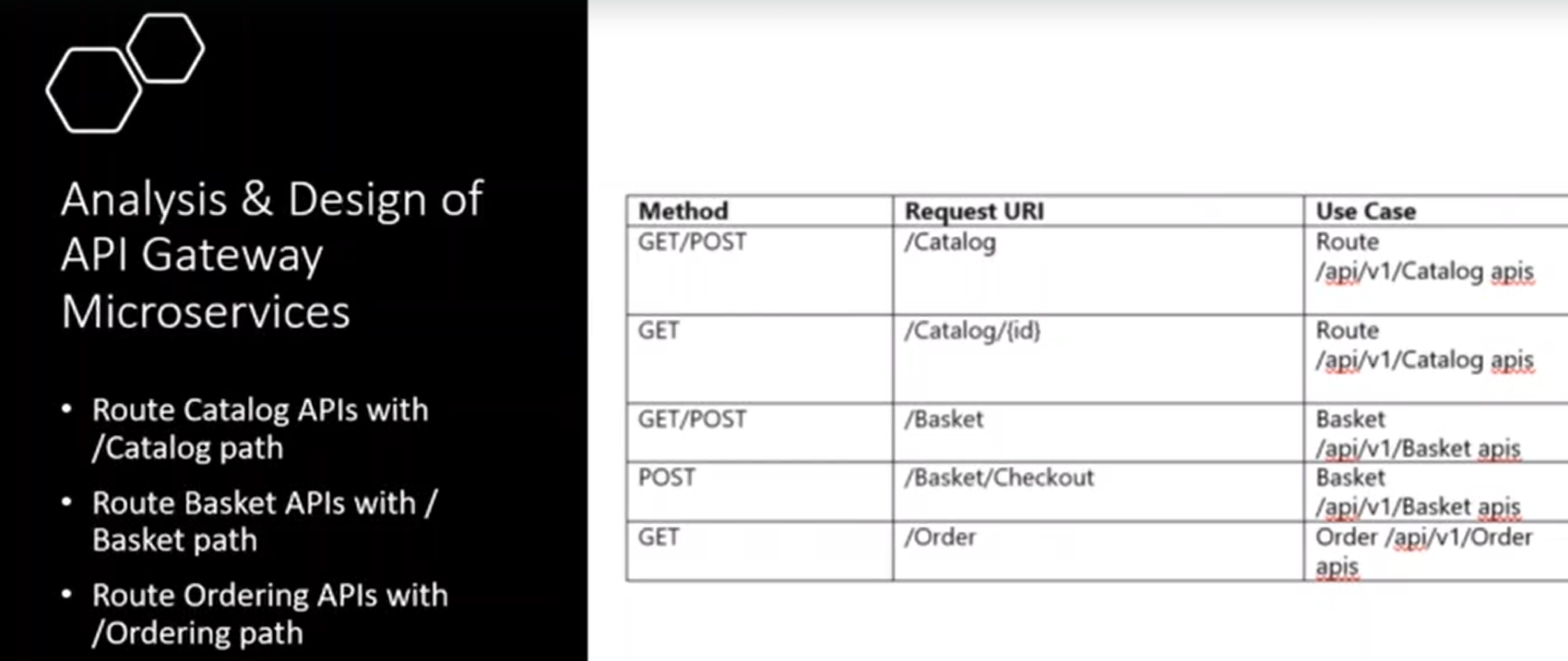
**6-ocelot is lightweight that apply the API Gateway design pattern and its open source lightweight, scalable, auto integration among many other featured, routing**

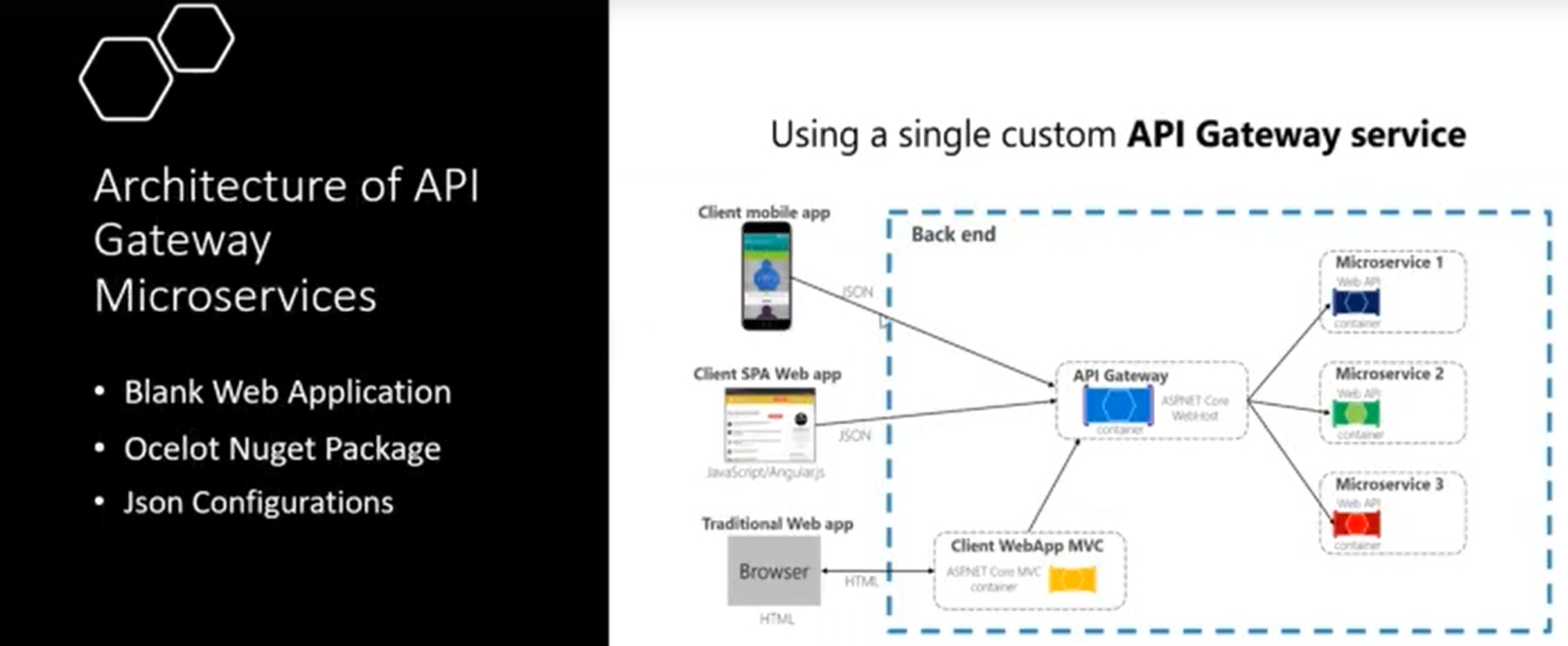


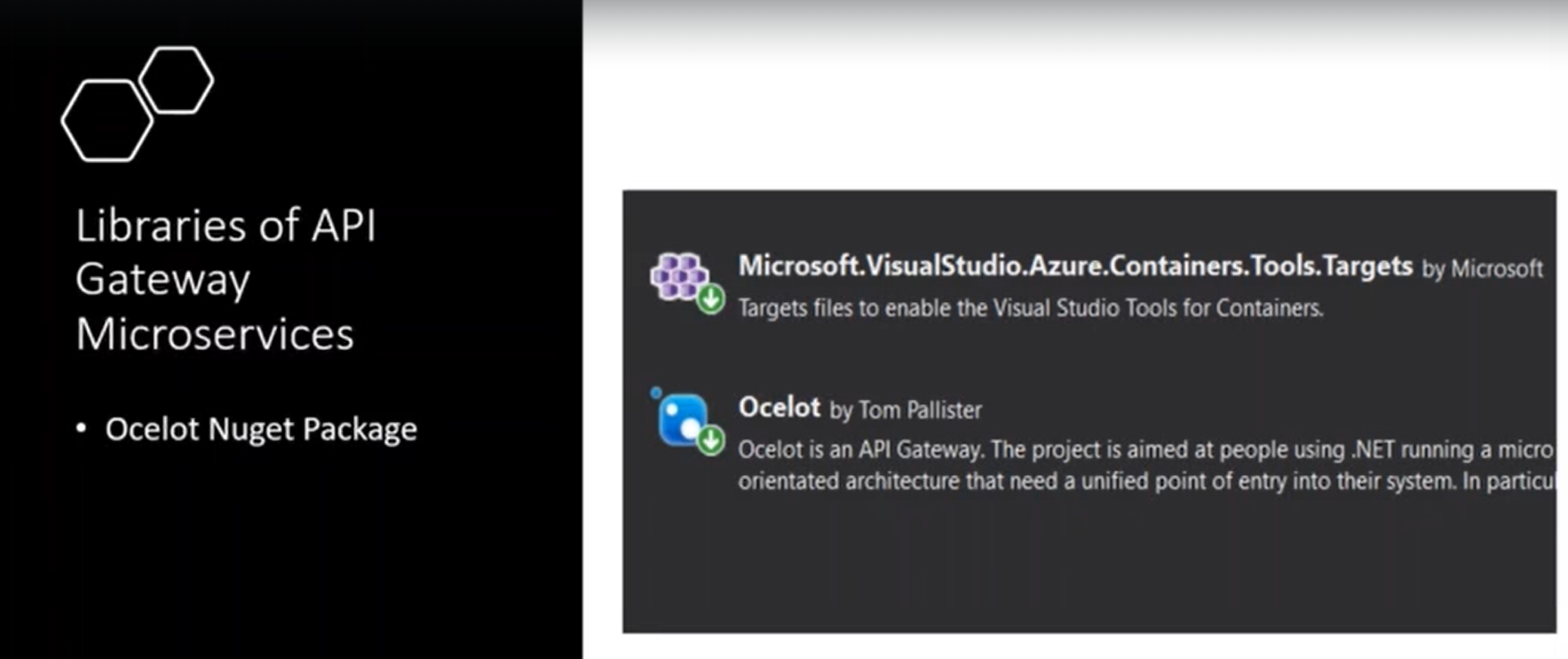
**Lesson03 Analysis & Design Ocelot API Gateway**

**Micro services with Managing REST APIs**

**Notes:-**







**Lesson04 create blank application for ocelot API gateway application**

**Notes:-**

**1-create new project > Asp.net core web application**

**2-on the program.cs we set the following code to detect the file name for configure API Gateway**

**public class Program{**

**public static void Main(string[] args){CreateHostBuilder(args).Build().Run();}**

**public static IHostBuilder CreateHostBuilder(string[] args) =>**

**Host.CreateDefaultBuilder(args)**

**.ConfigureWebHostDefaults(webBuilder =>{**

**webBuilder.UseStartup<Startup>();**

**}).ConfigureAppConfiguration((hostingContext, config) => {**

**config.AddJsonFile("ocelot.json");});}**

**3-on the startup.cs we set the following code**

**public class Startup{**

**public Startup(IConfiguration configuration){Configuration = configuration;}**

**public IConfiguration Configuration { get; }**

**// This method gets called by the runtime. Use this method to add services to the container.**

**public void ConfigureServices(IServiceCollection services){**

**services.AddControllers();**

**services.AddSwaggerGen(c =>{c.SwaggerDoc("v1", new OpenApiInfo { Title = "OcelotAPIGateway", Version = "v1" });});**

**//to add ocelot middleware**

**services.AddOcelot();}**

**// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.**

**public async void Configure(IApplicationBuilder app, IWebHostEnvironment env){**

**if (env.IsDevelopment()){**

**app.UseDeveloperExceptionPage();**

**app.UseSwagger();**

**app.UseSwaggerUI(c => c.SwaggerEndpoint("/swagger/v1/swagger.json", "OcelotAPIGateway v1"));}**

**app.UseRouting();**

**app.UseAuthorization();**

**app.UseEndpoints(endpoints =>{endpoints.MapControllers();});**

**//to using Ocelot middleware**

**await app.UseOcelot();}}**

**4-on the ocelot.json we configure the DownStreamPathTemplate and InputStreamPathTemplate**

{

"Routes": [

//Catalog API

{

"DownstreamPathTemplate": "/api/v1/Catalog",

"DownstreamScheme": "http",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "5000"

}

],

"UpstreamPathTemplate": "/Catalog",

"UpstreamHttpMethod": [ "GET", "POST", "PUT" ]

},

{

"DownstreamPathTemplate": "/api/v1/Catalog/{id}",

"DownstreamScheme": "http",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "5000"

}

],

"UpstreamPathTemplate": "/Catalog/{id}",

"UpstreamHttpMethod": [ "GET", "DELETE" ]

},

{

"DownstreamPathTemplate": "/api/v1/Catalog/GetProductByCategory/{category}",

"DownstreamScheme": "http",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "5000"

}

],

"UpstreamPathTemplate": "/Catalog/GetProductByCategory/{category}",

"UpstreamHttpMethod": [ "GET" ]

},

//Basket API

{

"DownstreamPathTemplate": "/api/v1/Basket",

"DownstreamScheme": "http",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "5001"

}

],

"UpstreamPathTemplate": "/Basket",

"UpstreamHttpMethod": [ "GET", "POST" ]

},

{

"DownstreamPathTemplate": "/api/v1/Basket/{userName}",

"DownstreamScheme": "http",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "5001"

}

],

"UpstreamPathTemplate": "/Basket/{userName}",

"UpstreamHttpMethod": [ "DELETE" ]

},

{

"DownstreamPathTemplate": "/api/v1/Basket/Checkout",

"DownstreamScheme": "http",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "5001"

}

],

"UpstreamPathTemplate": "/Basket/Checkout",

"UpstreamHttpMethod": [ "POST" ]

},

//Order API

{

"DownstreamPathTemplate": "/api/v1/Order",

"DownstreamScheme": "http",

"DownstreamHostAndPorts": [

{

"Host": "localhost",

"Port": "5002"

}

],

"UpstreamPathTemplate": "/Order",

"UpstreamHttpMethod": [ "GET" ]

}

]

}

//{

// "Routes": [

// //Catalog API

// {

// "DownstreamPathTemplate": "/api/v1/Catalog",

// "DownstreamScheme": "http",

// "DownstreamHostAndPorts": [

// {

// "Host": "catalogapi",

// "Port": "80"

// }

// ],

// "UpstreamPathTemplate": "/Catalog",

// "UpstreamHttpMethod": [ "GET", "POST", "PUT" ]

// },

// {

// "DownstreamPathTemplate": "/api/v1/Catalog/{id}",

// "DownstreamScheme": "http",

// "DownstreamHostAndPorts": [

// {

// "Host": "catalogapi",

// "Port": "80"

// }

// ],

// "UpstreamPathTemplate": "/Catalog/{id}",

// "UpstreamHttpMethod": [ "GET", "DELETE" ]

// },

// {

// "DownstreamPathTemplate": "/api/v1/Catalog/GetProductByCategory/{category}",

// "DownstreamScheme": "http",

// "DownstreamHostAndPorts": [

// {

// "Host": "catalogapi",

// "Port": "80"

// }

// ],

// "UpstreamPathTemplate": "/Catalog/GetProductByCategory/{category}",

// "UpstreamHttpMethod": [ "GET" ]

// },

// //Basket API

// {

// "DownstreamPathTemplate": "/api/v1/Basket",

// "DownstreamScheme": "http",

// "DownstreamHostAndPorts": [

// {

// "Host": "basketapi",

// "Port": "80"

// }

// ],

// "UpstreamPathTemplate": "/Basket",

// "UpstreamHttpMethod": [ "GET", "POST" ]

// },

// {

// "DownstreamPathTemplate": "/api/v1/Basket/{userName}",

// "DownstreamScheme": "http",

// "DownstreamHostAndPorts": [

// {

// "Host": "basketapi",

// "Port": "80"

// }

// ],

// "UpstreamPathTemplate": "/Basket/{userName}",

// "UpstreamHttpMethod": [ "DELETE" ]

// },

// {

// "DownstreamPathTemplate": "/api/v1/Basket/Checkout",

// "DownstreamScheme": "http",

// "DownstreamHostAndPorts": [

// {

// "Host": "basketapi",

// "Port": "80"

// }

// ],

// "UpstreamPathTemplate": "/Basket/Checkout",

// "UpstreamHttpMethod": [ "POST" ]

// },

// //Order API

// {

// "DownstreamPathTemplate": "/api/v1/Order",

// "DownstreamScheme": "http",

// "DownstreamHostAndPorts": [

// {

// "Host": "orderingapi",

// "Port": "80"

// }

// ],

// "UpstreamPathTemplate": "/Order",

// "UpstreamHttpMethod": [ "GET" ]

// }

// ]

//}

**5-on the project > right click 🡪 add Docker Support > choose linux**

**6-on the docker-compose.yml , docker-compose-override.yml we set dependencies of the catalog.api,basket.api,ordering.api**

apigateway:

image: ${DOCKER\_REGISTRY-}apigateway

build:

context: .

dockerfile: src/ApiGateway/OcelotAPIGateway/Dockerfile

apigateway:

container\_name: apigateway

environment:

- ASPNETCORE\_ENVIRONMENT=Development

depends\_on:

- catalog.api

- basket.api

- ordering.api

volumes:

- ${HOME}/.microsoft/usersecrets/:/root/.microsoft/usersecrets

- ${HOME}/.aspnet/https:/root/.aspnet/https/

ports:

- "7000:80"