Following are the microservices developed for the described use-case.  
  
1) AuthAPI  
2) ProductAPI  
2) ShoppingCartAPI  
3) OrderAPI  
  
**1) AuthAPI**  
This service is for user registration and user’s authentication and authorization. Using this user can also sign in as administrator. This service makes use of Microsoft’s Identity for the authentication and authorization of a user. When a new user registers a message ”Registration is successful” is sent to the rabbitmq and in this project itself there is message consumer which console logs the aforementioned message.

**2) ProductAPI**  
Using this service user (be it ADMIN or CUSTOMER) can see the products and their details. Only admin user can add, delete, update the product.

**3) ShoppingCartAPI**Using this service user can add items to the cart or remove items from the cart. User can also checkout from the cart using the endpoint provided in the API. This service makes use of **Eureka Service Discovery** to communicate with ProductAPI to fetch products. There is also an **asynchronous communication** between ShoppingCartAPI and OrderAPI, when a shopping cart is checked out a message is sent to the **message broker(rabbitmq)** and then the consumer which is in OrderAPI gets invoked then a new order gets created

**4) OrderAPI**This service is used to create new orders and update the existing orders.  
  
Note: For the asked use-case, I have not created any Payment service, which should kick-in with the checkout in ShoppingCartAPI service.   
**How to run the solution (for windows):**  
1) Copy the path where docker-compose file is stored.  
2) Open command prompt use, cd <path>  
3) Write **wsl** on command prompt  
4) Run docker-compose.yaml file provided by using following command  
**docker-compose up --build -d**5) Run all the services using ocelot gateway on port 5000, as described in the video attached.