

# Microservices on Docker Desktop



Dev Thakkar · [Follow](#)

3 min read · Mar 19, 2024



**Scenario:** Separately publish 4 microservices to Docker Desktop

**Tech stack:** .NET Core, Visual Studio, Docker Desktop (on Windows)

**GitHub:** [Ecommerce — Docker Run](#)

## (1) Services

- **CustomerService** : data is stored as in-memory database (CustomersProvider.cs file)

```
# example data
```

```
{ Id = 1, Name = "Test1 Client1", Address = "20 Elm St." }  
{ Id = 2, Name = "Test2 Client2", Address = "30 Main St." }  
{ Id = 3, Name = "Test3 Client3", Address = "100 10th St." }
```

```
#build docker image
```

```
docker build -t customerservice .
```

```
#Run docker command to publish customerservice
```

```
docker run -it --rm -p 3000:80 --name customerservicecontainer customerservice
```

- **OrderService** : data is stored as in-memory database (OrdersProvider.cs file)

```
# example data
```

```
{  
  Id = 1,  
  CustomerId = 1,  
  OrderDate = DateTime.Now,  
  Items = new List<OrderItem>()  
  {  
    new OrderItem() { OrderId = 1, ProductId = 1, Quantity = 10, UnitPrice = 10 }  
    new OrderItem() { OrderId = 1, ProductId = 2, Quantity = 10, UnitPrice = 10 }  
    new OrderItem() { OrderId = 1, ProductId = 3, Quantity = 10, UnitPrice = 10 }  
    new OrderItem() { OrderId = 2, ProductId = 2, Quantity = 10, UnitPrice = 10 }  
    new OrderItem() { OrderId = 3, ProductId = 3, Quantity = 1, UnitPrice = 100 }  
  },  
  Total = 100  
});
```

```
{  
  Id = 2,  
  CustomerId = 1,  
  OrderDate = DateTime.Now.AddDays(-1),  
  Items = new List<OrderItem>()  
  {  
    new OrderItem() { OrderId = 1, ProductId = 1, Quantity = 10, UnitPrice = 10 }  
    new OrderItem() { OrderId = 1, ProductId = 2, Quantity = 10, UnitPrice = 10 }  
    new OrderItem() { OrderId = 1, ProductId = 3, Quantity = 10, UnitPrice = 10 }  
    new OrderItem() { OrderId = 2, ProductId = 2, Quantity = 10, UnitPrice = 10 }  
    new OrderItem() { OrderId = 3, ProductId = 3, Quantity = 1, UnitPrice = 100 }  
  },  
  Total = 100  
});
```

```
{  
  Id = 3,  
  CustomerId = 2,  
  OrderDate = DateTime.Now,  
  Items = new List<OrderItem>()  
  {  
    new OrderItem() { OrderId = 1, ProductId = 1, Quantity = 10, UnitPrice = 10 }  
    new OrderItem() { OrderId = 2, ProductId = 2, Quantity = 10, UnitPrice = 10 }  
    new OrderItem() { OrderId = 3, ProductId = 3, Quantity = 1, UnitPrice = 100 }  
  },  
  Total = 100  
});
```

```
#build docker image
docker build -t orderservice .

#Run docker command to publish orderservice
docker run -it --rm -p 3002:80 --name orderservicecontainer orderservice
```

- **ProductService** : data is stored as in-memory database (ProductsProvider.cs file)

```
# example data

{ Id = 1, Name = "Keyboard", Price = 20, Inventory = 100 }
{ Id = 2, Name = "Mouse", Price = 5, Inventory = 200 }
{ Id = 3, Name = "Monitor", Price = 150, Inventory = 1000 }
{ Id = 4, Name = "CPU", Price = 200, Inventory = 2000 }
```

```
#build docker image
docker build -t productservice .

#Run docker command to publish productservice
docker run -it --rm -p 3001:80 --name productservicecontainer productservice
```

- **SearchService** : Fetches customer orders based on customerid

```
#pseudo code

(1) Fetch from CustomerService using Id
(2) Fetch all the Orders from OrderService with same Id as in (1)
(3) Fetch all the products from ProductService
(4) Loop through the Orders and map product description
```

```
#build docker image
docker build -t searchservice .
```

```
#Run docker command to publish searchservice
```

```
docker run -it --rm -p 3003:80 --name searchservicecontainer searchservice
```

(2) API calls are setup in “appsettings.json” file of SearchService

```
"Services": {  
  "Products": "http://localhost:3001",  
  "Orders": "http://localhost:3002",  
  "Customers": "http://localhost:3000"  
}
```

## Written by Dev Thakkar

3 Followers · 3 Following

### No responses yet



What are your thoughts?

Respond