**How to Use Docker Network Instead**

If you're using Docker's networking capabilities (which is recommended for multi-container applications), you **don't need --add-host**. Docker provides built-in DNS resolution for containers on the same custom network.

**Steps:**

1. **Create a Docker Network**:

docker network create my-network

1. **Run the Eureka Server** on the custom network:

docker run --network my-network --name eureka-server -p 8761:8761 eureka-server

1. **Run Microservices** on the same network:

docker run --network my-network --name microservice1 -p 8081:8081 microservice1

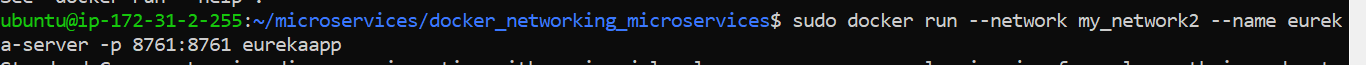
docker run --network my-network --name microservice2 -p 8082:8082 microservice2

1. **Configure Microservices to Use Eureka Server**: In the application.yml of the microservices, set defaultZone to the Eureka Server's container name (e.g., eureka-server):

eureka.client.service-url.defaultZone: <http://eureka-server:8761/eureka/>

//here eureka-server is the name of the container

Since all containers are on the same Docker network, Docker's internal DNS will resolve the hostname eureka-server to the correct container's IP address.







Application.properties file for productcataligue and order-service

eureka.client.service-url.defaultZone: <http://eureka-server:8761/eureka/>

here eureka-server is the name of the container