It is important to understand Docker and how it operates before getting into how the containers communicate.

Docker is a platform for developing, shipping, and running applications inside lightweight, isolated containers. Containers package an application and its dependencies, ensuring consistency across different environments. Below is a high level walkthrough in setting up a docker container.

1. Install Docker – Download and install Docker from docker.com.
2. Verify Installation – Run docker --version to confirm Docker is installed.
3. Create a Dockerfile – Define how your app should be built inside a container.
    a. We already have this.
4. Build a Docker Image – Use docker build -t my-app . to package your app.
5. Run a Container – Start your app with docker run -d -p 8080:8080 my-app.
6. Create a Docker Network – docker network create my-network to allow container communication.
7. Run API Gateway and Microservices – Start services using
    a. docker run --network=my-network ...
8. Use Docker Compose (Optional) – Define all services in a docker-compose.yml file for easier management.

How Docker containers communicate:

Docker containers communicate through an API Gateway by utilizing a shared network that enables seamless service-to-service communication. The API Gateway serves as the single entry point for external requests, directing traffic to the appropriate microservices based on predefined routes. Each container registers with a common Docker network, allowing them to interact using service names instead of IP addresses (e.g., http://service1:port). The API Gateway efficiently forwards requests, handles load balancing to distribute traffic across multiple instances, and enforces security policies such as authentication and rate limiting. Additionally, it can perform request transformation, caching, and logging to optimize performance. By centralizing control, the API Gateway simplifies microservice interactions, improves scalability, and enhances security while abstracting internal service complexities from the client.