

# An-Najah National University Faculty of Engineering

**Computer Engineering Department**

## Distributed Operation Systems

Lab 1: Bazar.com: A Multi-tier Online Book Store

## Layan Othman, Nagham Hanini

[Contents 1](#_bookmark0)

[Introduction 2](#_bookmark1)

[Materials 2](#_bookmark2)

[Cache Consistency 2](#_bookmark3)

1. [cache timing 2](#_bookmark4)
2. [Invalidate Message 2](#_bookmark5)

[Replecation 3](#_bookmark7)

[Loadbalancing 3](#_bookmark8)

[Dockerize your Application (Optional part) 3](#_bookmark9)

[Conclusion 6](#_bookmark11)

**Introduction**

This project aims to enhance the performance of Bazar.com, a simulated online bookstore. As Bazar.com became more popular, users will complain about slow request processing, by introducing replication, caching, and consistency mechanisms. These techniques are commonly used to improve the scalability.

# Materials

To complete this lab, we used the following:

##### Docker Desktop

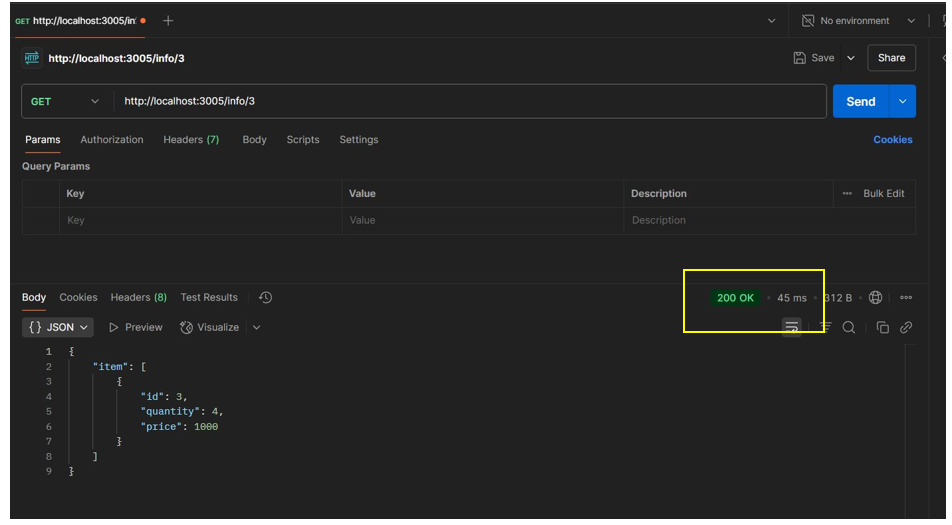
* **Docker Compose**

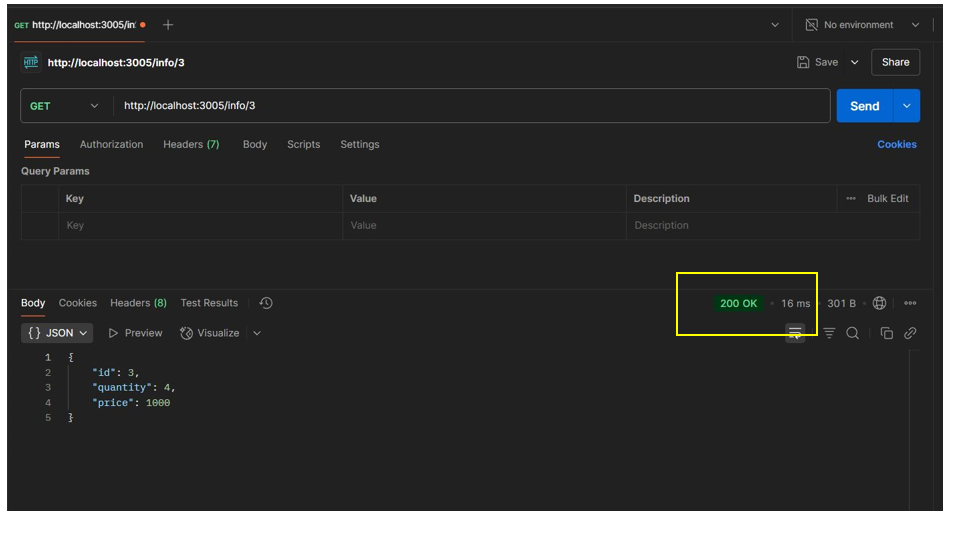
##### Node.js (Express)

* **Postman (Test API)**

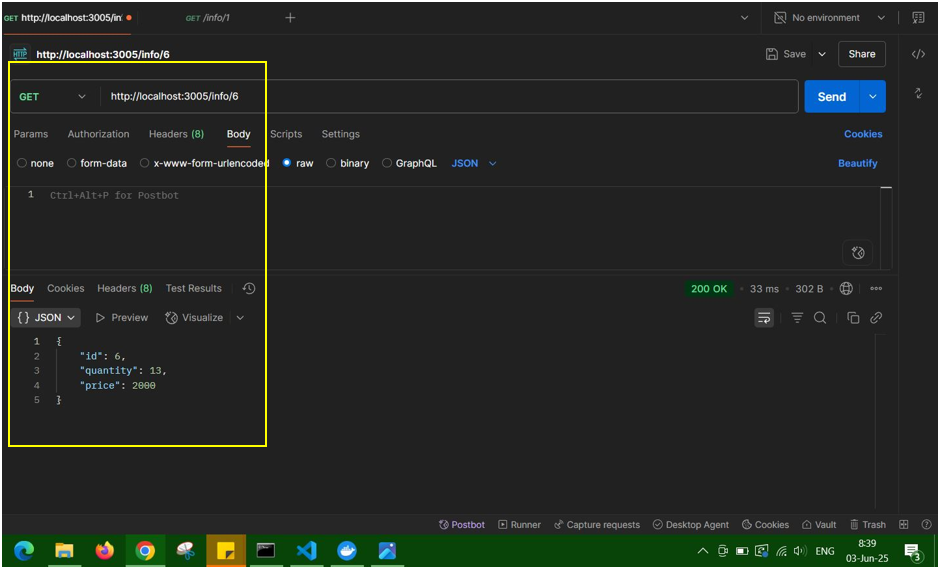
**Cache Consistency**

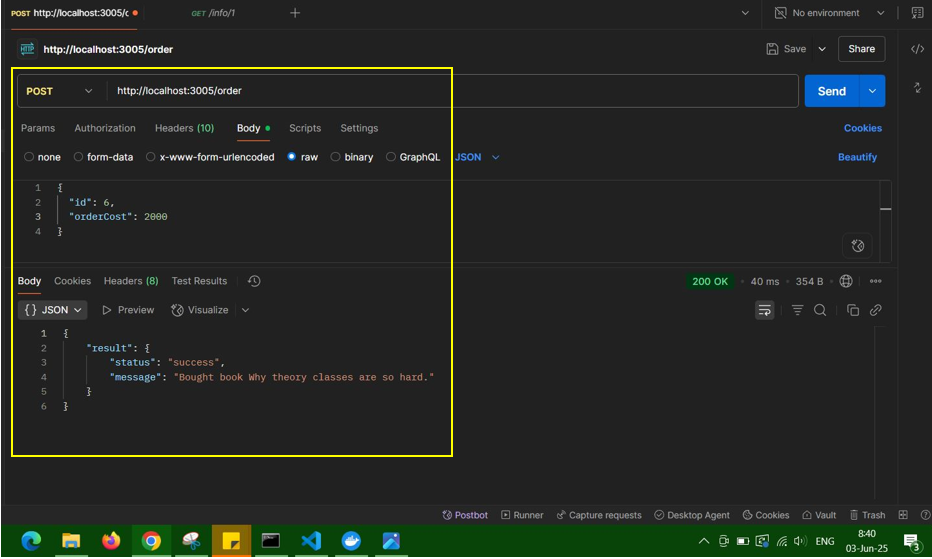
Cache timing

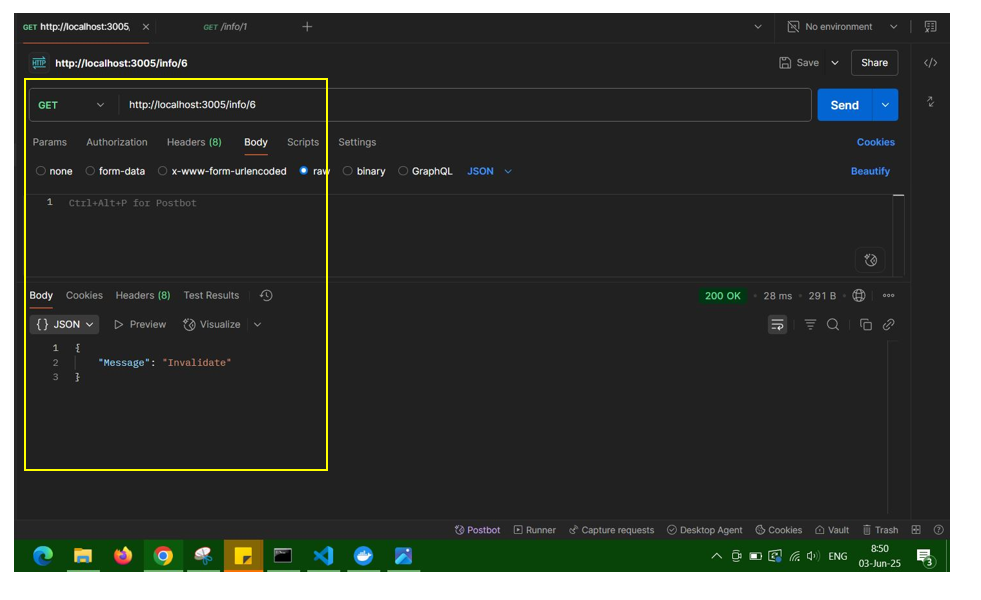


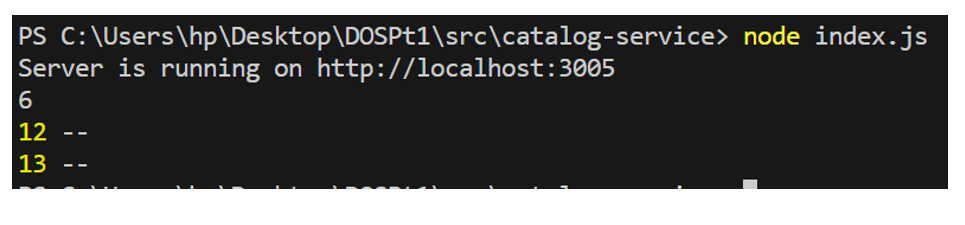


* Invalidate cache







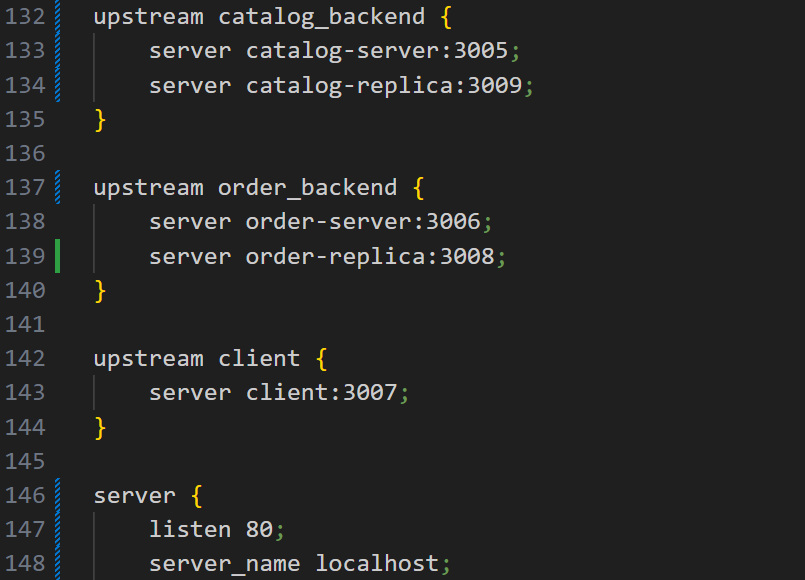


**Replecation :**

To improve **scalability**, **fault tolerance**, and **performance** of the Bazar system by:

1. Adding **replication** for the Catalog and Order services.
2. Using **NGINX** as a load balancer to distribute traffic across replicas.
3. Introducing **in-memory caching** via Redis to speed up read operations.

**NGINX Load Balancing**

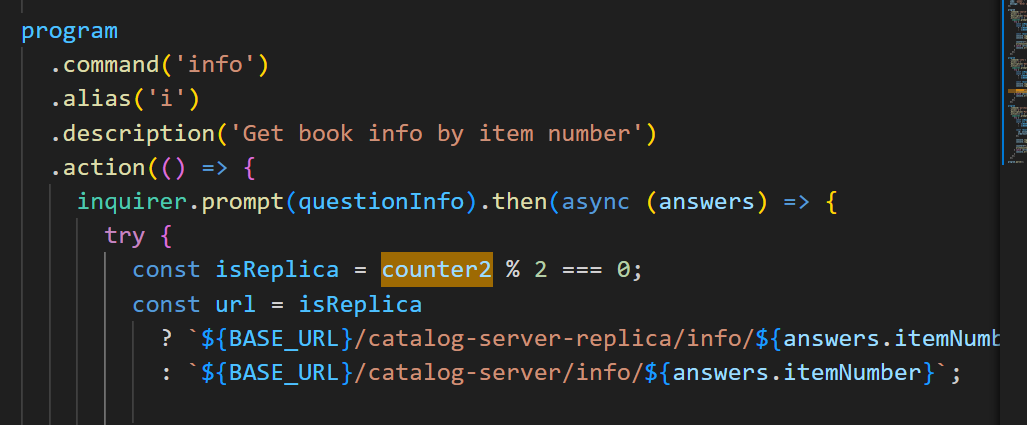
* NGINX serves as a reverse proxy for the following upstreams:
  + catalog\_servers (port 3005 + 3009)
  + order\_servers (port 3006 + 3008)
* Each upstream balances between the primary and replica.
* NGINX uses **round-robin** as the default load balancing strategy.
* 

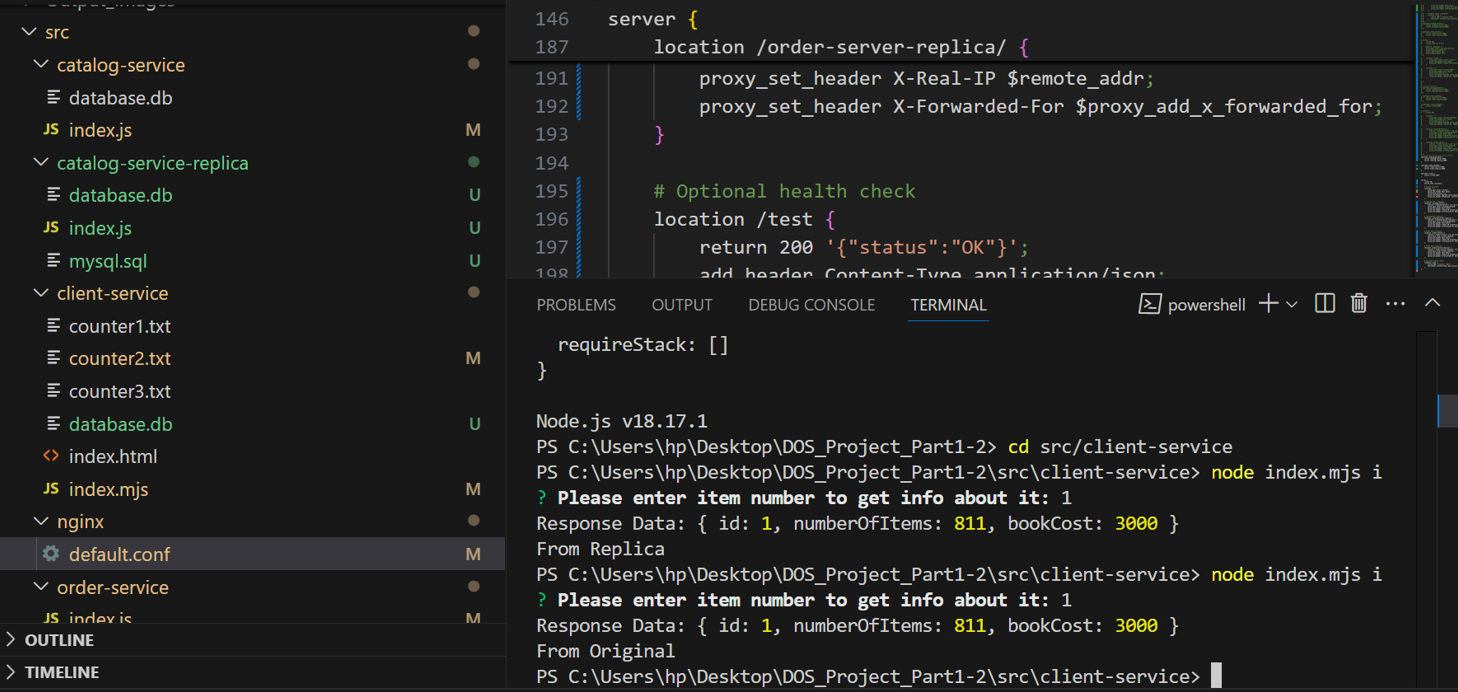
Client-Side Load Balancing (Fallback & Counters)

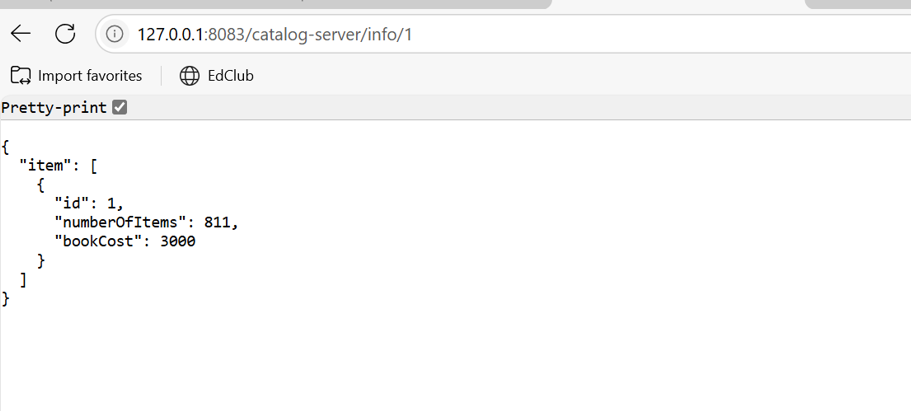
Client CLI tracks the number of requests to each service.

Uses local text files as counters.

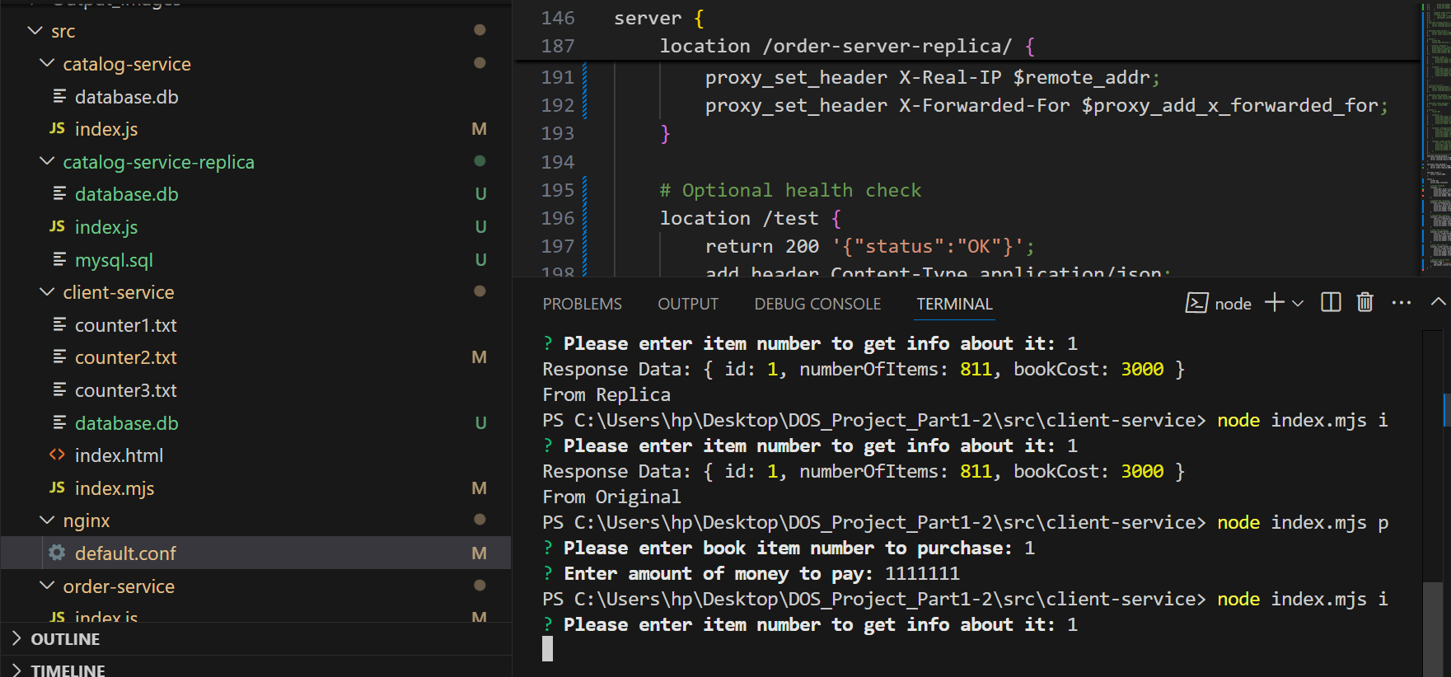
Alternates requests between main and replica using modulo logic (counter % 2 === 0).

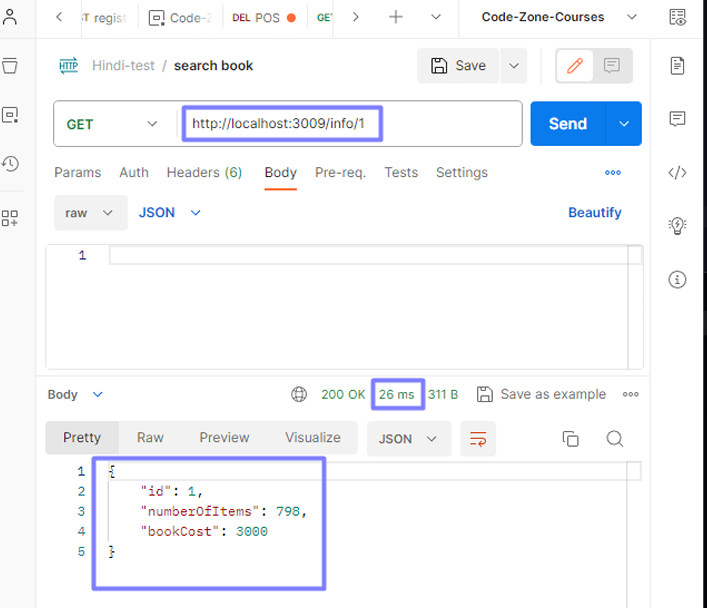










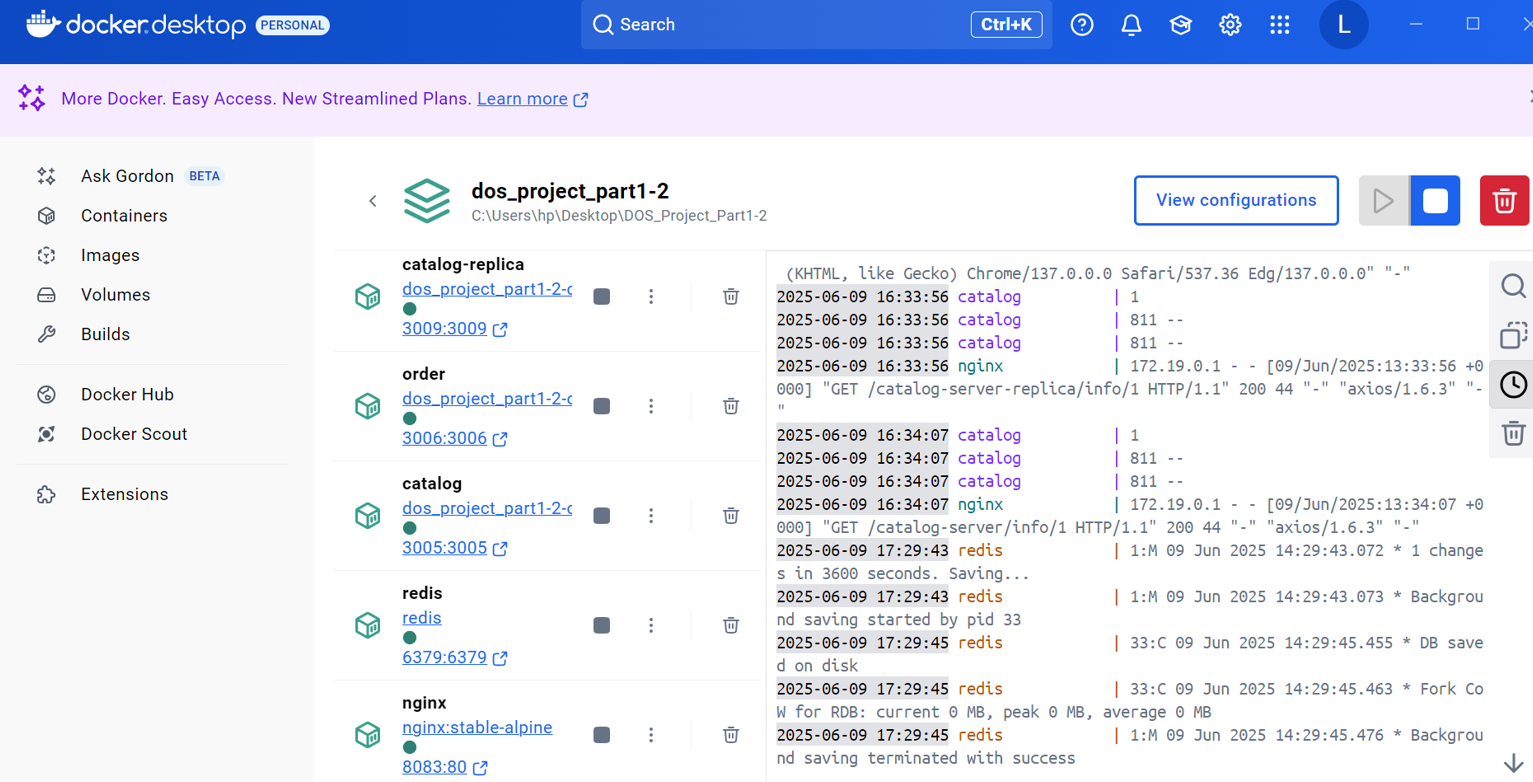


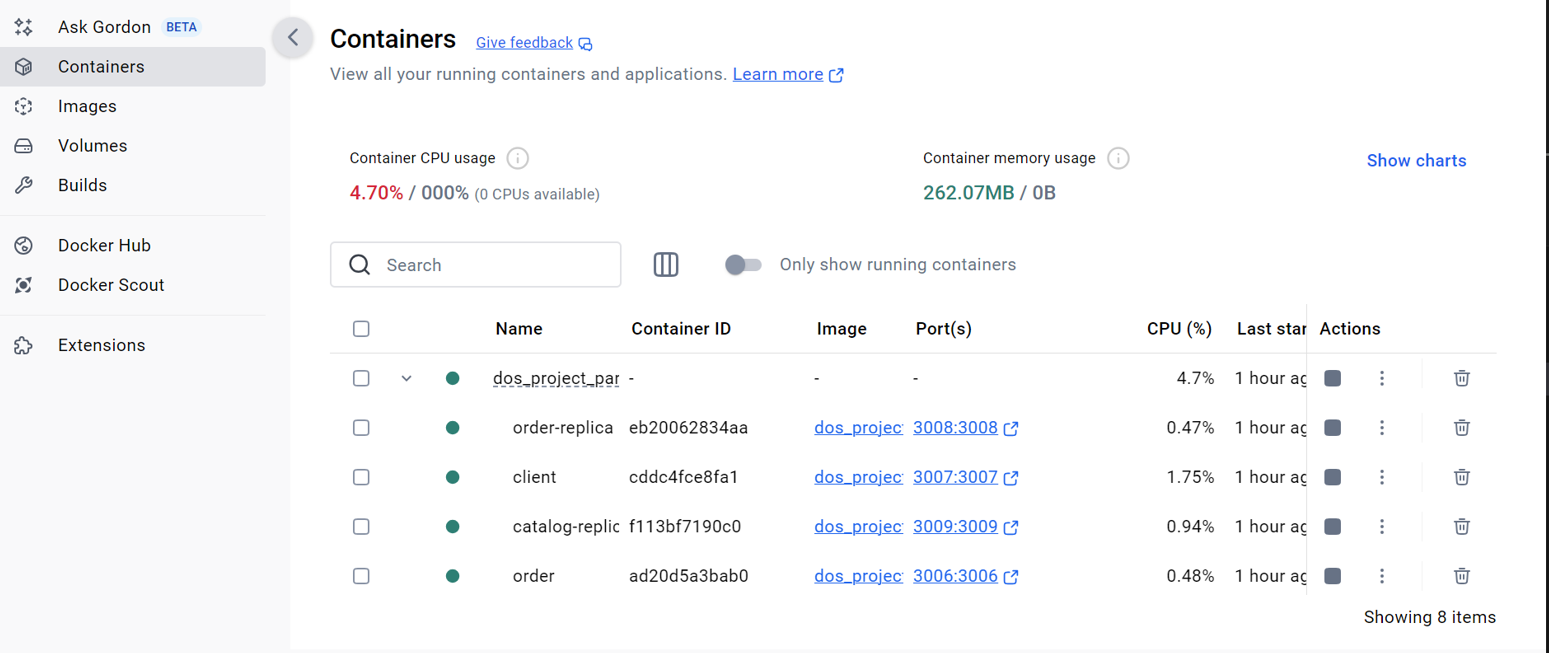
# 

# Setup and Installation

* + 1. Clone the repository to your local machine: https://github.com/layanothman/dosprojectp1.git
    2. Navigate to the project directory
    3. Build the Docker images and start the containers: docker-compose up –build
    4. The system should now be running on Docker containers.

**Dockerize your Application (Optional part):**





In this project, we successfully implemented several techniques aimed at improving the performance and scalability of Bazar.com, transforming it into a more efficient and responsive online bookstore system. The use of Docker containers developed an easy deployment process whereby each service can be deployed and managed independently. Future Work: • Cache Eviction Strategies: Implementing a more sophisticated cache eviction policy, such as Least Recently Used (LRU). • Horizontal Scaling: allowing for more replicas to handle traffic.