Міністерство освіти і науки України

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Фізико-технічний інститут

ПРОЕКТУВАННЯ РОЗПОДІЛЕНИХ СИСТЕМ

Проект

Ітерація №2

Виконав студент гр. ФБ-31мп:

Шевченко С. Ш.

Київ – 2024

# Загальна інформація:

[Репозиторій](https://github.com/dedkuzmich/distributed-systems-design-2024) Github з кодом (папка [project/v2](https://github.com/dedkuzmich/distributed-systems-design-2024/tree/main/project/v2))

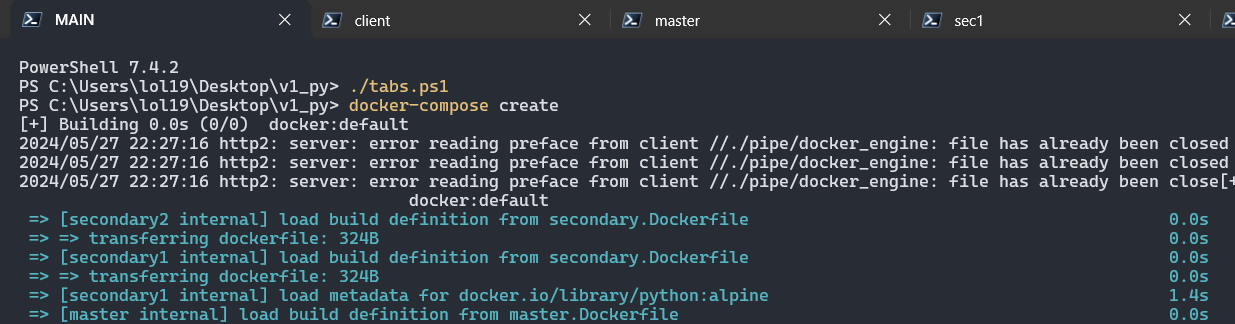
# Демонстрація роботи:

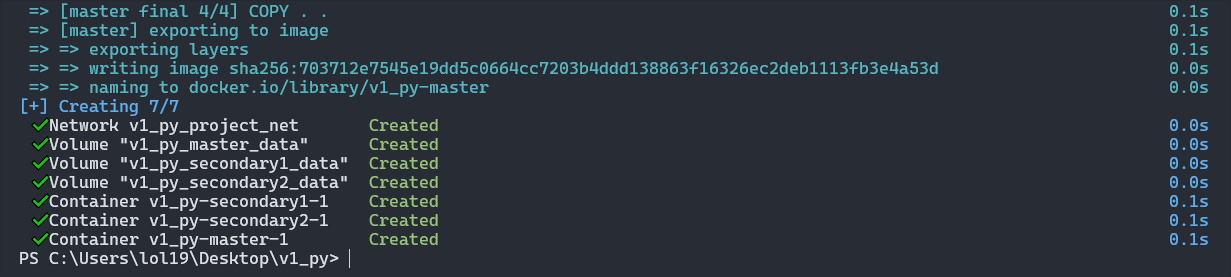
Створимо 4 вкладки у Windows Terminal за допомогою PowerShell 7 скрипту tab.ps1:

**$ ./tabs.ps1**

Створимо Docker контейнери:

**$ docker-compose create**





Видалити контейнери можна так:

**$ docker-compose down --rmi all --volumes**

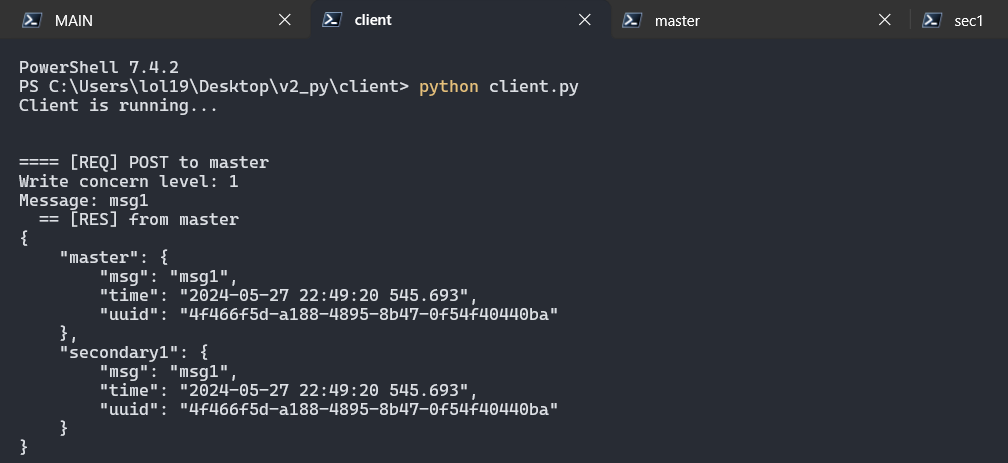
Запустимо всі сервіси **окрім secondary2** в такій послідовності у відповідних вкладках:

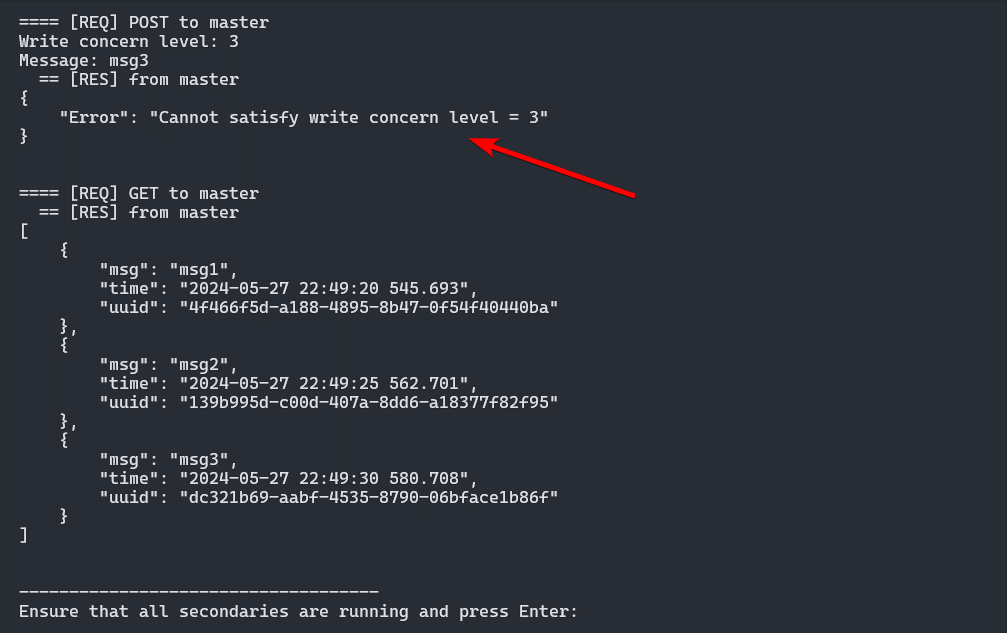
**$ docker-compose up master --no-recreate**

**$ docker-compose up secondary1 --no-recreate**

**$ python client.py**

## Вивід client



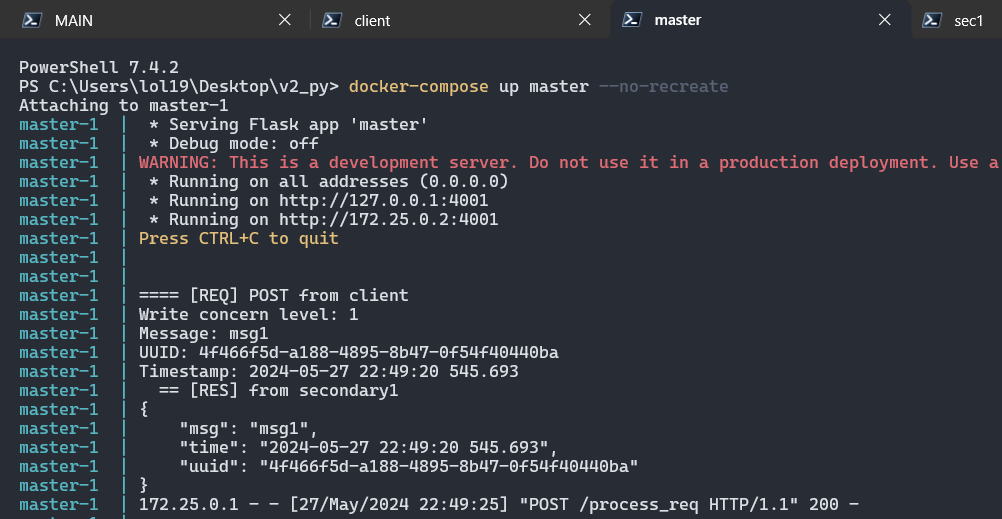


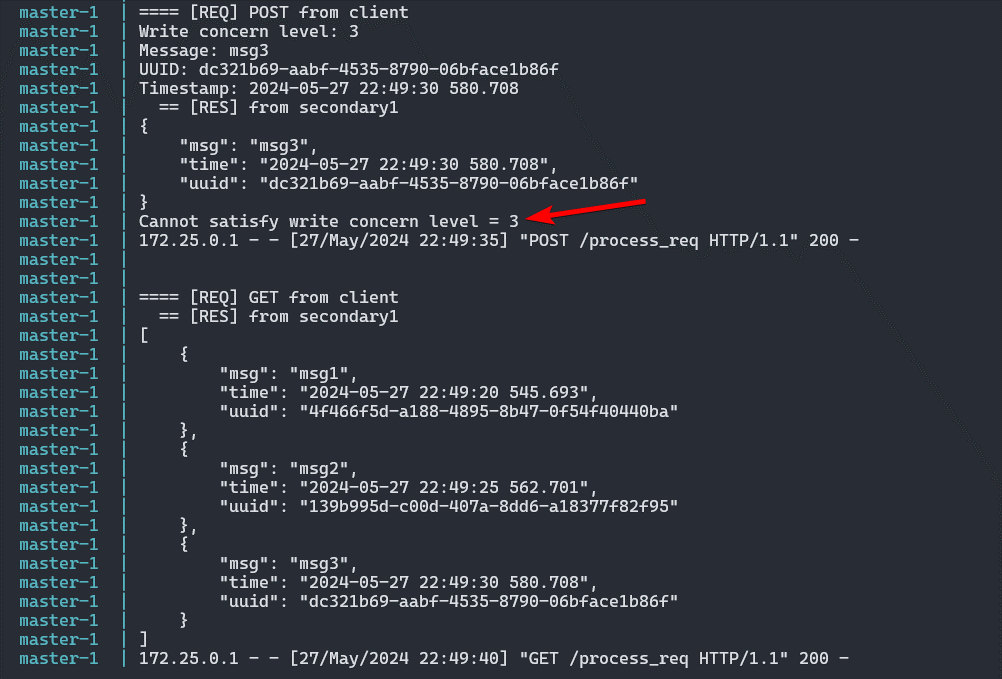
POST-запитом відправляються 3 повідомлення "msg1"-"msg3" на master. Master повертає поточне повідомлення зі згенерованим UUID та міткою часу. Видно, що при write concern = 3, master повертає помилку, так як не може задовільнити такий рівень write concern.

GET-запитом клієнт зчитує всі збережені повідомлення.

Після цього, клієнт очікує включення всіх secondaries.

## Вивід master

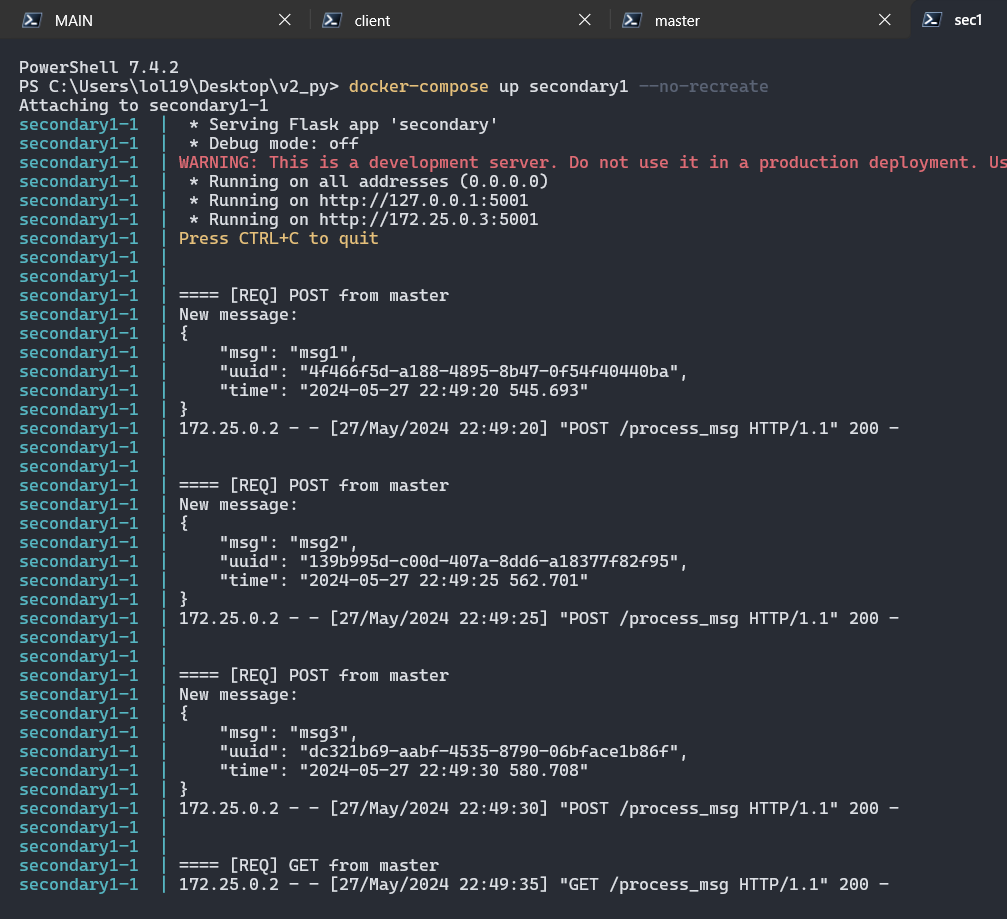




При надходженні POST-запиту з повідомленням, master генерує для нього UUID, мітку часу та відправляє дані на всі secondaries. Якщо к-сть відповідей від secondaries >= рівня write concern, то операція вважається успішною.

При надходженні GET-запиту, master зчитує повідомлення із secondaries, проводить дедублікацію, сортує за часом та надсилає їх клієнту. Якщо жоден secondary не повертає відповідь, то master надсилає клієнту повідомлення зі свого списку.

## Вивід secondary1



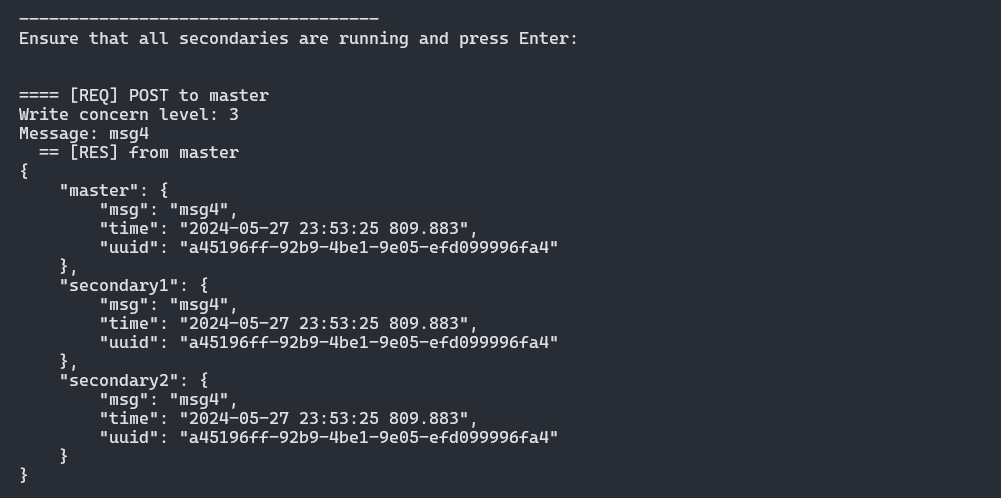
При надходженні POST-запиту, secondary додає повідомлення, його UUID та мітку часу в список.

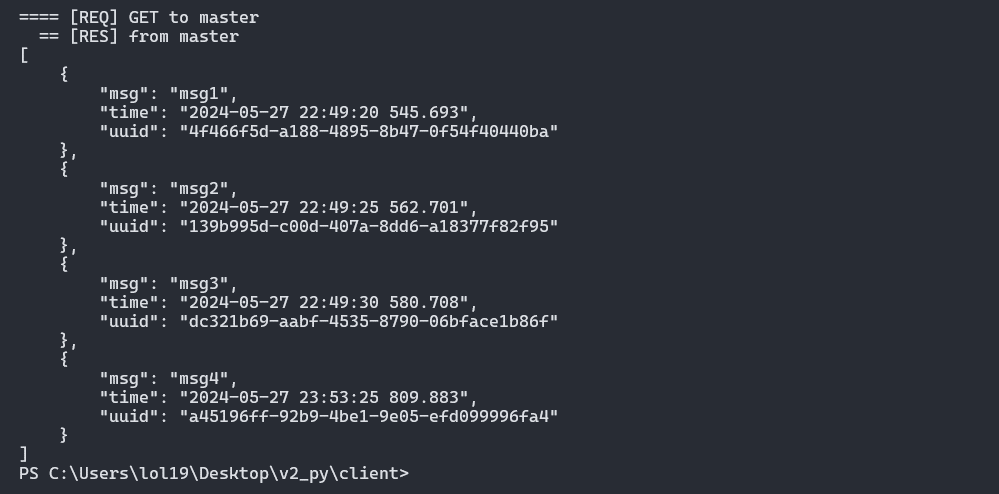
При надходженні GET-запиту, secondary надсилає вміст свого списку.

Тепер запустимо secondary2:

**$ docker-compose up secondary2 --no-recreate**

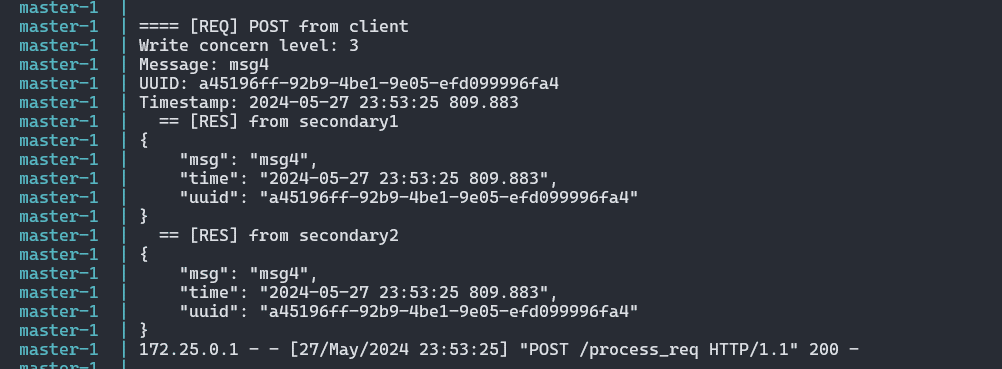
Повернемося до клієнта і натиснемо Enter:

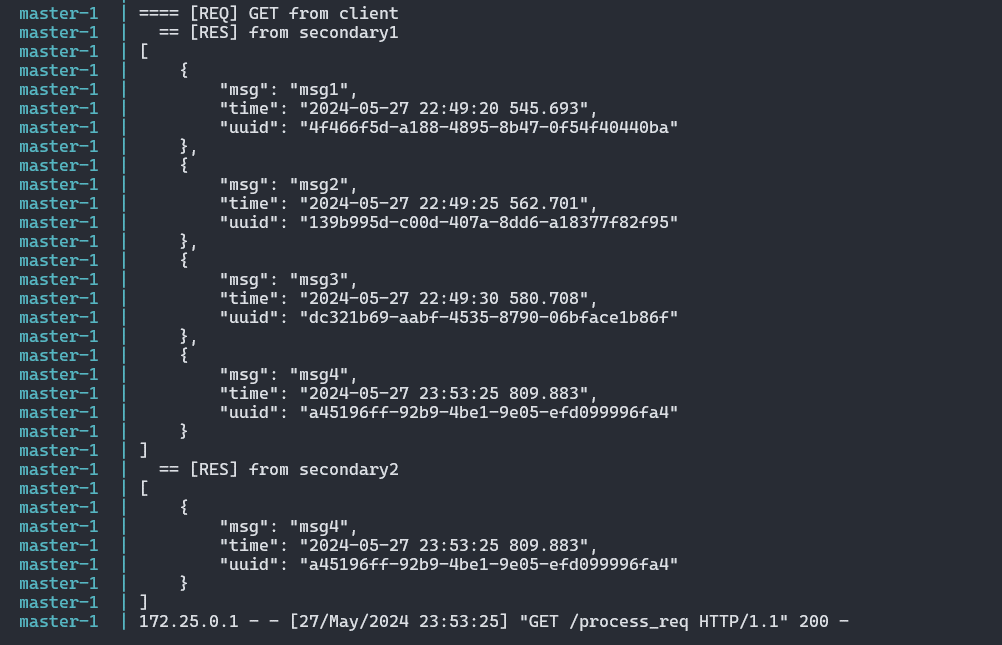




Тепер реплікація завершується коректно, write concern = 3 задовільнено.

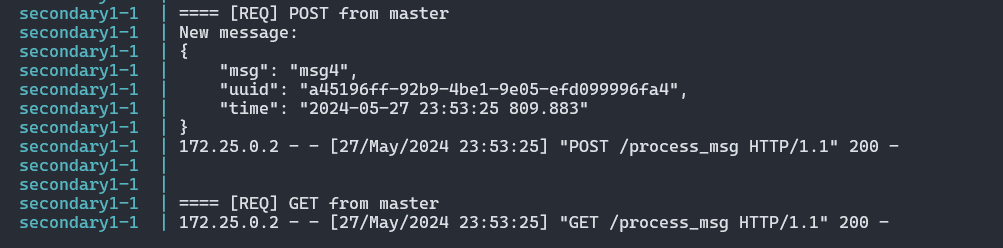
## Вивід master





При обробці GET запиту чітко видно, що secondary1 та secondary2 повертають різні списки (replica’s inconsistency).

## Вивід secondary1



## Вивід secondary2

