Міністерство освіти і науки України

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Фізико-технічний інститут

ПРОЕКТУВАННЯ РОЗПОДІЛЕНИХ СИСТЕМ

Проект

Ітерація №1

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/v1](https://github.com/dedkuzmich/distributed-systems-design-2024/tree/main/project/v1))

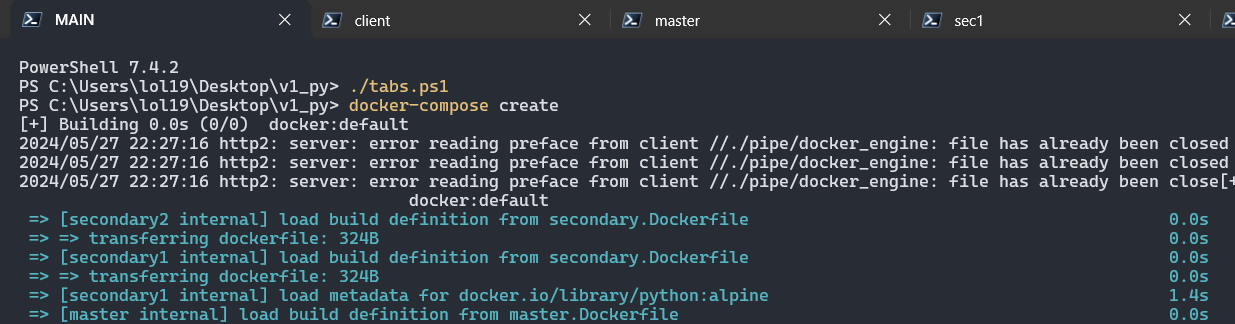
# Демонстрація роботи:

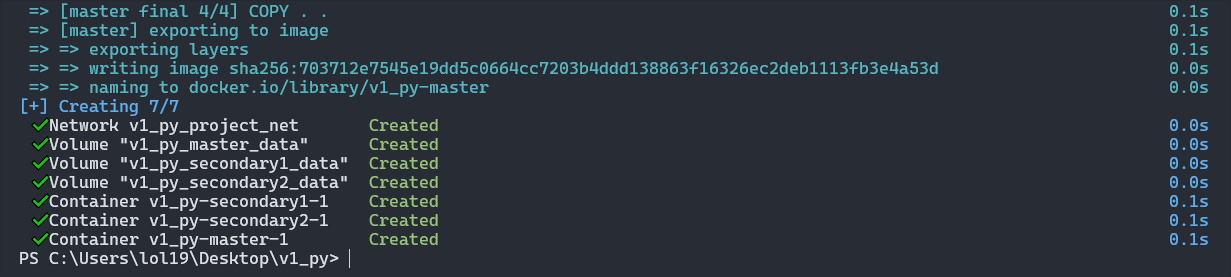
Створимо 4 вкладки у Windows Terminal за допомогою PowerShell 7 скрипту tab.ps1:

**$ ./tabs.ps1**

Створимо Docker контейнери:

**$ docker-compose create**





Видалити контейнери можна так:

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

Запустимо всі сервіси в такій послідовності у відповідних вкладках:

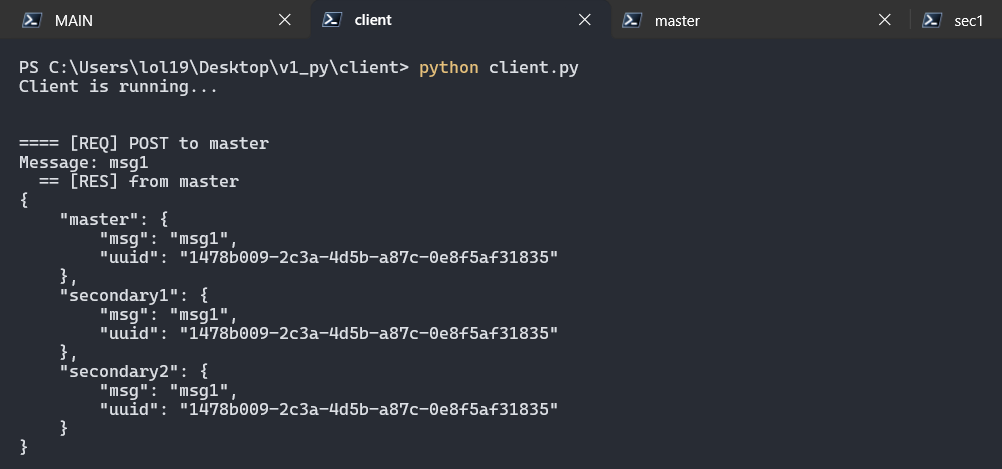
**$ docker-compose up master --no-recreate**

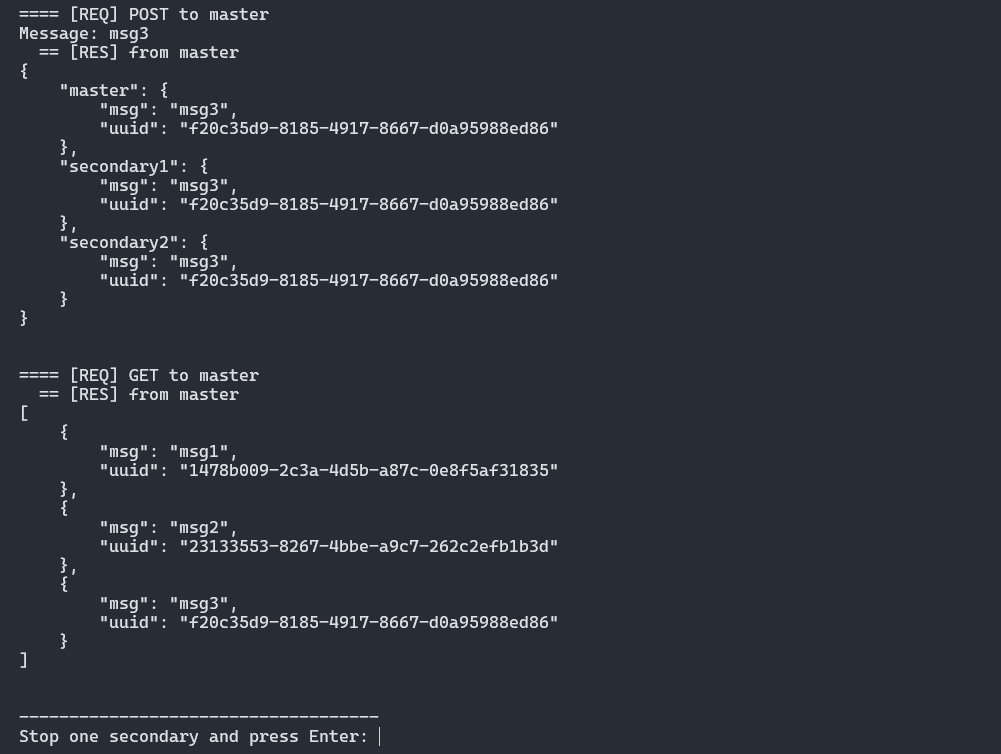
**$ docker-compose up secondary1 --no-recreate**

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

**$ python client.py**

## Вивід client



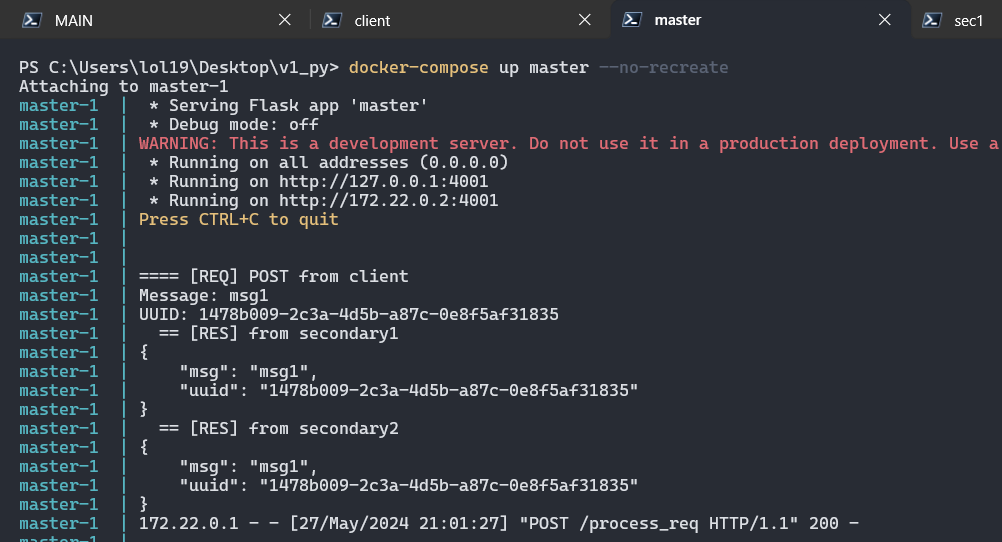


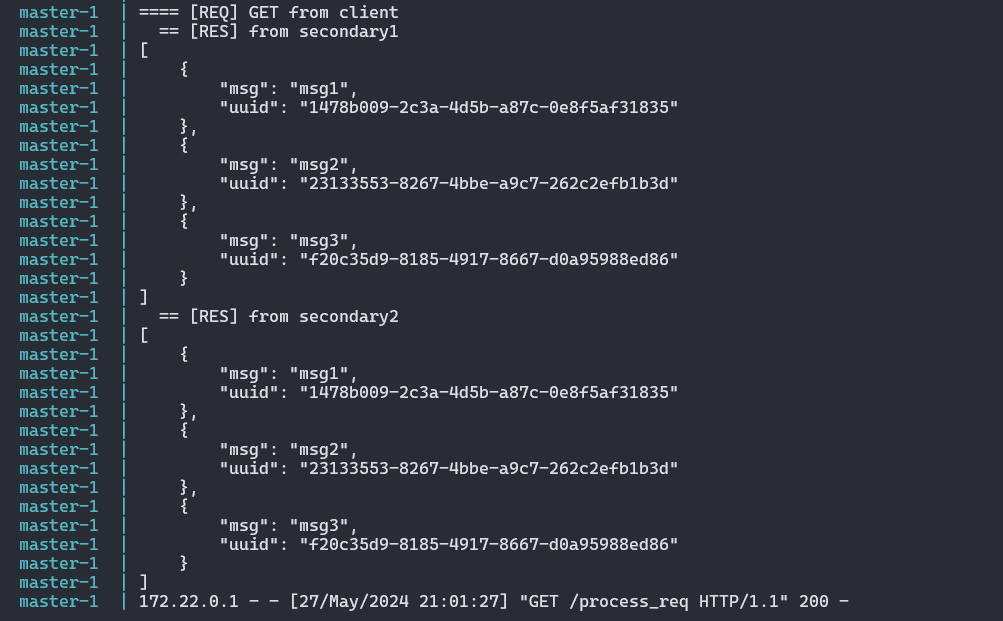
POST-запитом відправляються 3 повідомлення "msg1"-"msg3" на master. Master повертає поточне повідомлення зі згенерованим UUID.

GET-запитом клієнт зчитує всі збережені повідомлення.

Після цього, клієнт очікує відключення 1 екземпляра secondary.

## Вивід master

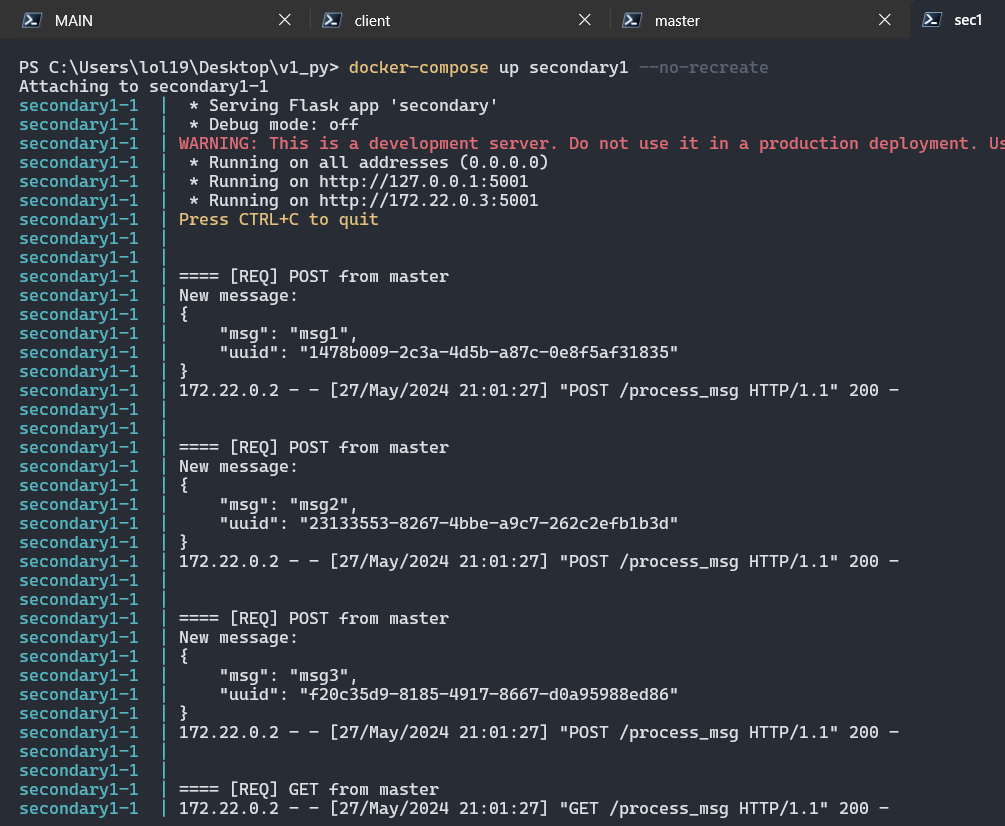




При надходженні POST-запиту з повідомленням, master генерує для нього UUID та відправляє дані на всі secondaries. Якщо хоча б від 1 secondary немає відповіді, то master надсилає клієнту повідомлення про помилку (реалізація blocking replication).

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

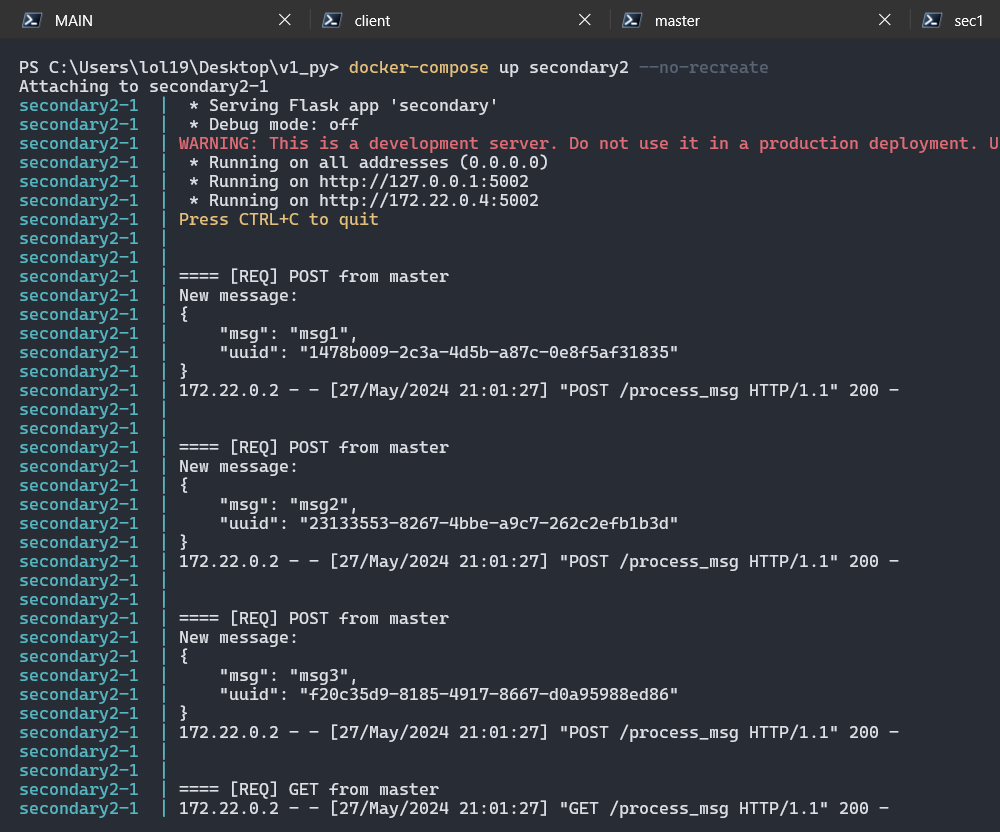
## Вивід secondary1



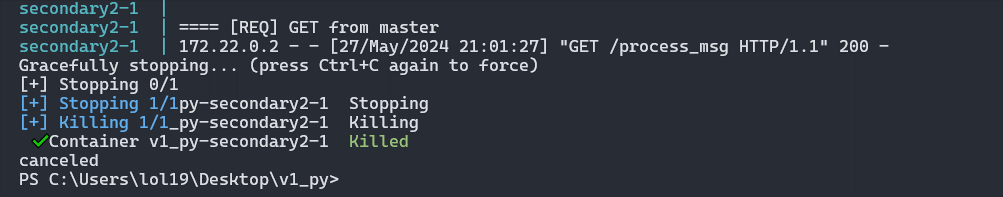
При надходженні POST-запиту, secondary додає повідомлення та його UUID в список.

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

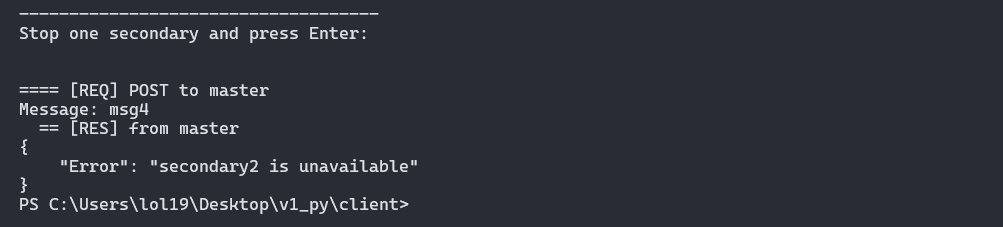
## Вивід secondary2



Тепер відключимо secondary2 (2x CTRL+C):

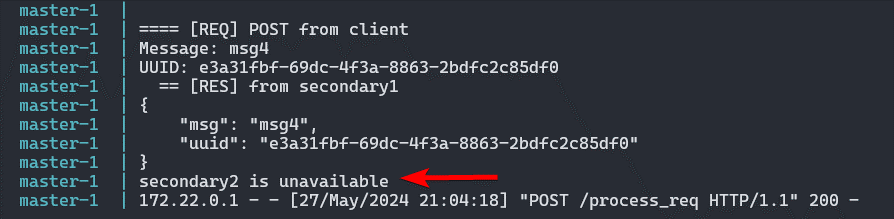


Повернемося до клієнта і натиснемо Enter:

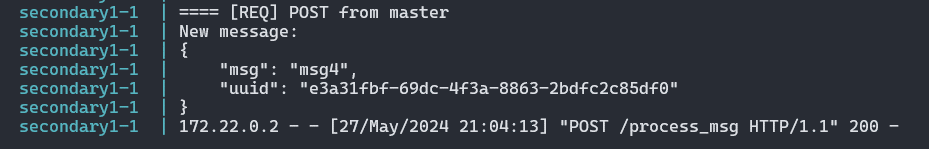


Реплікація завершується помилкою, яка вказує на те, що сервіс secondary2 недоступний. Така поведінка відповідає принципу blocking replication.

## Вивід master



## Вивід secondary1



Кожен сервіс логує всі події у файл service.log, що знаходиться в папці /app:

