1. Приклад запуску: (посилання на кращу якість скріншоту)

Тут в вікнах де вже щось запущено - це hazelcast ноди - аналогічним запуском, як в лабораторній роботі 2:

```
| Member | (172.17.0.3):5701 - d374c32f-2152-4469-a5f1-aaca38e79fc8 this | Member | (172.17.0.3):5701 - d374c32f-2152-4469-a5f1-aaca38e79fc8 this | Member | (172.17.0.3):5701 - d374c32f-2152-4469-a5f1-aaca38e79fc8 | (172.17.0.3):5701 - d374c32f-2152-4469-a5f1-aaca38e79fc8 | (172.17.0.3):5701 - d374c32f-2152-4469-a5f1-aaca38e79fc8 | (172.17.0.3):5701 | (dev) | (5.2.2) | (at safe) | (at sa
```

sudo docker run -p 5701:5701 hazelcast/hazelcast sudo docker run -p 5702:5701 hazelcast/hazelcast sudo docker run -p 5703:5701 hazelcast/hazelcast

В інших вікнах показано що буде запущено там - на всіх наступних скріншотах в кожному розділі раниться відповідне

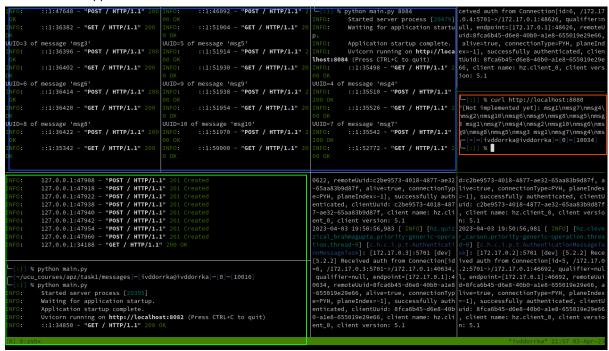
)

```
::1:47640 - "POST / HTTP/1.1" 20
                                                      ::1:46092 - "POST / HTTP/1.1"
         ::1:36382 - "GET / HTTP/1.1" 200
                                                      ::1:51904 - "GET / HTTP/1.1" 2
                                                                                                Waiting for application startu
                                                      ::1:51914 - "POST / HTTP/1.1"
         ::1:36396 - "POST / HTTP/1.1" 200
                                                                                                Uvicorn running on http://loca
                                                                                       lhost:8084 (Press CTRL+C to quit
                                                                                                 ::1:35498 - "GET / HTTP/1.1"
         ::1:36402 - "GET / HTTP/1.1" 200
                                                      ::1:51930 - "GET / HTTP/1.1" 2
UID=6 of message 'msg6'
                                            UUID=9 of message 'msg9'
                                                                                       UUID=4 of message 'msg4'
         ::1:36414 - "POST / HTTP/1.1" 20
                                                      ::1:51938 - "POST / HTTP/1.1"
                                                                                                ::1:35510 - "POST / HTTP/1.1"
         ::1:36420 - "GET / HTTP/1.1" 200
                                                      ::1:51954 - "GET / HTTP/1.1" 2
                                                                                                ::1:35526 - "GET / HTTP/1.1"
JUID=8 of message 'msg8'
[NFO: ::1:36422 - "POST / HTTP/1.1" 20
                                            UUID=10 of message 'msg10'
INFO: ::1:51970 - "POST / HTTP/1.1"
                                                                                                 ::1:35542 - "POST / HTTP/1.1"
```

(посилання на кращу якість фото

- 2.1 Запустили 3 екземпляри logging-service зліва направо відповідно порт 8081, 8083, 8084
- 2.2 Запустила <u>цей баш-файл</u> щоб запостити 10 повідомлень і скрін зверху показує як розподілились меседжі.

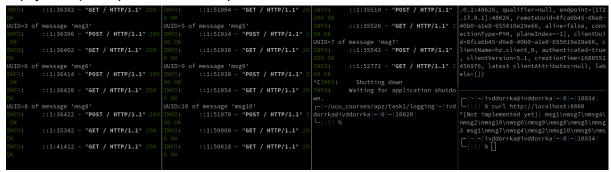
2.3 GET метод:



(посилання на кращу якість фото)

- В червоному віконечку всі меседжі виведені з усіх екземплярів logging-service
- В синьому бачимо як пройшов GET
- І відповідно в зеленому GET запрос прийшов до фасаду і месседж-сервісу

3. Якщо відключити один з екземплярів logging-service (я відключила той, що діє на порту 8084) і запросити GET ще раз:



(посилання на кращу якість фото)

Бачимо, що з того ендпоінта не прийшли повідомлення під час гет-запросу, тому відповідно якщо відключити ще один екземпляр logging-service - то під час гету матимемо очевидний результат - лише з працюючих ендпоінтів прийдуть повідомлення.