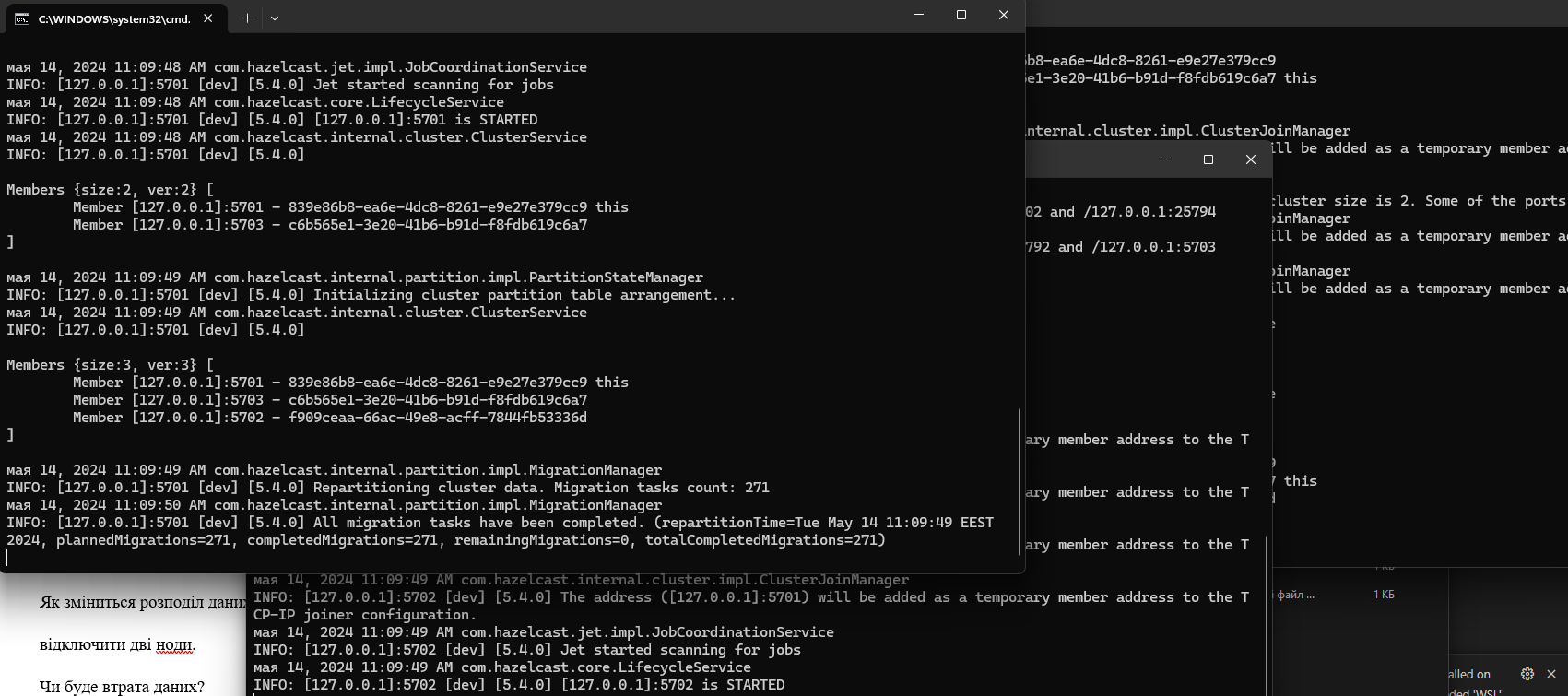
**Task 2**

**Розгортання і робота з distributed in-memory data structures на основі Hazelcast: Distributed Map**

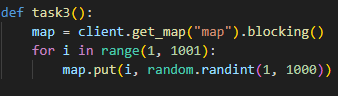
**Завдання:**

1. Встановити і налаштувати Hazelcast
2. Сконфігурувати і запустити 3 ноди (інстанси) об'єднані в кластер або як частину Java-застосування, або як окремі застосування

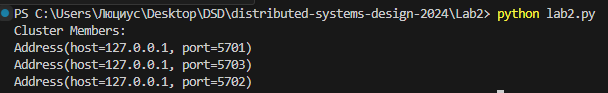


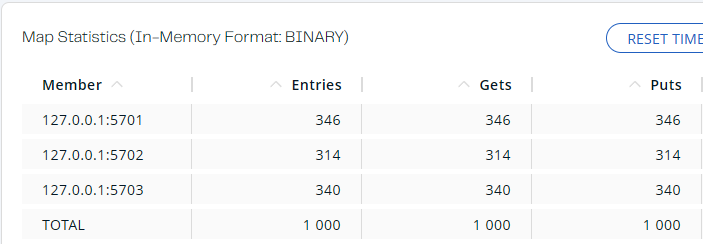
1. Продемонструйте роботу Distributed Map

Використовуючи API створіть Distributed Map запишіть в неї 1000 значень з ключем від 0 до 1



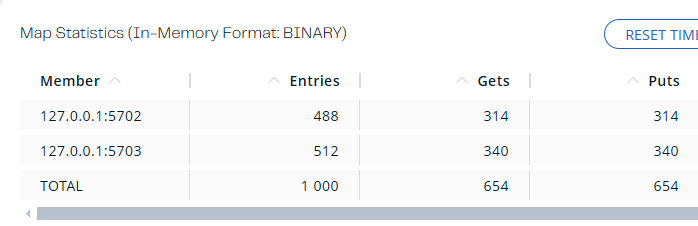
за допомогою Management Center подивиться на розподіл значень по нодах



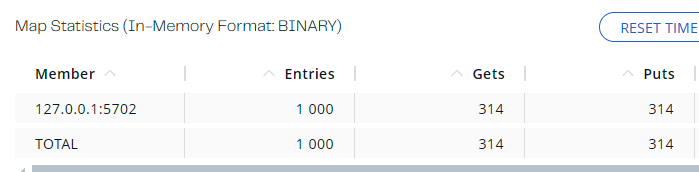


подивитись як зміниться розподіл даних по нодах:

якщо відключити одну ноду



відключити дві ноди.



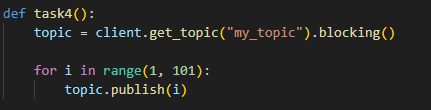
Чи буде втрата даних?

Втрат немає

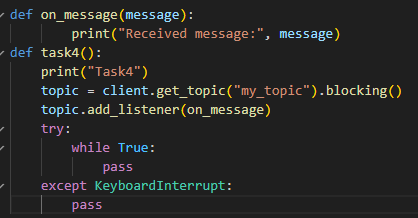
1. Продемонструйте роботу з Topic

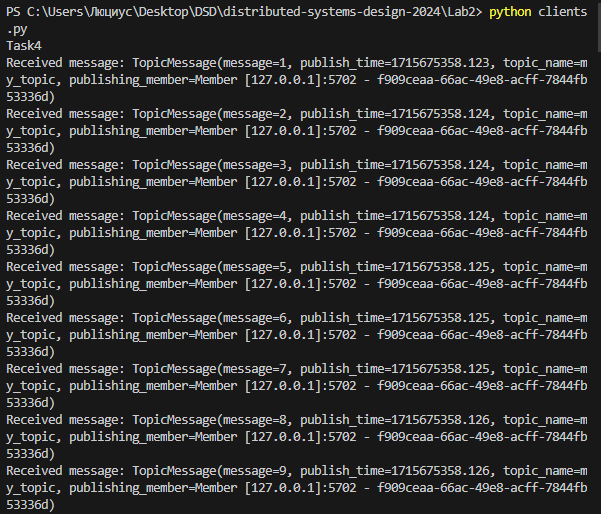
Запустіть одного клієнта який буде писати в Topic значення 1..100, а двох інших які будуть читати з Topic

Publisher



Readers





**яким чином будуть вичитуватись значення з Topic двома клієнтами?**

Коли повідомлення публікується у topic, воно реплікується на всі вузли кластера, де є підписники на цей topic.

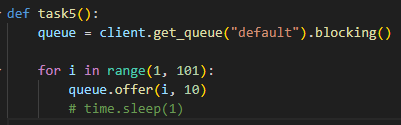
**якщо один з читачів буде певний час неактивний, чи отримає він повідомлення які він пропустив?**

Коли повідомлення публікується у topic, воно зберігається у внутрішній черзі на короткий час. Тому, якщо один з підписників topic був певний час неактивний, то він не отримає повідомлення, які були опубліковані під час його простою.

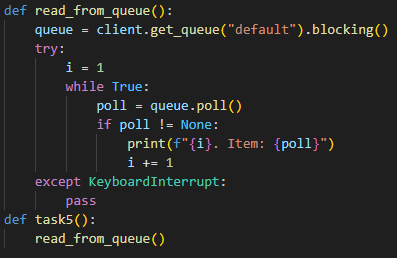
1. Робота з Bounded queue

На основі Distributed Queue налаштуйте Bounded queue на 10 елементів

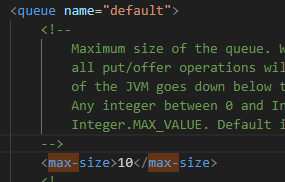
Publisher



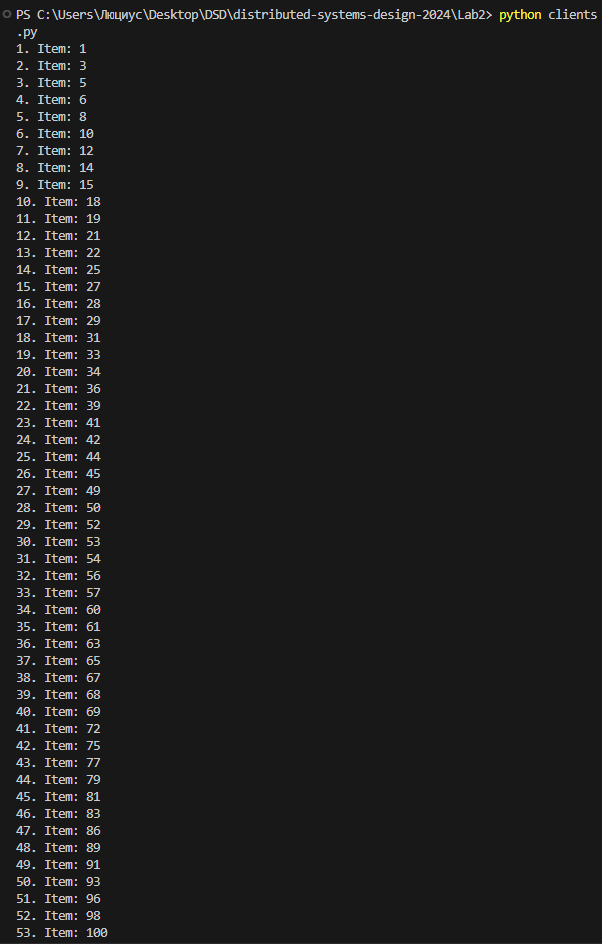
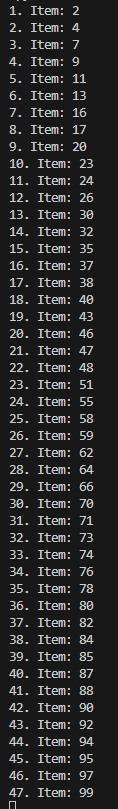
Readers



hazelcast.xml



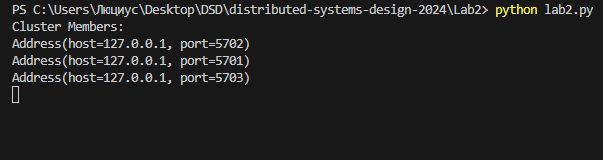
запустіть одного клієнта який буде писати в чергу значення 1..100, а двох інших які будуть читати з черги

**яким чином будуть вичитуватись значення з черги двома клієнтами?**

Вичитування буде відбуватися за принципом «хто встиг той з’їв», де один елемент черги забирається лише одним читачем.

**перевірте яка буде поведінка на запис якщо відсутнє читання, і черга заповнена**



Запис у чергу припиняється поки не звільниться місце.

GitHub:

<https://github.com/DruzDanil/distributed-systems-design-2024/tree/main/Lab2>