МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Проектування високонавантажених систем

Лабораторна робота №1

Реалізація лічильника з використанням Hazelcast

Перевірив:
Родіонов А. М.

Виконав:
студент І курсу
групи ФБ-41мп
Сахній Н. Р.

Київ 2024

Мета роботи: Імплементувати дуже простий функціонал, який є частиною соціальних мереж, відео-платформ, сервісів фото, тощо. Це є лічильник лайків/переглядів/ретвітів …, що збільшується на один при відповідній дії.

Завдання до виконання:

1. **Встановити і налаштувати Hazelcast 5.4.0 (**у новіших версіях частина необхідного для виконання завдань функціоналу є платною**).**
   - https://docs.hazelcast.com/hazelcast/5.4/getting-started/install-hazelcast
   - https://hazelcast.com/community-edition-projects/downloads/archives/#hazelcast-platform-5-4-0

1) `docker network create -d bridge hazelcast-network`

2) `pip install hazelcast-python-client==5.4.0`

3) `cat ./docker-compose.yml`

```yaml
version: "3.2"
services:
  # Hazelcast Node 1
  hazelcast-node1:
    container_name: 'massivecounter-node1'
    image: 'hazelcast/hazelcast:5.4.0'
    network_mode: 'hazelcast-network'
    environment:
      - HZ_NETWORK_PUBLICADDRESS=192.168.50.48:5701
      - HZ_CLUSTERNAME=massive-counter-cluster
      - HAZELCAST_CONFIG=/opt/hazelcast/cp_distmap-hazelcast.xml
    volumes:
      - ./cp_distmap-hazelcast.xml:/opt/hazelcast/cp_distmap-hazelcast.xml
    ports:
      - '5701:5701'

  # Hazelcast Node 2
  hazelcast-node2:
    container_name: 'massivecounter-node2'
    image: 'hazelcast/hazelcast:5.4.0'
    network_mode: 'hazelcast-network'
    environment:
      - HZ_NETWORK_PUBLICADDRESS=192.168.50.48:5702
      - HZ_CLUSTERNAME=massive-counter-cluster
      - HAZELCAST_CONFIG=/opt/hazelcast/cp_distmap-hazelcast.xml
    volumes:
      - ./cp_distmap-hazelcast.xml:/opt/hazelcast/cp_distmap-hazelcast.xml
    ports:
      - '5702:5701'

```

```
  # Hazelcast Node 3
  hazelcast-node3:
    container_name: 'massivecounter-node3'
    image: 'hazelcast/hazelcast:5.4.0'
    network_mode: 'hazelcast-network'
    environment:
      - HZ_NETWORK_PUBLICADDRESS=192.168.50.48:5703
      - HZ_CLUSTERNAME=massive-counter-cluster
      - HAZELCAST_CONFIG=/opt/hazelcast/cp_distmap-hazelcast.xml
    volumes:
      - ./cp_distmap-hazelcast.xml:/opt/hazelcast/cp_distmap-hazelcast.xml
    ports:
      - '5703:5701'

  # Management Center
  hazelcast-management:
    container_name: 'massivecounter-management-center'
    image: 'hazelcast/management-center:5.4.0'
    network_mode: 'hazelcast-network'
    depends_on:
      - hazelcast-node1
      - hazelcast-node2
      - hazelcast-node3
    ports:
      - '8080:8080'

```

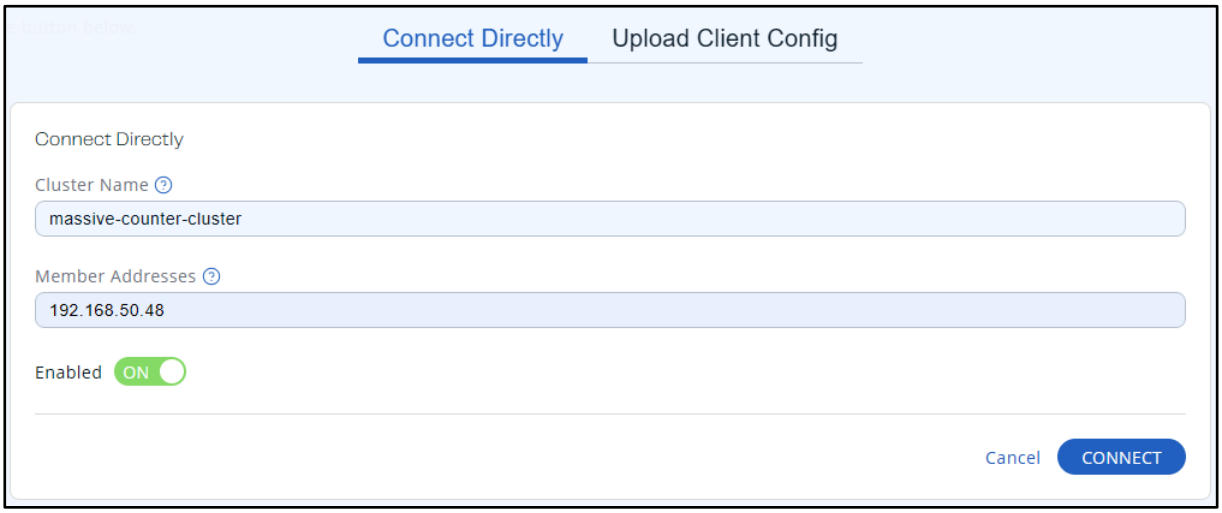
4) `docker-compose up`

```
t-1000@DESKTOP-DRIOPBB MINGW64 /d/KPI/5 курс Магістрат/Проектування ВС/Task_1-MassiveCounterIncrement
$ docker-compose up
Starting massivecounter-node1 ...
Starting massivecounter-node3 ...
Starting massivecounter-node2 ...
Starting massivecounter-node3 ... done
Starting massivecounter-node1 ... done
Starting massivecounter-node2 ... done
Creating massivecounter-management-center ...
Creating massivecounter-management-center ... done
Attaching to massivecounter-node3, massivecounter-node1, massivecounter-node2, massivecounter-management-center
massivecounter-node1    | ########################################
massivecounter-node1    | # JAVA=/usr/bin/java
massivecounter-node1    | # JAVA_OPTS=--add-modules java.se --add-exports java.base/jdk.internal.ref=ALL-UNNAMED
ens java.management/sun.management=ALL-UNNAMED --add-opens jdk.management/com.sun.management.internal=ALL-UNNAMED
4j2 -Dlog4j.configurationFile=file:/opt/hazelcast/config/log4j2.properties -Dhazelcast.config=/opt/hazelcast/cp_d
ue -XX:MaxRAMPercentage=80.0
massivecounter-node1    | # CLASSPATH=/opt/hazelcast/*:/opt/hazelcast/lib:/opt/hazelcast/lib/*:/opt/hazelcast/bin
massivecounter-node1    | ########################################
massivecounter-node2    | ########################################
massivecounter-node2    | # JAVA=/usr/bin/java
massivecounter-node2    | # JAVA_OPTS=--add-modules java.se --add-exports java.base/jdk.internal.ref=ALL-UNNAMED
ens java.management/sun.management=ALL-UNNAMED --add-opens jdk.management/com.sun.management.internal=ALL-UNNAMED
4j2 -Dlog4j.configurationFile=file:/opt/hazelcast/config/log4j2.properties -Dhazelcast.config=/opt/hazelcast/cp_d
ue -XX:MaxRAMPercentage=80.0
massivecounter-node2    | # CLASSPATH=/opt/hazelcast/*:/opt/hazelcast/lib:/opt/hazelcast/lib/*:/opt/hazelcast/bin
massivecounter-node2    | ########################################
massivecounter-node3    | ########################################
massivecounter-node3    | # JAVA=/usr/bin/java
massivecounter-node3    | # JAVA_OPTS=--add-modules java.se --add-exports java.base/jdk.internal.ref=ALL-UNNAMED
ens java.management/sun.management=ALL-UNNAMED --add-opens jdk.management/com.sun.management.internal=ALL-UNNAMED
4j2 -Dlog4j.configurationFile=file:/opt/hazelcast/config/log4j2.properties -Dhazelcast.config=/opt/hazelcast/cp_d
ue -XX:MaxRAMPercentage=80.0
massivecounter-node3    | # CLASSPATH=/opt/hazelcast/*:/opt/hazelcast/lib:/opt/hazelcast/lib/*:/opt/hazelcast/bin
massivecounter-node3    | ########################################
```

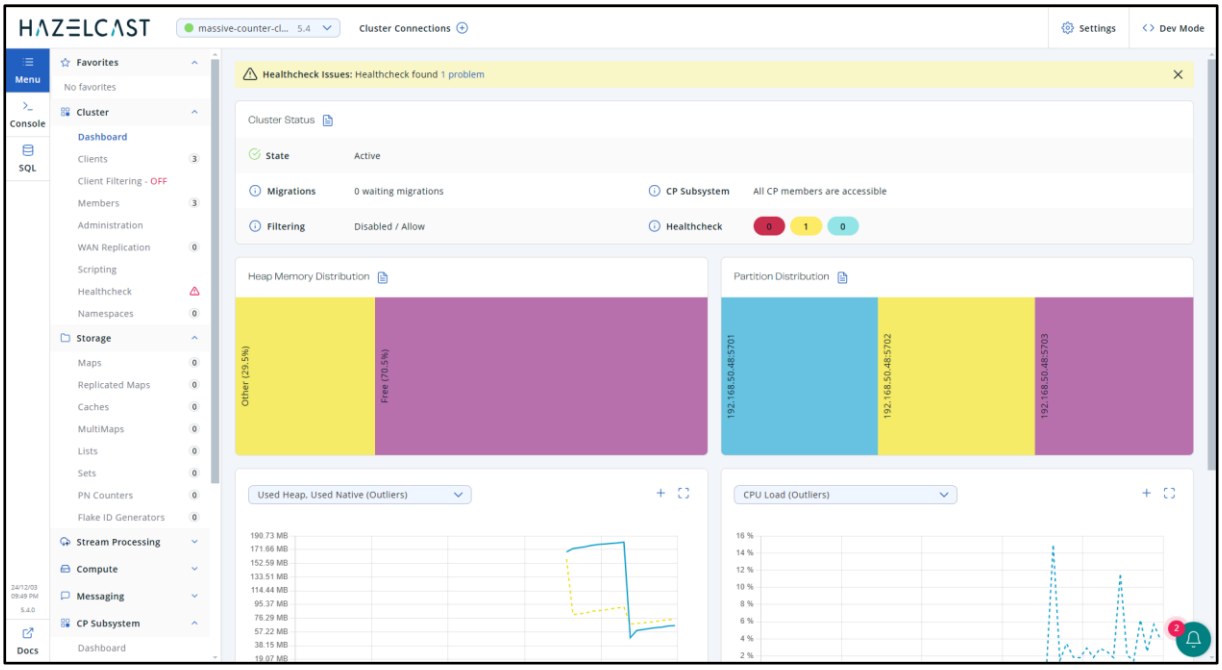
2. Сконфігурувати і запустити 3 ноди (інстанси) об'єднані в кластер або як частину Python-застосування, або як окремі застосування.

1) Open URL-link: http://localhost:8080/cluster-connections

2) Connect Cluster



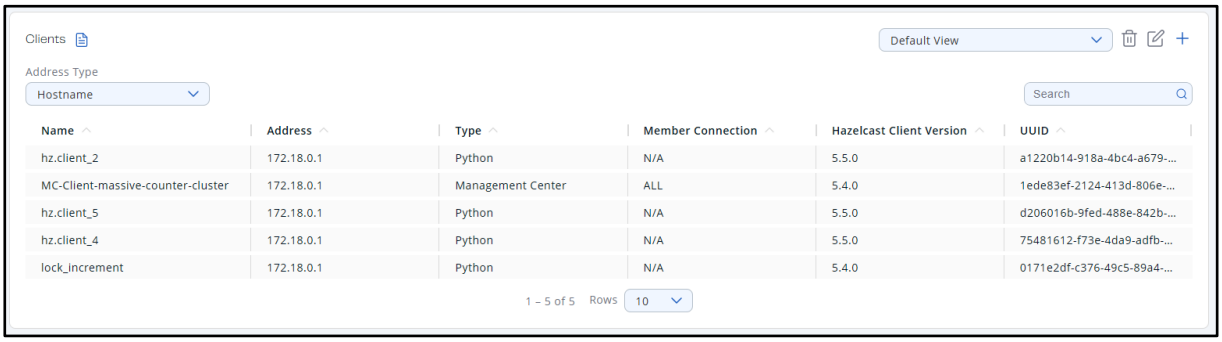
3) View Cluster



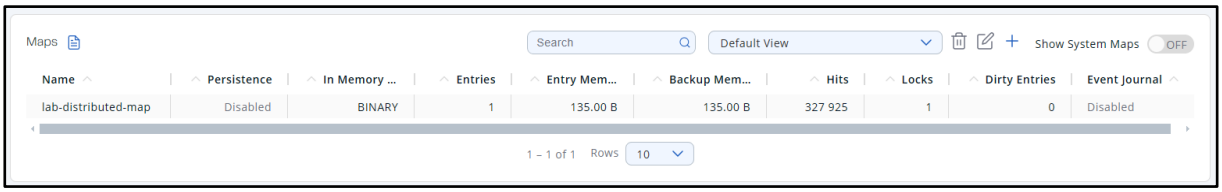
4) Snippet of Code

```python
if __name__ == "__main__":
    # Підключення до Hazelcast через Python-клієнта
    hz = hazelcast.HazelcastClient(cluster_name="massive-counter-cluster",
                                   client_name="lock_increment")
    distributed_map = hz.get_map("lab-distributed-map").blocking()
    print("Successful connection to Hazelcast!")
```

5) [MGTM.] Menu > Cluster > Clients



6) [MGTM.] Menu > Storage > Maps



3. **Далі, на основі прикладу з Distributed Map, напишіть код який буде емулювати інкремент значення для одного й того самого ключа у циклі до 10К.** Це необхідно робити у 10 потоках.

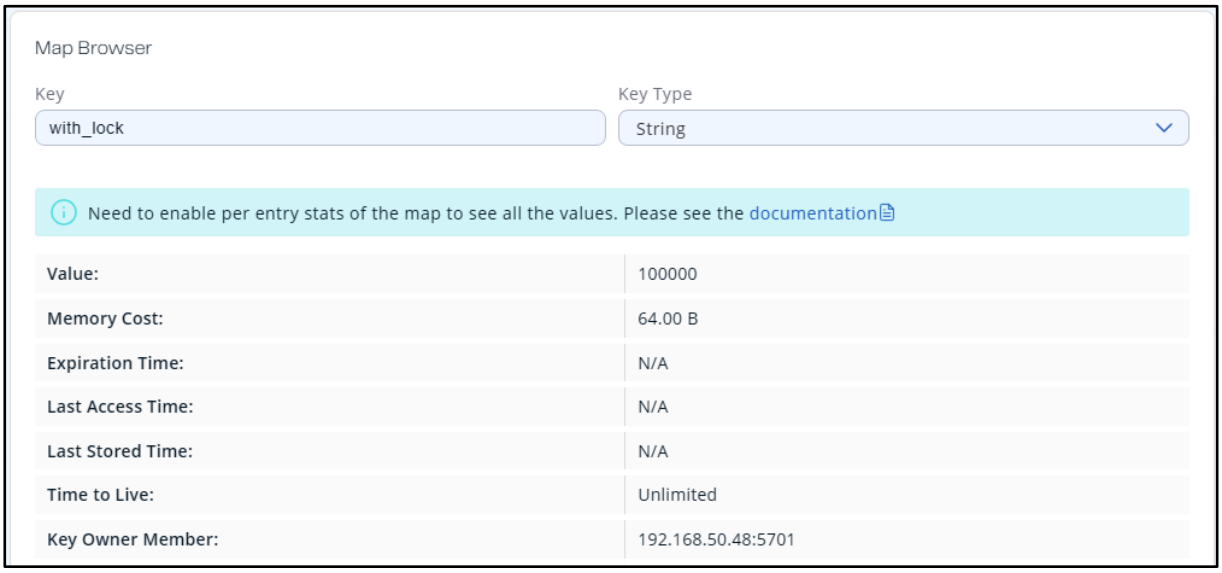
1) Snippet of Code – **with_lock**

```
# Функція для інкременту значення лічильника з використанням Lock для синхронізації
def with_lock(key):
    global main_counter
    thread_lock = threading.Lock()
    for _ in range(10000):
        with thread_lock:
            main_counter += 1
            distributed_map.put(key, main_counter)
```

2) Executing Result – **with_lock**

```
--------------------------------------------------
Executing 1. Lock for Increment...
--------------------------------------------------
Task '1. Lock for Increment' executed in: 34.67 seconds
Final counter value: 100000
--------------------------------------------------
```

3) Map Browser – **with_lock**



4. **На основі прикладу реалізуйте лічильник без блокувань.** Поміряйте час виконання, та подивиться чи коректне кінцеве значення лічильника ви отримаєте.

   - https://docs.hazelcast.com/imdg/latest/data-structures/map#locking-maps

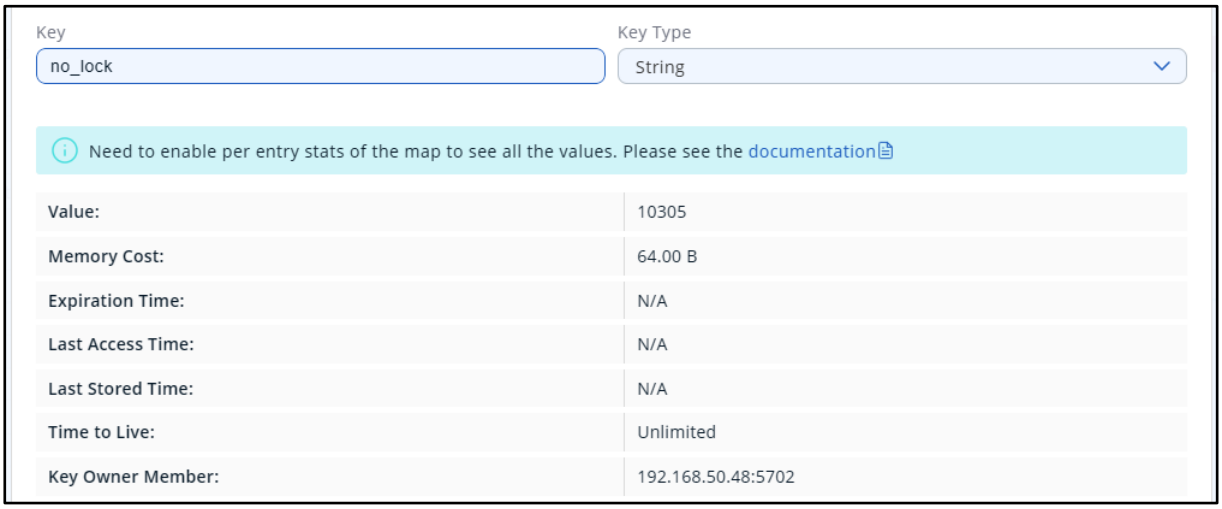
1) Snippet of Code – **no_lock**

```
# Функція для інкременту значення лічильника без Lock, з простим читанням і записом
def no_lock(key):
    distributed_map.put(key, 0)
    for _ in range(10000):
        counter = distributed_map.get(key) + 1
        distributed_map.put(key, counter)
```

2) Executing Result – **no_lock**

```
--------------------------------------------------
Executing 2. No Lock for Increment...
--------------------------------------------------
Task '2. No Lock for Increment' executed in: 56.46 seconds
Final counter value: 10305
--------------------------------------------------
```

3) Map Browser – **no_lock**



5. **На основі прикладу реалізуйте лічильник з використанням песимістичного блокування.** Поміряйте час виконання, та подивиться чи коректне кінцеве значення лічильника ви отримаєте.
   - https://docs.hazelcast.com/imdg/latest/data-structures/map#pessimistic-locking

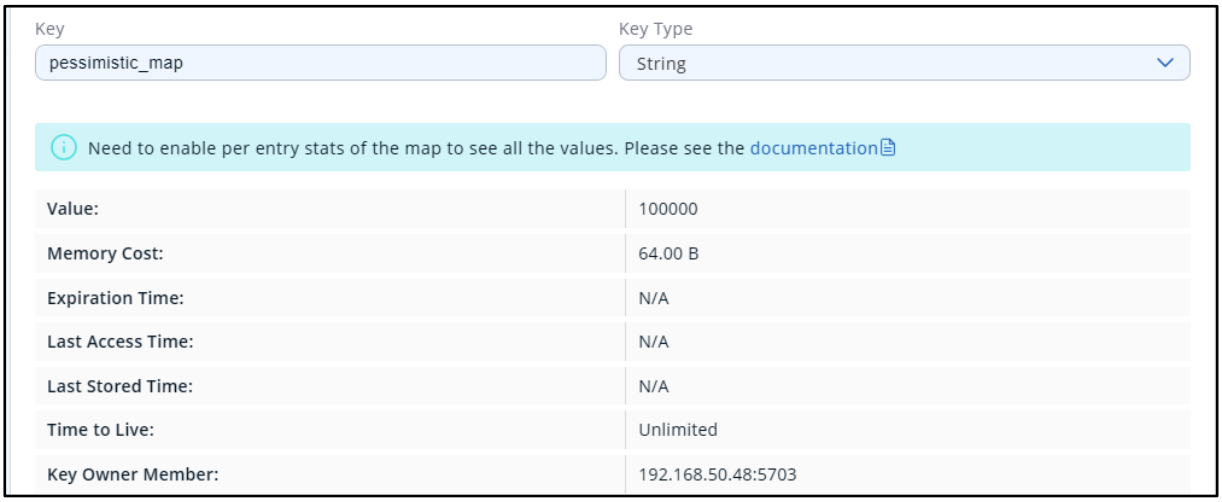
1) Snippet of Code – **pessimistic_map**

```python
# Функція для інкременту значення лічильника з використанням Lock-мапи (map.lock)
def pessimistic_map(key):
    if (not distributed_map.contains_key(key)): distributed_map.put(key,0)
    for _ in range(10000):
        distributed_map.lock(key)
        try:
            counter = distributed_map.get(key) + 1
            distributed_map.put(key, counter)
        finally:
            distributed_map.unlock(key)
```

2) Executing Result – **pessimistic_map**

```
--------------------------------------------------
Executing 3. Map Lock for Increment...
--------------------------------------------------
Task '3. Map Lock for Increment' executed in: 599.18 seconds
Final counter value: 100000
--------------------------------------------------
```

3) Map Browser – **pessimistic_map**



6. **На основі прикладу реалізуйте лічильник з використанням оптимістичного блокування.** Поміряйте час виконання, та подивиться чи коректне кінцеве значення лічильника ви отримаєте.
   - https://docs.hazelcast.com/imdg/latest/data-structures/map#optimistic-locking

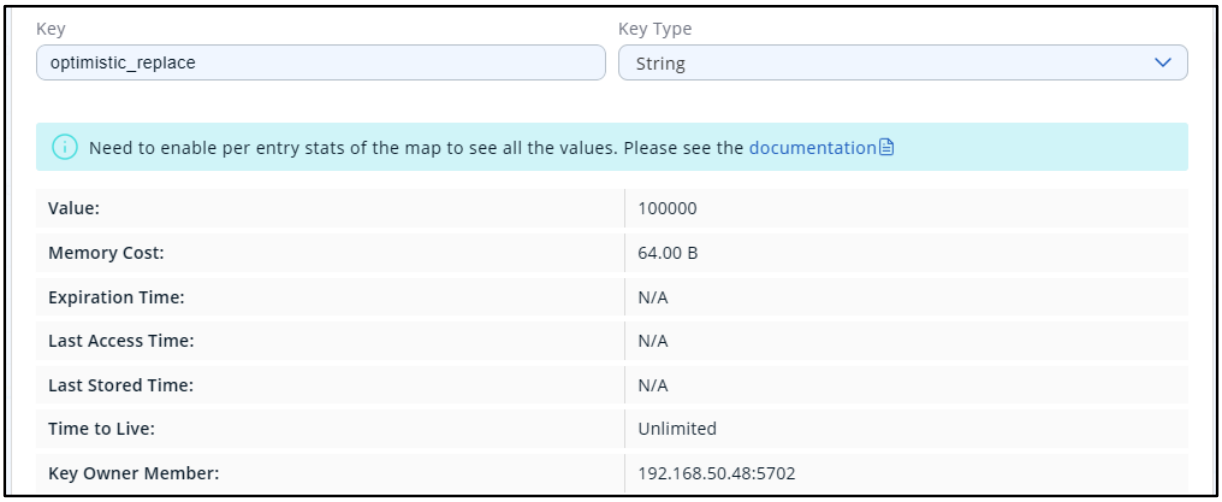
1) Snippet of Code – **optimistic_replace**

```python
# Функція для інкременту значення з використанням механізму заміни значень (replace_if_same)
def optimistic_replace(key):
    if (not distributed_map.contains_key(key)): distributed_map.put(key,0)
    for _ in range(10000):
        while True:
            oldcounter = distributed_map.get(key)
            newcounter = oldcounter + 1
            if distributed_map.replace_if_same(key, oldcounter, newcounter): break
```

2) Executing Result – **optimistic_replace**

```
--------------------------------------------------
Executing 4. Replace if Same Increment...
--------------------------------------------------
Task '4. Replace if Same Increment' executed in: 287.34 seconds
Final counter value: 100000
--------------------------------------------------
```

3) Map Browser – **optimistic_replace**



7. **Реалізуйте лічильник з використанням структури “*IAtomicLong*” та увімкнувши підтримку “*CP Sysbsystem*” на основі 3-ох нод.** Поміряйте час виконання, та подивиться чи коректне кінцеве значення лічильника ви отримаєте.

   - https://docs.hazelcast.com/hazelcast/5.4/data-structures/iatomiclong
   - https://docs.hazelcast.com/hazelcast/5.4/cp-subsystem/configuration

-1) ./cp_distmap-hazelcast.xml

```
▼<cp-subsystem>
   <!--  Кількість членів CP Subsystem для забезпечення консенсусу  -->
   <cp-member-count>3</cp-member-count>
   <!--  Має бути 3 ноди для Raft  -->
   <group-size>3</group-size>
 </cp-subsystem>
 <!--  Налаштування Distributed Map  -->
▼<map name="counter-map">
   <!--  Кількість резервних копій даних для здійснення реплікації  -->
   <backup-count>2</backup-count>
   <!--  Реплікація на 2 резервні ноди  -->
   <time-to-live-seconds>0</time-to-live-seconds>
   <max-idle-seconds>0</max-idle-seconds>
 </map>
```

0) [MGTM.] – Menu > CP Subsystem > Dashboard

1) Snippet of Code – **atomic_counter**

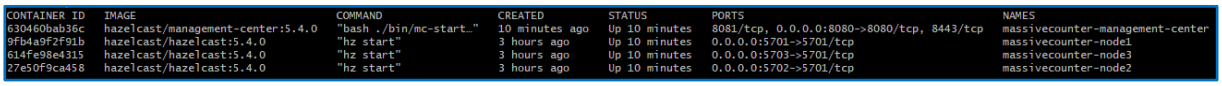
```python
# Функція для інкременту значення з використанням атомарного лічильника Hazelcast
def atomic_counter(key):
    count = hz.cp_subsystem.get_atomic_long(key).blocking()
    for _ in range(10000):
        counter = count.add_and_get(1)
        distributed_map.put(key, counter)
```

2) Executing Result – **atomic_counter**

```
--------------------------------------------------
Executing 5. Atomic Counter Increment...
--------------------------------------------------
Task '5. Atomic Counter Increment' executed in: 143.87 seconds
Final counter value: 100000
--------------------------------------------------
```

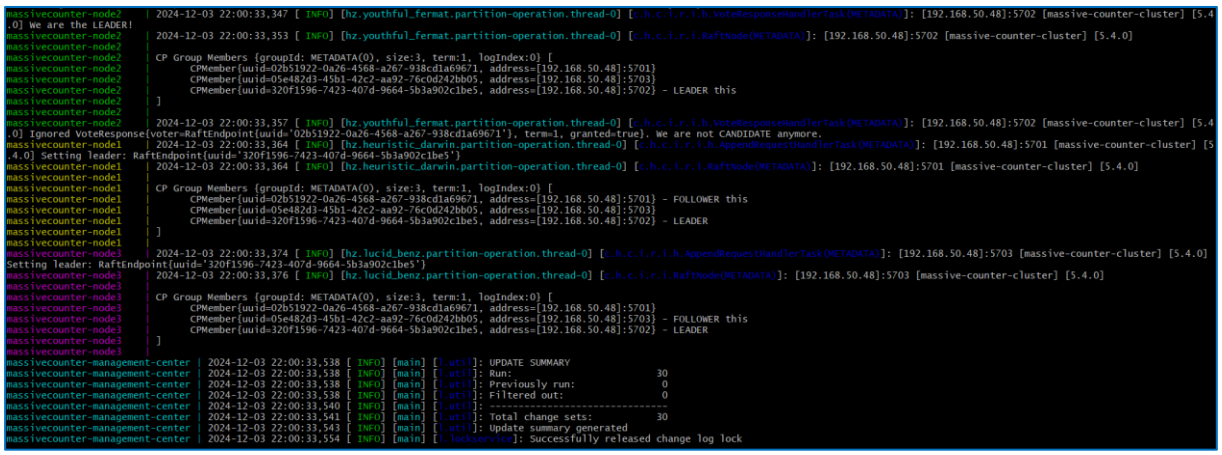
3) Atomic Longs – **atomic_counter**

* **Логи, які продукують ноди Hazelcast**
  * `docker ps`



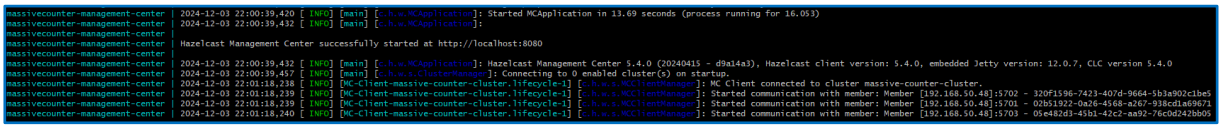
```
CONTAINER ID   IMAGE                                    COMMAND                  CREATED          STATUS          PORTS                                       NAMES
630460bab36c   hazelcast/management-center:5.4.0        "bash ./bin/mc-start…"   10 minutes ago   Up 10 minutes   8081/tcp, 0.0.0.0:8080->8080/tcp, 8443/tcp  massivecounter-management-center
9fb4a9f2f91b   hazelcast/hazelcast:5.4.0                "hz start"               3 hours ago      Up 10 minutes   0.0.0.0:5701->5701/tcp                      massivecounter-node1
614fe98e4315   hazelcast/hazelcast:5.4.0                "hz start"               3 hours ago      Up 10 minutes   0.0.0.0:5703->5701/tcp                      massivecounter-node3
27e50f9ca458   hazelcast/hazelcast:5.4.0                "hz start"               3 hours ago      Up 10 minutes   0.0.0.0:5702->5701/tcp                      massivecounter-node2
```

  * `docker-compose logs -f`



```
massivecounter-node2    | 2024-12-03 22:00:33,347 [ INFO] [hz.youthful_fermat.partition-operation.thread-0] [c.h.c.i.r.i.h.VoteResponseHandlerTask(METADATA)]: [192.168.50.48]:5702 [massive-counter-cluster] [5.4
.0] We are the LEADER!
massivecounter-node2    | 2024-12-03 22:00:33,353 [ INFO] [hz.youthful_fermat.partition-operation.thread-0] [c.h.c.i.r.i.RaftNode(METADATA)]: [192.168.50.48]:5702 [massive-counter-cluster] [5.4.0]
massivecounter-node2    |
massivecounter-node2    | CP Group Members {groupId: METADATA(0), size:3, term:1, logIndex:0} [
massivecounter-node2    |       CPMember{uuid=02b51922-0a26-4568-a267-938cd1a69671, address=[192.168.50.48]:5701}
massivecounter-node2    |       CPMember{uuid=05e482d3-45b1-42c2-aa92-76c0d242bb05, address=[192.168.50.48]:5703}
massivecounter-node2    |       CPMember{uuid=320f1596-7423-407d-9664-5b3a902c1be5, address=[192.168.50.48]:5702} - LEADER this
massivecounter-node2    | ]
massivecounter-node2    |
massivecounter-node2    | 2024-12-03 22:00:33,357 [ INFO] [hz.youthful_fermat.partition-operation.thread-0] [c.h.c.i.r.i.h.VoteResponseHandlerTask(METADATA)]: [192.168.50.48]:5702 [massive-counter-cluster] [5.4
.0] Ignored VoteResponse{voter=RaftEndpoint{uuid='02b51922-0a26-4568-a267-938cd1a69671'}, term=1, granted=true}. We are not CANDIDATE anymore.
massivecounter-node1    | 2024-12-03 22:00:33,364 [ INFO] [hz.heuristic_darwin.partition-operation.thread-0] [c.h.c.i.r.i.h.AppendRequestHandlerTask(METADATA)]: [192.168.50.48]:5701 [massive-counter-cluster] [5
.4.0] Setting leader: RaftEndpoint{uuid='320f1596-7423-407d-9664-5b3a902c1be5'}
massivecounter-node1    | 2024-12-03 22:00:33,364 [ INFO] [hz.heuristic_darwin.partition-operation.thread-0] [c.h.c.i.r.i.RaftNode(METADATA)]: [192.168.50.48]:5701 [massive-counter-cluster] [5.4.0]
massivecounter-node1    |
massivecounter-node1    | CP Group Members {groupId: METADATA(0), size:3, term:1, logIndex:0} [
massivecounter-node1    |       CPMember{uuid=02b51922-0a26-4568-a267-938cd1a69671, address=[192.168.50.48]:5701} - FOLLOWER this
massivecounter-node1    |       CPMember{uuid=05e482d3-45b1-42c2-aa92-76c0d242bb05, address=[192.168.50.48]:5703}
massivecounter-node1    |       CPMember{uuid=320f1596-7423-407d-9664-5b3a902c1be5, address=[192.168.50.48]:5702} - LEADER
massivecounter-node1    | ]
massivecounter-node1    |
massivecounter-node1    | 2024-12-03 22:00:33,374 [ INFO] [hz.lucid_benz.partition-operation.thread-0] [c.h.c.i.r.i.h.AppendRequestHandlerTask(METADATA)]: [192.168.50.48]:5703 [massive-counter-cluster] [5.4.0]
Setting leader: RaftEndpoint{uuid='320f1596-7423-407d-9664-5b3a902c1be5'}
massivecounter-node3    | 2024-12-03 22:00:33,376 [ INFO] [hz.lucid_benz.partition-operation.thread-0] [c.h.c.i.r.i.RaftNode(METADATA)]: [192.168.50.48]:5703 [massive-counter-cluster] [5.4.0]
massivecounter-node3    |
massivecounter-node3    | CP Group Members {groupId: METADATA(0), size:3, term:1, logIndex:0} [
massivecounter-node3    |       CPMember{uuid=02b51922-0a26-4568-a267-938cd1a69671, address=[192.168.50.48]:5701}
massivecounter-node3    |       CPMember{uuid=05e482d3-45b1-42c2-aa92-76c0d242bb05, address=[192.168.50.48]:5703} - FOLLOWER this
massivecounter-node3    |       CPMember{uuid=320f1596-7423-407d-9664-5b3a902c1be5, address=[192.168.50.48]:5702} - LEADER
massivecounter-node3    | ]
massivecounter-node3    |
massivecounter-management-center | 2024-12-03 22:00:33,538 [ INFO] [main] [c.u.util]: UPDATE SUMMARY
massivecounter-management-center | 2024-12-03 22:00:33,538 [ INFO] [main] [c.u.util]: Run:                    30
massivecounter-management-center | 2024-12-03 22:00:33,538 [ INFO] [main] [c.u.util]: Previously run:           0
massivecounter-management-center | 2024-12-03 22:00:33,538 [ INFO] [main] [c.u.util]: Filtered out:            0
massivecounter-management-center | 2024-12-03 22:00:33,540 [ INFO] [main] [c.u.util]: -------------------------
massivecounter-management-center | 2024-12-03 22:00:33,541 [ INFO] [main] [c.u.util]: Total change sets:      30
massivecounter-management-center | 2024-12-03 22:00:33,543 [ INFO] [main] [c.u.util]: Update summary generated
massivecounter-management-center | 2024-12-03 22:00:33,554 [ INFO] [main] [c.u.lockservice]: Successfully released change log lock
```

  * `cat ./cluster_nodes.log | grep -i "management-center"`



```
massivecounter-management-center | 2024-12-03 22:00:39,420 [ INFO] [main] [c.h.w.MCApplication]: Started MCApplication in 13.69 seconds (process running for 16.053)
massivecounter-management-center | 2024-12-03 22:00:39,432 [ INFO] [main] [c.h.w.MCApplication]:
massivecounter-management-center |
massivecounter-management-center | Hazelcast Management Center successfully started at http://localhost:8080
massivecounter-management-center |
massivecounter-management-center | 2024-12-03 22:00:39,432 [ INFO] [main] [c.h.w.MCApplication]: Hazelcast Management Center 5.4.0 (20240415 - d9a14a3), Hazelcast client version: 5.4.0, embedded Jetty version: 12.0.7, CLC version 5.4.0
massivecounter-management-center | 2024-12-03 22:00:39,457 [ INFO] [main] [c.h.w.s.ClusterManager]: Connecting to 0 enabled cluster(s) on startup.
massivecounter-management-center | 2024-12-03 22:01:18,238 [ INFO] [MC-Client-massive-counter-cluster.lifecycle-1] [c.h.w.s.MCClientManager]: MC Client connected to cluster massive-counter-cluster.
massivecounter-management-center | 2024-12-03 22:01:18,239 [ INFO] [MC-Client-massive-counter-cluster.lifecycle-1] [c.h.w.s.MCClientManager]: Started communication with member: Member [192.168.50.48]:5702 - 320f1596-7423-407d-9664-5b3a902c1be5
massivecounter-management-center | 2024-12-03 22:01:18,239 [ INFO] [MC-Client-massive-counter-cluster.lifecycle-1] [c.h.w.s.MCClientManager]: Started communication with member: Member [192.168.50.48]:5701 - 02b51922-0a26-4568-a267-938cd1a69671
massivecounter-management-center | 2024-12-03 22:01:18,240 [ INFO] [MC-Client-massive-counter-cluster.lifecycle-1] [c.h.w.s.MCClientManager]: Started communication with member: Member [192.168.50.48]:5703 - 05e482d3-45b1-42c2-aa92-76c0d242bb05
```

  * `cat ./cluster_nodes.log | grep -i "CP Subsystem"`

```
massivecounter-node3    | 2024-12-03 22:00:23,691 [ INFO] [main] [c.h.c.i.CPSubsystem]: [192.168.50.48]:5703 [massive-counter-cluster] [5.4.0] CP Subsystem is enabled with 3 members.
massivecounter-node3    | 2024-12-03 22:00:30,080 [ INFO] [hz.lucid_benz.cached.thread-2] [c.h.c.i.MetadataRaftGroupManager]: [192.168.50.48]:5703 [massive-counter-cluster] [5.4.0] CP Subsy
stem is waiting for 3 members to join the cluster. Current member count: 2
massivecounter-node3    | 2024-12-03 22:00:34,024 [ INFO] [hz.lucid_benz.cached.thread-2] [c.h.c.i.MetadataRaftGroupManager]: [192.168.50.48]:5703 [massive-counter-cluster] [5.4.0] CP Subsy
stem is initialized with: [CPMember{uuid=02b51922-0a26-4568-a267-938cd1a69671, address=[192.168.50.48]:5701}, CPMember{uuid=05e482d3-45b1-42c2-aa92-76c0d242bb05, address=[192.168.50.48]:570
3}, CPMember{uuid=320f1596-7423-407d-9664-5b3a902c1be5, address=[192.168.50.48]:5702}]
massivecounter-node2    | 2024-12-03 22:00:23,687 [ INFO] [main] [c.h.c.i.CPSubsystem]: [192.168.50.48]:5702 [massive-counter-cluster] [5.4.0] CP Subsystem is enabled with 3 members.
massivecounter-node2    | 2024-12-03 22:00:34,023 [ INFO] [hz.youthful_fermat.cached.thread-3] [c.h.c.i.MetadataRaftGroupManager]: [192.168.50.48]:5702 [massive-counter-cluster] [5.4.0] CP
Subsystem is initialized with: [CPMember{uuid=02b51922-0a26-4568-a267-938cd1a69671, address=[192.168.50.48]:5701}, CPMember{uuid=05e482d3-45b1-42c2-aa92-76c0d242bb05, address=[192.168.50.48
]:5703}, CPMember{uuid=320f1596-7423-407d-9664-5b3a902c1be5, address=[192.168.50.48]:5702}]
massivecounter-node1    | 2024-12-03 22:00:23,684 [ INFO] [main] [c.h.c.i.CPSubsystem]: [192.168.50.48]:5701 [massive-counter-cluster] [5.4.0] CP Subsystem is enabled with 3 members.
massivecounter-node1    | 2024-12-03 22:00:30,097 [ INFO] [hz.heuristic_darwin.cached.thread-5] [c.h.c.i.MetadataRaftGroupManager]: [192.168.50.48]:5701 [massive-counter-cluster] [5.4.0] CP
 Subsystem is waiting for 3 members to join the cluster. Current member count: 2
massivecounter-node1    | 2024-12-03 22:00:34,026 [ INFO] [hz.heuristic_darwin.cached.thread-5] [c.h.c.i.MetadataRaftGroupManager]: [192.168.50.48]:5701 [massive-counter-cluster] [5.4.0] CP
 Subsystem is initialized with: [CPMember{uuid=02b51922-0a26-4568-a267-938cd1a69671, address=[192.168.50.48]:5701}, CPMember{uuid=05e482d3-45b1-42c2-aa92-76c0d242bb05, address=[192.168.50.4
8]:5703}, CPMember{uuid=320f1596-7423-407d-9664-5b3a902c1be5, address=[192.168.50.48]:5702}]
```