МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра Інформаційної Безпеки

Проектування розподілених систем
Лабораторна робота №4
Мікросервіси з використанням Messaging queue

Виконав:
студент V курсу
групи ФБ-41мп
Африканський О. М.

Київ 2025

Мета: навчитися розгортати та конфігурувати кластер Hazelcast у системі мікросервісів з використанням черги повідомлень, реалізувати взаємодію між сервісами через розподілену чергу у конфігурації producer/consumer, забезпечити можливість парального запуску кількох екземплярів сервісів та перевірити механізми реплікації і відмовостійкості кластера при втраті одного з вузлів.

Хід роботи

Розглянемо всі функції на прикладі:
Спочатку запускаємо кластери kafka та hazelcast:

```
user@ubuntu:~/Documents/2_sem/dist_sys/4$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS                                                  NAMES
93cb95955f88   hazelcast/management-center:5.4.0    "bash ./bin/mc-start…"   28 minutes ago   Up 28 minutes   8081/tcp, 8443/tcp, 0.0.0.0:8000->8080/tcp, [::]:8000->8080/tcp   distmap-management-center
44a51a1772de   hazelcast/hazelcast:5.4.0            "hz start"               28 minutes ago   Up 28 minutes   0.0.0.0:5702->5701/tcp, [::]:5702->5701/tcp             hz-2
5ea2280a0c98   hazelcast/hazelcast:5.4.0            "hz start"               28 minutes ago   Up 28 minutes   0.0.0.0:5703->5701/tcp, [::]:5703->5701/tcp             hz-3
4033c6bc8ac5   hazelcast/hazelcast:5.4.0            "hz start"               28 minutes ago   Up 28 minutes   0.0.0.0:5701->5701/tcp, :::5701->5701/tcp               hz-1
aca3c5102b62   apache/kafka:latest                  "/__cacert_entrypoin…"   2 hours ago      Up 2 hours      9092/tcp, 0.0.0.0:8092->8091/tcp, [::]:8092->8091/tcp   kafka-01
bc7c970a91e2   apache/kafka:latest                  "/__cacert_entrypoin…"   2 hours ago      Up 2 hours      9092/tcp, 0.0.0.0:8093->8091/tcp, [::]:8093->8091/tcp   kafka-02
```

Тепер запускаємо усі сервіси:

- config-service

```
user@ubuntu:~/Documents/2_sem/dist_sys/4/config-service$ go run main.go
2025/06/21 21:24:39 Config server running on port 7201
```

- facade-service

```
user@ubuntu:~/Documents/2_sem/dist_sys/4/facade-service$ go run main.go
2025/06/21 21:09:40 Starting facade-service on port 8080
```

- message-service

```
user@ubuntu:~/Documents/2_sem/dist_sys/4/messages-service$ go run main.go -port 8081
2025/06/21 21:17:15 Starting messages-service on port 8081
```

```
user@ubuntu:~/Documents/2_sem/dist_sys/4/messages-service$ go run main.go -port 8082
2025/06/21 21:17:24 Starting messages-service on port 8082
```

- logging-service

```
user@ubuntu:~/Documents/2_sem/dist_sys/4/logging-service$ go run main.go -port 8201 -hazelcast-addresses 127.0.0.1:5701 -map dev-map

2025/06/21 21:22:42 INFO : trying to connect to cluster: dev-map
2025/06/21 21:22:42 INFO : connected to cluster: dev-map
2025/06/21 21:22:42 INFO :

Members {size:3, ver:3} [
        Member 172.18.0.1:5701 - 06c320d8-b99a-44fe-adb2-9291031b913a
        Member 172.18.0.1:5703 - 35c5abf6-d8ce-46df-84d3-a1d8f8614b0b
        Member 172.18.0.1:5702 - 69c6eb08-ee45-404f-9b22-1c33813856c5
]

Successfully connected to Hazelcast map 'dev-map' at addresses [127.0.0.1:5701]
Starting logging-service on port :8201
```

```
user@ubuntu:~/Documents/2_sem/dist_sys/4/logging-service$ go run main.go -port 8202 -hazelcast-addresses 127.0.0.1:5702 -map dev-map

2025/06/21 21:23:35 INFO : trying to connect to cluster: dev-map
2025/06/21 21:23:35 INFO : connected to cluster: dev-map
2025/06/21 21:23:35 INFO :

Members {size:3, ver:3} [
        Member 172.18.0.1:5701 - 06c320d8-b99a-44fe-adb2-9291031b913a
        Member 172.18.0.1:5703 - 35c5abf6-d8ce-46df-84d3-a1d8f8614b0b
        Member 172.18.0.1:5702 - 69c6eb08-ee45-404f-9b22-1c33813856c5
]

Successfully connected to Hazelcast map 'dev-map' at addresses [127.0.0.1:5702]
Starting logging-service on port :8202
```

```
user@ubuntu:~/Documents/2_sem/dist_sys/4/logging-service$ go run main.go -port 8203 -hazelcast-addresses 127.0.0.1:5703 -map dev-map

2025/06/21 21:24:07 INFO : trying to connect to cluster: dev-map
2025/06/21 21:24:07 INFO : connected to cluster: dev-map
2025/06/21 21:24:07 INFO :

Members {size:3, ver:3} [
        Member 172.18.0.1:5701 - 06c320d8-b99a-44fe-adb2-9291031b913a
        Member 172.18.0.1:5703 - 35c5abf6-d8ce-46df-84d3-a1d8f8614b0b
        Member 172.18.0.1:5702 - 69c6eb08-ee45-404f-9b22-1c33813856c5
]

Successfully connected to Hazelcast map 'dev-map' at addresses [127.0.0.1:5703]
Starting logging-service on port :8203
```

Як можна побачити, всі сервіси увімкнулись нормально.

Перевірка отримання конфігурації з config-service:

```
user@ubuntu:~/Documents/2_sem/dist_sys/4$ curl http://localhost:7201/logging-service
{"addresses":["localhost:8201","localhost:8202","localhost:8203"]}
user@ubuntu:~/Documents/2_sem/dist_sys/4$ curl http://localhost:7201/messages-service
{"kafka_brokers":["localhost:8092","localhost:8093"]}
```

Перейдемо до демонстрації роботи кластеру:

Запускаємо bash скрипт, який створю 10 запитів на facade-service та записую десять повідомлень. Переглянемо вивід усіх сервісів:

консоль

```
user@ubuntu:~/Documents/2_sem/dist_sys/4$ for i in {1..10}; do curl -X POST -H "Content-Type: application/json" -d "{\"id\":\"msg$i\",\"msg\":\"Message $i\"}" http://localhos
t:8080/message; echo; done
Message sent with ID: bd7cdf67-8434-4df9-9ed1-8b7299bb6058 to logging-service (localhost:8202) and messages-service (localhost:8082)
Message sent with ID: 3600e7ec-001d-488d-ae49-534a1d0dcd43 to logging-service (localhost:8202) and messages-service (localhost:8082)
Message sent with ID: c2db9e08-ccf2-4b47-823b-7efd56781057 to logging-service (localhost:8203) and messages-service (localhost:8082)
Message sent with ID: f04c00cd-d193-41b7-82b8-abe667fb5a22 to logging-service (localhost:8202) and messages-service (localhost:8081)
Message sent with ID: d5db8d13-dc6d-4b1a-8ea0-60c7c4d8c717 to logging-service (localhost:8201) and messages-service (localhost:8081)
Message sent with ID: c5e0ade0-dafb-449d-a119-6221592ef8f0 to logging-service (localhost:8201) and messages-service (localhost:8082)
Message sent with ID: 479ff179-93fc-4a88-ac8f-c0aba41981cf to logging-service (localhost:8203) and messages-service (localhost:8081)
Message sent with ID: 2802f530-8ccd-4578-a26b-86b52cc68bcf to logging-service (localhost:8202) and messages-service (localhost:8082)
Message sent with ID: 4a5719b3-3a0c-4a9f-87bb-c59087c718db to logging-service (localhost:8203) and messages-service (localhost:8082)
Message sent with ID: 3603c419-d332-499a-b376-519f1bcb36f3 to logging-service (localhost:8202) and messages-service (localhost:8081)
```

config-service

```
2025/06/22 13:22:10 Returning logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:10 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:10 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:11 Returning logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:11 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:11 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:12 Returning logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:12 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:12 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:13 Returning logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:13 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:13 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:14 Returning logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:14 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:14 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:15 Returning logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:15 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:15 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:16 Returning logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:16 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:16 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:17 Returning logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:17 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:17 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:18 Returning logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:18 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
2025/06/22 13:22:18 Returning messages-service addresses: [localhost:8081 localhost:8082], KafkaBrokers: [localhost:8092 localhost:8093]
```

facade-service

```
2025/06/22 13:22:10 Attempting to get addresses for logging-service from http://localhost:7201/logging-service
2025/06/22 13:22:10 Raw response body for logging-service: {"addresses":["localhost:8201","localhost:8202","localhost:8203"]}
2025/06/22 13:22:10 Extracted logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:10 Attempting to get addresses for messages-service from http://localhost:7201/messages-service
2025/06/22 13:22:10 Raw response body for messages-service: {"addresses":["localhost:8081","localhost:8082"],"kafka_brokers":["localhost:8092","localhost:8093"]}
2025/06/22 13:22:10 Extracted messages-service addresses: [localhost:8081 localhost:8082]
2025/06/22 13:22:10 Attempt 1: Sending POST request to logging-service http://localhost:8202/log with payload: {"id":"bd7cdf67-8434-4df9-9ed1-8b7299bb6058","msg":"{\"id\":\"msg1\",\"msg\":\"Message 1\"}"}
2025/06/22 13:22:10 Successfully sent message to logging-service localhost:8202 on attempt 1
2025/06/22 13:22:10 Attempt 1: Sending POST request to messages-service http://localhost:8082/message with payload: {"id":"bd7cdf67-8434-4df9-9ed1-8b7299bb6058","msg":"{\"id\":\"msg1\",\"msg\":\"Message 1\"}"}
2025/06/22 13:22:11 Successfully sent message to messages-service localhost:8082 on attempt 1
2025/06/22 13:22:11 Attempting to get addresses for logging-service from http://localhost:7201/logging-service
2025/06/22 13:22:11 Raw response body for logging-service: {"addresses":["localhost:8201","localhost:8202","localhost:8203"]}
2025/06/22 13:22:11 Extracted logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
2025/06/22 13:22:11 Attempting to get addresses for messages-service from http://localhost:7201/messages-service
2025/06/22 13:22:11 Raw response body for messages-service: {"addresses":["localhost:8081","localhost:8082"],"kafka_brokers":["localhost:8092","localhost:8093"]}
2025/06/22 13:22:11 Extracted messages-service addresses: [localhost:8081 localhost:8082]
2025/06/22 13:22:11 Attempt 1: Sending POST request to logging-service http://localhost:8202/log with payload: {"id":"3600e7ec-001d-488d-ae49-534a1d0dcd43","msg":"{\"id\":\"msg2\",\"msg\":\"Message 2\"}"}
2025/06/22 13:22:11 Successfully sent message to logging-service localhost:8202 on attempt 1
2025/06/22 13:22:11 Attempt 1: Sending POST request to messages-service http://localhost:8082/message with payload: {"id":"3600e7ec-001d-488d-ae49-534a1d0dcd43","msg":"{\"id\":\"msg2\",\"msg\":\"Message 2\"}"}
2025/06/22 13:22:12 Successfully sent message to messages-service localhost:8082 on attempt 1
2025/06/22 13:22:12 Attempting to get addresses for logging-service from http://localhost:7201/logging-service
2025/06/22 13:22:12 Raw response body for logging-service: {"addresses":["localhost:8201","localhost:8202","localhost:8203"]}
2025/06/22 13:22:12 Extracted logging-service addresses: [localhost:8201 localhost:8202 localhost:8203]
```

message-service

```
2025/06/22 13:22:14 Message written to Kafka
2025/06/22 13:22:15 Message written to Kafka
2025/06/22 13:22:17 Message written to Kafka
2025/06/22 13:22:20 Message written to Kafka
```

```
2025/06/22 13:22:11 Message written to Kafka
2025/06/22 13:22:12 Message written to Kafka
2025/06/22 13:22:13 Message written to Kafka
2025/06/22 13:22:14 Received message: ID=bd7cdf67-8434-4df9-9ed1-8b7299bb6058, Msg={"id":"msg1","msg":"Message 1"}
2025/06/22 13:22:14 Received message: ID=3600e7ec-001d-488d-ae49-534a1d0dcd43, Msg={"id":"msg2","msg":"Message 2"}
2025/06/22 13:22:14 Received message: ID=c2db9e08-ccf2-4b47-823b-7efd56781057, Msg={"id":"msg3","msg":"Message 3"}
2025/06/22 13:22:16 Message written to Kafka
2025/06/22 13:22:18 Message written to Kafka
2025/06/22 13:22:19 Message written to Kafka
2025/06/22 13:22:23 Received message: ID=f04c00cd-d193-41b7-82b8-abe667fb5a22, Msg={"id":"msg4","msg":"Message 4"}
2025/06/22 13:22:23 Received message: ID=d5db8d13-dc6d-4b1a-8ea0-60c7c4d8c717, Msg={"id":"msg5","msg":"Message 5"}
2025/06/22 13:22:23 Received message: ID=c5e0ade0-dafb-449d-a119-6221592ef8f0, Msg={"id":"msg6","msg":"Message 6"}
2025/06/22 13:22:23 Received message: ID=479ff179-93fc-4a88-ac8f-c0aba41981cf, Msg={"id":"msg7","msg":"Message 7"}
2025/06/22 13:22:23 Received message: ID=2802f530-8ccd-4578-a26b-86b52cc68bcf, Msg={"id":"msg8","msg":"Message 8"}
2025/06/22 13:22:23 Received message: ID=4a5719b3-3a0c-4a9f-87bb-c59087c718db, Msg={"id":"msg9","msg":"Message 9"}
2025/06/22 13:22:23 Received message: ID=3603c419-d332-499a-b376-519f1bcb36f3, Msg={"id":"msg10","msg":"Message 10"}
```

logging-service

```
Received message: ID=d5db8d13-dc6d-4b1a-8ea0-60c7c4d8c717, Msg={"id":"msg5","msg":"Message 5"}. Already exists or successfully added to map.
Received message: ID=c5e0ade0-dafb-449d-a119-6221592ef8f0, Msg={"id":"msg6","msg":"Message 6"}. Already exists or successfully added to map.
```

```
Received message: ID=bd7cdf67-8434-4df9-9ed1-8b7299bb6058, Msg={"id":"msg1","msg":"Message 1"}. Already exists or successfully added to map.
Received message: ID=3600e7ec-001d-488d-ae49-534a1d0dcd43, Msg={"id":"msg2","msg":"Message 2"}. Already exists or successfully added to map.
Received message: ID=f04c00cd-d193-41b7-82b8-abe667fb5a22, Msg={"id":"msg4","msg":"Message 4"}. Already exists or successfully added to map.
Received message: ID=2802f530-8ccd-4578-a26b-86b52cc68bcf, Msg={"id":"msg8","msg":"Message 8"}. Already exists or successfully added to map.
Received message: ID=3603c419-d332-499a-b376-519f1bcb36f3, Msg={"id":"msg10","msg":"Message 10"}. Already exists or successfully added to map.
```

```
Received message: ID=c2db9e08-ccf2-4b47-823b-7efd56781057, Msg={"id":"msg3","msg":"Message 3"}. Already exists or successfully added to map.
Received message: ID=479ff179-93fc-4a88-ac8f-c0aba41981cf, Msg={"id":"msg7","msg":"Message 7"}. Already exists or successfully added to map.
Received message: ID=4a5719b3-3a0c-4a9f-87bb-c59087c718db, Msg={"id":"msg9","msg":"Message 9"}. Already exists or successfully added to map.
```

Тепер отримаємо всі повідомлення:

```
user@ubuntu:~/Documents/2_sem/dist_sys/4$ curl http://localhost:8080/messages
Messages from logging-service (localhost:8201):
{"id":"msg1","msg":"Message 1"}
{"id":"msg7","msg":"Message 7"}
{"id":"msg2","msg":"Message 2"}
{"id":"msg5","msg":"Message 5"}
{"id":"msg6","msg":"Message 6"}
{"id":"msg9","msg":"Message 9"}
{"id":"msg3","msg":"Message 3"}
{"id":"msg10","msg":"Message 10"}
{"id":"msg4","msg":"Message 4"}
{"id":"msg8","msg":"Message 8"}

Messages from messages-service (localhost:8082):
{"id":"bd7cdf67-8434-4df9-9ed1-8b7299bb6058","msg":"{\"id\":\"msg1\",\"msg\":\"Message 1\"}"}
{"id":"3600e7ec-001d-488d-ae49-534a1d0dcd43","msg":"{\"id\":\"msg2\",\"msg\":\"Message 2\"}"}
{"id":"c2db9e08-ccf2-4b47-823b-7efd56781057","msg":"{\"id\":\"msg3\",\"msg\":\"Message 3\"}"}
{"id":"f04c00cd-d193-41b7-82b8-abe667fb5a22","msg":"{\"id\":\"msg4\",\"msg\":\"Message 4\"}"}
{"id":"d5db8d13-dc6d-4b1a-8ea0-60c7c4d8c717","msg":"{\"id\":\"msg5\",\"msg\":\"Message 5\"}"}
{"id":"c5e0ade0-dafb-449d-a119-6221592ef8f0","msg":"{\"id\":\"msg6\",\"msg\":\"Message 6\"}"}
{"id":"479ff179-93fc-4a88-ac8f-c0aba41981cf","msg":"{\"id\":\"msg7\",\"msg\":\"Message 7\"}"}
{"id":"2802f530-8ccd-4578-a26b-86b52cc68bcf","msg":"{\"id\":\"msg8\",\"msg\":\"Message 8\"}"}
{"id":"4a5719b3-3a0c-4a9f-87bb-c59087c718db","msg":"{\"id\":\"msg9\",\"msg\":\"Message 9\"}"}
{"id":"3603c419-d332-499a-b376-519f1bcb36f3","msg":"{\"id\":\"msg10\",\"msg\":\"Message 10\"}"}
```

Представлений вивід curl http://localhost:8080/messages демонструє цікаві відмінності у поведінці двох сервісів щодо порядку повідомлень, що цілком відповідає природі їхніх базових технологій.

З боку logging-service, який використовує кластер Hazelcast, ми бачимо, що повідомлення відображаються не в тому порядку, в якому вони, були відправлені. Повідомлення йдуть у, здавалося б, випадковій послідовності (наприклад, 1, 7, 2, 5...). Це є типовою поведінкою для розподілених сховищ ключ-значення, таких як Hazelcast Map.

Натомість, вивід від messages-service, що базується на Kafka, демонструє чіткий послідовний порядок повідомлень. Ми бачимо, що повідомлення відображаються рівно у тій послідовності, в якій вони, були відправлені (Message 1, Message 2, ..., Message 10). Ця властивість є ключовою особливістю Kafka: він є розподіленим журналом комітів, який гарантує порядок повідомлень.

Тепер спробуємо вимкнути одну ноду kafka та отримати дані знову:

```
user@ubuntu:~/Documents/2_sem/dist_sys/4$ sudo docker compose -f kafka/compose.yml ps
[sudo] password for user:
NAME       IMAGE                 COMMAND                  SERVICE    CREATED        STATUS        PORTS
kafka-01   apache/kafka:latest   "/__cacert_entrypoin…"   kafka-01   20 hours ago   Up 20 hours   9092/tcp, 0.0.0.0:8092->8091/tcp, [::]:8092->8091/tcp
kafka-02   apache/kafka:latest   "/__cacert_entrypoin…"   kafka-02   20 hours ago   Up 20 hours   9092/tcp, 0.0.0.0:8093->8091/tcp, [::]:8093->8091/tcp
user@ubuntu:~/Documents/2_sem/dist_sys/4$ sudo docker compose -f kafka/compose.yml stop kafka-02
[+] Stopping 1/1
 ✔ Container kafka-02  Stopped
user@ubuntu:~/Documents/2_sem/dist_sys/4$ sudo docker compose -f kafka/compose.yml ps
NAME       IMAGE                 COMMAND                  SERVICE    CREATED        STATUS        PORTS
kafka-01   apache/kafka:latest   "/__cacert_entrypoin…"   kafka-01   20 hours ago   Up 20 hours   9092/tcp, 0.0.0.0:8092->8091/tcp, [::]:8092->8091/tcp
```

```
kafka-01  | [2025-06-22 13:26:27,942] INFO [RaftManager id=1] Election timed out before receiving sufficient vote responses to become candidate. Current epoch election state:
 EpochElection(voterStates={1=VoterState(replicaKey=ReplicaKey(id=1, directoryId=<undefined>), state=GRANTED), 2=VoterState(replicaKey=ReplicaKey(id=2, directoryId=<undefined
>), state=UNRECORDED)}) (org.apache.kafka.raft.KafkaRaftClient)
kafka-01  | [2025-06-22 13:26:27,943] INFO [RaftManager id=1] Attempting durable transition to UnattachedState(epoch=5, leaderId=OptionalInt.empty, votedKey=Optional.empty, v
oters=[1, 2], electionTimeoutMs=0, highWatermark=Optional.empty) from ProspectiveState(epoch=5, leaderId=OptionalInt.empty, retries=1, votedKey=Optional.empty, epochElection=
EpochElection(voterStates={1=VoterState(replicaKey=ReplicaKey(id=1, directoryId=<undefined>), state=GRANTED), 2=VoterState(replicaKey=ReplicaKey(id=2, directoryId=<undefined>
), state=UNRECORDED)}), electionTimeoutMs=1752, highWatermark=Optional.empty) (org.apache.kafka.raft.QuorumState)
kafka-01  | [2025-06-22 13:26:27,945] INFO [RaftManager id=1] Completed transition to UnattachedState(epoch=5, leaderId=OptionalInt.empty, votedKey=Optional.empty, voters=[1,
 2], electionTimeoutMs=0, highWatermark=Optional.empty) from ProspectiveState(epoch=5, leaderId=OptionalInt.empty, retries=1, votedKey=Optional.empty, epochElection=EpochElec
tion(voterStates={1=VoterState(replicaKey=ReplicaKey(id=1, directoryId=<undefined>), state=GRANTED), 2=VoterState(replicaKey=ReplicaKey(id=2, directoryId=<undefined>), state=
UNRECORDED)}), electionTimeoutMs=1752, highWatermark=Optional.empty) (org.apache.kafka.raft.QuorumState)
kafka-01  | [2025-06-22 13:26:27,945] INFO [RaftManager id=1] Completed transition to ProspectiveState(epoch=5, leaderId=OptionalInt.empty, retries=1, votedKey=Optional.empty
, epochElection=EpochElection(voterStates={1=VoterState(replicaKey=ReplicaKey(id=1, directoryId=<undefined>), state=GRANTED), 2=VoterState(replicaKey=ReplicaKey(id=2, directo
ryId=<undefined>), state=UNRECORDED)}), electionTimeoutMs=1416, highWatermark=Optional.empty) from UnattachedState(epoch=5, leaderId=OptionalInt.empty, votedKey=Optional.empt
y, voters=[1, 2], electionTimeoutMs=0, highWatermark=Optional.empty) (org.apache.kafka.raft.QuorumState)
kafka-01  | [2025-06-22 13:26:28,130] WARN [RaftManager id=1] Error connecting to node kafka-02:8093 (id: 2 rack: null isFenced: false) (org.apache.kafka.clients.NetworkClien
t)
kafka-01  | java.net.UnknownHostException: kafka-02
kafka-01  |     at java.base/java.net.InetAddress$CachedLookup.get(Unknown Source) ~[?:?]
kafka-01  |     at java.base/java.net.InetAddress.getAllByName0(Unknown Source) ~[?:?]
kafka-01  |     at java.base/java.net.InetAddress.getAllByName(Unknown Source) ~[?:?]
kafka-01  |     at org.apache.kafka.clients.DefaultHostResolver.resolve(DefaultHostResolver.java:27) ~[kafka-clients-4.0.0.jar:?]
kafka-01  |     at org.apache.kafka.clients.ClientUtils.resolve(ClientUtils.java:125) ~[kafka-clients-4.0.0.jar:?]
kafka-01  |     at org.apache.kafka.clients.ClusterConnectionStates$NodeConnectionState.resolveAddresses(ClusterConnectionStates.java:536) ~[kafka-clients-4.0.0.jar:?]
kafka-01  |     at org.apache.kafka.clients.ClusterConnectionStates$NodeConnectionState.currentAddress(ClusterConnectionStates.java:511) ~[kafka-clients-4.0.0.jar:?]
kafka-01  |     at org.apache.kafka.clients.ClusterConnectionStates.currentAddress(ClusterConnectionStates.java:173) ~[kafka-clients-4.0.0.jar:?]
kafka-01  |     at org.apache.kafka.clients.NetworkClient.initiateConnect(NetworkClient.java:1140) [kafka-clients-4.0.0.jar:?]
kafka-01  |     at org.apache.kafka.clients.NetworkClient.ready(NetworkClient.java:368) [kafka-clients-4.0.0.jar:?]
kafka-01  |     at org.apache.kafka.server.util.InterBrokerSendThread.sendRequests(InterBrokerSendThread.java:146) [kafka-server-common-4.0.0.jar:?]
kafka-01  |     at org.apache.kafka.server.util.InterBrokerSendThread.pollOnce(InterBrokerSendThread.java:109) [kafka-server-common-4.0.0.jar:?]
kafka-01  |     at org.apache.kafka.server.util.InterBrokerSendThread.doWork(InterBrokerSendThread.java:137) [kafka-server-common-4.0.0.jar:?]
kafka-01  |     at org.apache.kafka.server.util.ShutdownableThread.run(ShutdownableThread.java:136) [kafka-server-common-4.0.0.jar:?]
```

Отримуємо дані:

```
user@ubuntu:~/Documents/2_sem/dist_sys/4$ curl http://localhost:8080/messages
Messages from logging-service (localhost:8203):
{"id":"msg1","msg":"Message 1"}
{"id":"msg7","msg":"Message 7"}
{"id":"msg2","msg":"Message 2"}
{"id":"msg5","msg":"Message 5"}
{"id":"msg6","msg":"Message 6"}
{"id":"msg9","msg":"Message 9"}
{"id":"msg3","msg":"Message 3"}
{"id":"msg10","msg":"Message 10"}
{"id":"msg4","msg":"Message 4"}
{"id":"msg8","msg":"Message 8"}

Messages from messages-service (localhost:8082):
{"id":"bd7cdf67-8434-4df9-9ed1-8b7299bb6058","msg":"{\"id\":\"msg1\",\"msg\":\"Message 1\"}"}
{"id":"3600e7ec-001d-488d-ae49-534a1d0dcd43","msg":"{\"id\":\"msg2\",\"msg\":\"Message 2\"}"}
{"id":"c2db9e08-ccf2-4b47-823b-7efd56781057","msg":"{\"id\":\"msg3\",\"msg\":\"Message 3\"}"}
{"id":"f04c00cd-d193-41b7-82b8-abe667fb5a22","msg":"{\"id\":\"msg4\",\"msg\":\"Message 4\"}"}
{"id":"d5db8d13-dc6d-4b1a-8ea0-60c7c4d8c717","msg":"{\"id\":\"msg5\",\"msg\":\"Message 5\"}"}
{"id":"c5e0ade0-dafb-449d-a119-6221592ef8f0","msg":"{\"id\":\"msg6\",\"msg\":\"Message 6\"}"}
{"id":"479ff179-93fc-4a88-ac8f-c0aba41981cf","msg":"{\"id\":\"msg7\",\"msg\":\"Message 7\"}"}
{"id":"2802f530-8ccd-4578-a26b-86b52cc68bcf","msg":"{\"id\":\"msg8\",\"msg\":\"Message 8\"}"}
{"id":"4a5719b3-3a0c-4a9f-87bb-c59087c718db","msg":"{\"id\":\"msg9\",\"msg\":\"Message 9\"}"}
{"id":"3603c419-d332-499a-b376-519f1bcb36f3","msg":"{\"id\":\"msg10\",\"msg\":\"Message 10\"}"}
```

Як можна побачити, ми отримали всі дані у повному обсязі.

Висновки: Ця лабораторна робота успішно продемонструвала реалізацію мікросервісної архітектури, де facade-service ефективно інтегрує дані з logging-service (на базі Hazelcast) та messages-service (на базі Kafka). Аналіз виводу підтвердив очікувану поведінку: Hazelcast забезпечує розподілене зберігання без гарантії порядку повідомлень, тоді як Kafka підтримує їх чітку послідовність. Робота підтвердила функціональність та взаємодію компонентів розподіленої системи.