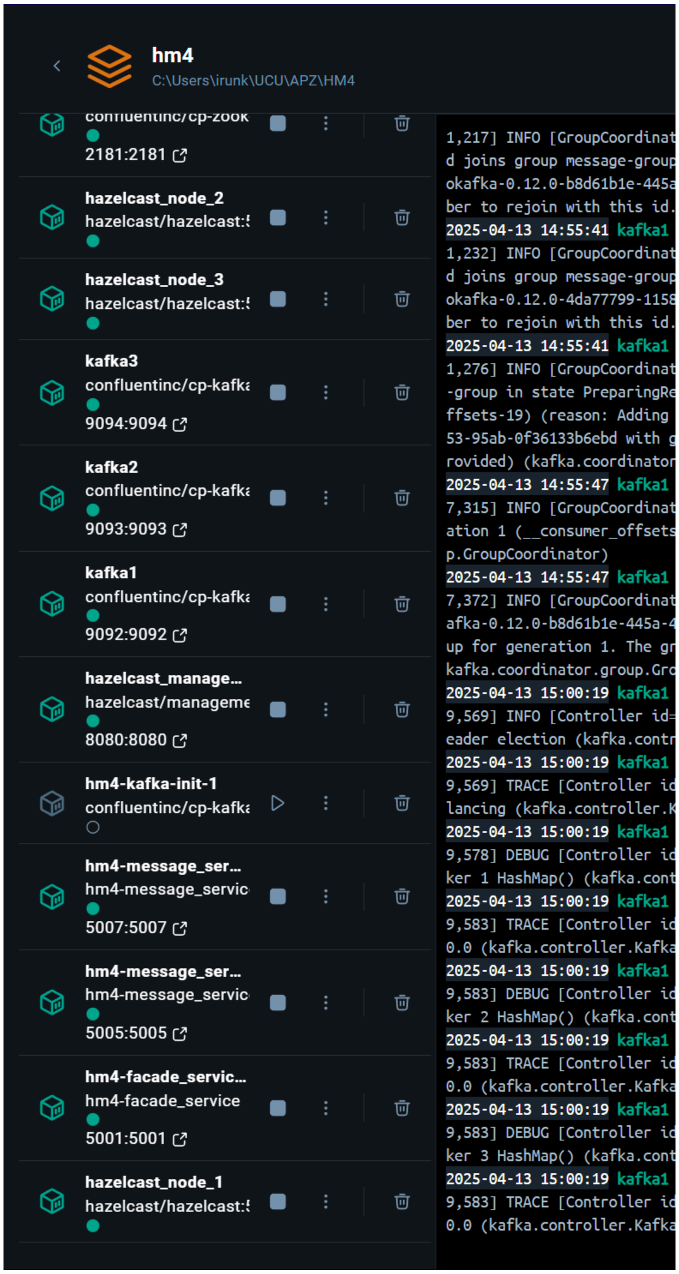
Homework 4

So basically I just added 3 `kafka` (you can see that in the `.yml` file). There is also a `kafka-init`, which is not needed for our task generally, but it creates a topic and defining `kafka1:9092` as a leader. Then it just stops.

Everything can be started from the `yml` file with the command: `docker compose up --build` .

Main Task

To check if everything works correctly, I need to start it and weight till all containers will be ready:



hm4

C:\Users\irunk\UCU\APZ\HM4

confluentinc/cp-zook

2181:2181

hazelcast_node_2

hazelcast/hazelcast:!

hazelcast_node_3

hazelcast/hazelcast:!

kafka3

confluentinc/cp-kafka

9094:9094

kafka2

confluentinc/cp-kafka

9093:9093

kafka1

confluentinc/cp-kafka

9092:9092

hazelcast_manage...

hazelcast/manageme

8080:8080

hm4-kafka-init-1

confluentinc/cp-kafka

hm4-message_ser...

hm4-message_servic

5007:5007

hm4-message_ser...

hm4-message_servic

5005:5005

hm4-facade_servic...

hm4-facade_service

5001:5001

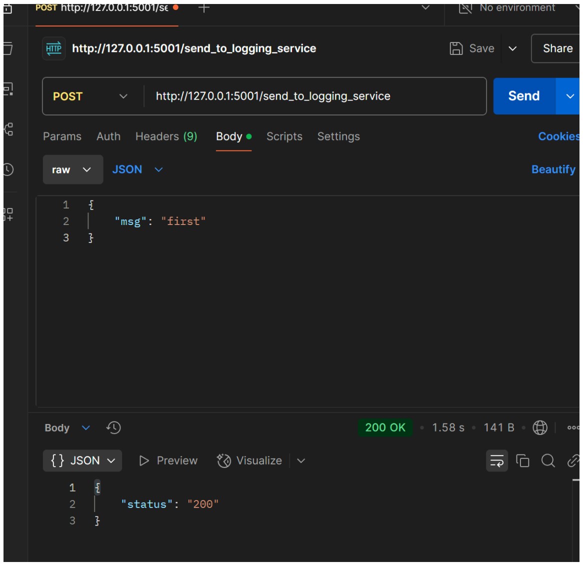
hazelcast_node_1

hazelcast/hazelcast:!

```
1,217] INFO [GroupCoordinat
d joins group message-group
okafka-0.12.0-b8d61b1e-445a
ber to rejoin with this id.
2025-04-13 14:55:41 kafka1
1,232] INFO [GroupCoordinat
d joins group message-group
okafka-0.12.0-4da77799-1158
ber to rejoin with this id.
2025-04-13 14:55:41 kafka1
1,276] INFO [GroupCoordinat
-group in state PreparingRe
ffsets-19) (reason: Adding 
53-95ab-0f36133b6ebd with g
rovided) (kafka.coordinator
2025-04-13 14:55:47 kafka1
7,315] INFO [GroupCoordinat
ation 1 (__consumer_offsets
p.GroupCoordinator)
2025-04-13 14:55:47 kafka1
7,372] INFO [GroupCoordinat
afka-0.12.0-b8d61b1e-445a-4
up for generation 1. The gr
kafka.coordinator.group.Gro
2025-04-13 15:00:19 kafka1
9,569] INFO [Controller id=
eader election (kafka.contr
2025-04-13 15:00:19 kafka1
9,569] TRACE [Controller id
lancing (kafka.controller.K
2025-04-13 15:00:19 kafka1
9,578] DEBUG [Controller id
ker 1 HashMap() (kafka.cont
2025-04-13 15:00:19 kafka1
9,583] TRACE [Controller id
0.0 (kafka.controller.Kafka
2025-04-13 15:00:19 kafka1
9,583] DEBUG [Controller id
ker 2 HashMap() (kafka.cont
2025-04-13 15:00:19 kafka1
9,583] TRACE [Controller id
0.0 (kafka.controller.Kafka
2025-04-13 15:00:19 kafka1
9,583] DEBUG [Controller id
ker 3 HashMap() (kafka.cont
2025-04-13 15:00:19 kafka1
9,583] TRACE [Controller id
0.0 (kafka.controller.Kafka
```

As we can see, there we have two *message services* (with ports 5005 and 5007).

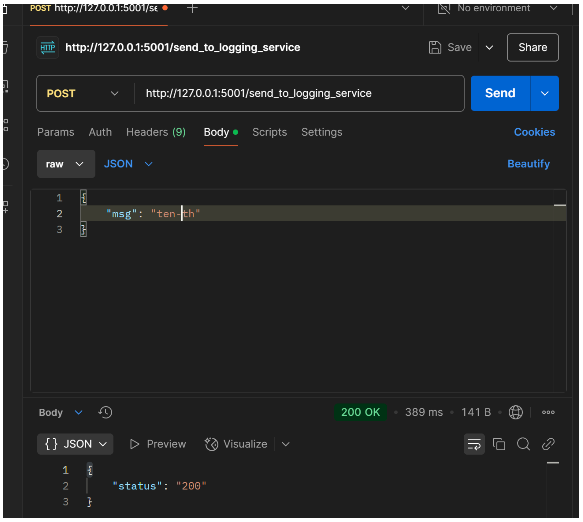
We need to fill out structure with ten messages. I'll do that using Postman:



POST http://127.0.0.1:5001/send_to_logging_service

```
{
    "msg": "first"
}
```

```
{
    "status": "200"
}
```

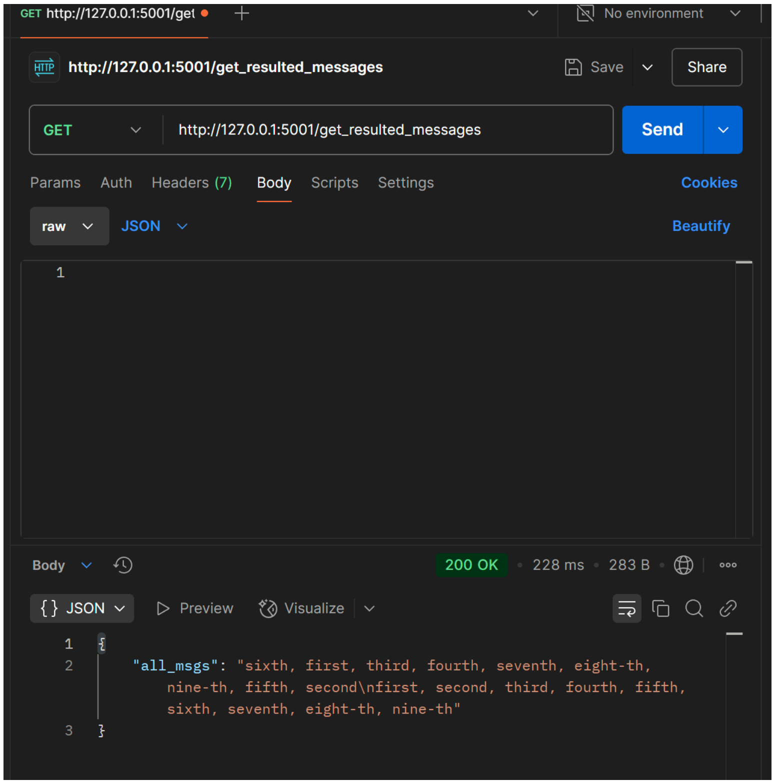


POST http://127.0.0.1:5001/send_to_logging_service

```
{
    "msg": "ten-th"
}
```

```
{
    "status": "200"
}
```

When we request to get everything from the logging service and message service, we will get (I made a screen of ten-th, but forgot to push it 🙈):



GET http://127.0.0.1:5001/get_resulted_messages

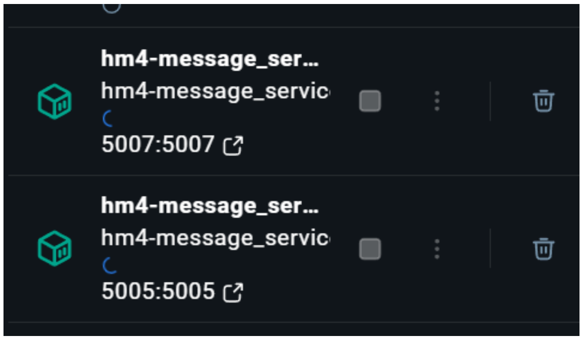
```
{
    "all_msgs": "sixth, first, third, fourth, seventh, eight-th,
        nine-th, fifth, second\nfirst, second, third, fourth, fifth,
        sixth, seventh, eight-th, nine-th"
}
```

Both services got all ten messages.

Checking message queue fault tolerance

I will rerun all services in order to clear them and make a clean experiment.

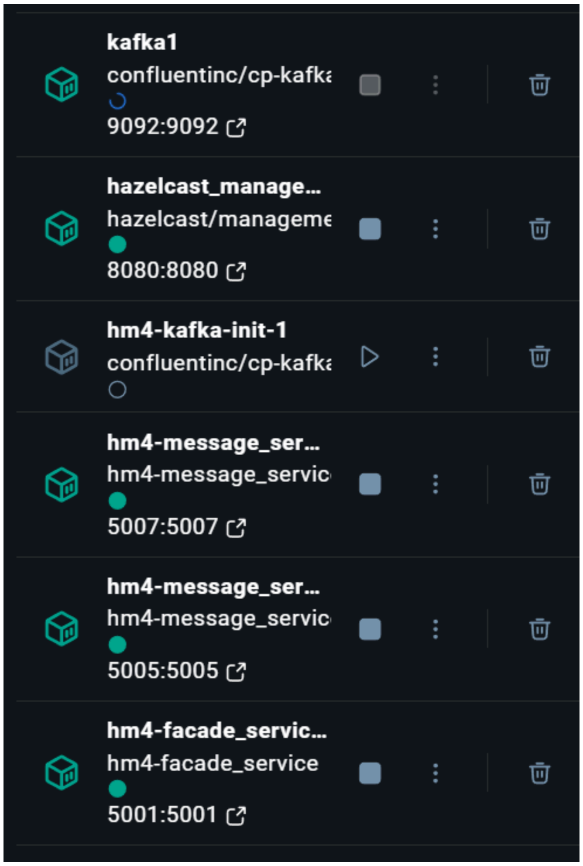
I turned off containers of message services:



hm4-message_ser...
hm4-message_servic
5007:5007

hm4-message_ser...
hm4-message_servic
5005:5005

Then I pushed ten messages. After that I turned off `kafka1` because it was a leader, and turned on two message services:



kafka1
confluentinc/cp-kafka
9092:9092

hazelcast_manage...
hazelcast/manageme
●
8080:8080

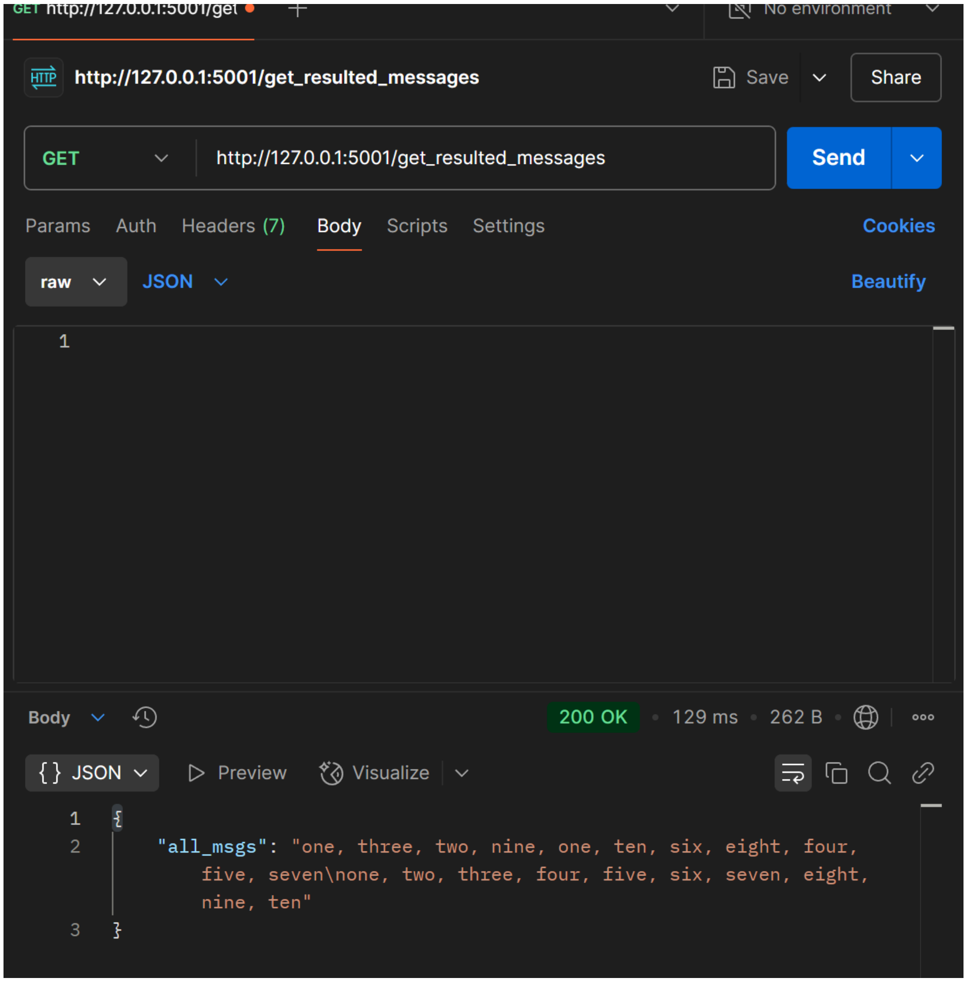
hm4-kafka-init-1
confluentinc/cp-kafka
○

hm4-message_ser...
hm4-message_servic
●
5007:5007

hm4-message_ser...
hm4-message_servic
●
5005:5005

hm4-facade_servic...
hm4-facade_service
●
5001:5001

As a result of `get` option:



http://127.0.0.1:5001/get_resulted_messages

GET http://127.0.0.1:5001/get_resulted_messages

200 OK • 129 ms • 262 B

```
{
    "all_msgs": "one, three, two, nine, one, ten, six, eight, four,
        five, seven\none, two, three, four, five, six, seven, eight,
        nine, ten"
}
```

We get all questions (*note: it is not "none" there, it is just "\n" and "one"*).

It looks like everything is working good!