

Report to homework 3 – “Microservices using Hazelcast Distributed Map”

Part 1:

- 3 instances of the logging_service are connecting to their nodes of the Hazelcast cluster (in my case, the “dev” cluster)
- If logging_service fails – the node is disconnecting; if logging_service restarts – the node restarts as well
- facade_service randomly chooses logging_service and writes a message to the distributed map. If logging_service is unavailable – choose another one

Implementation:

I used a *docker-compose* to run the services. It is easier for me to manage the whole system this way. I am starting 3 logging_service services with names: *logging_service_1*, *logging_service_2*, *logging_service_3*. Using these names, all services will communicate with each other.

For each *logging_service_#*, we should have a node from the hazelcast. They depend on the *logging_service_#* (if this server runs, the node should also be running). Hazelcast is using port 5701.

In the logging_service, I added usage of the hazelcast distributed map:

```
client = hazelcast.HazelcastClient(cluster_name="dev", cluster_members=["hazelcast_node_1:5701", "hazelcast_node_2:5701", "hazelcast_node_3:5701"])
msg_map = client.get_map("messages").blocking()
```

“.../fetching_message” saves messages to the distributed map, i.e.:

```
if msg_map.contains_key(unique_id):
    return {"status": "duplicate, ok"}

msg_map.put(unique_id, message)
print("Saved message: ", message)

return {"status": "ok"}
```

“.../get_fetched_messages” returns all messages from the distributed map, i.e.:

```
return {"msgs": ", ".join(msg_map.values())}
```

If this service stops, it stops, and the node has to be restarted, so it was decided to use a .sh file that monitors events.

Tests:

I used commands to run the docker-compose:

```
PS C:\Users\irunk\UCU\APZ\HM3> docker-compose up --build
time="2025-03-12T15:01:43+02:00" level=warning msg="C:\\Users\\irunk\\UCU\\APZ\\HM3\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Building 3.6s (42/46)
=> [logging_service_2 internal] load build definition from Dockerfile.logging
=> => transferring dockerfile: 319B
=> [logging_service_1 internal] load build definition from Dockerfile.logging
=> => transferring dockerfile: 319B
=> [logging_service_3 internal] load build definition from Dockerfile.logging
=> => transferring dockerfile: 319B
=> [message_service internal] load build definition from Dockerfile.message
=> => transferring dockerfile: 309B
=> [config_service internal] load build definition from Dockerfile.config.service
=> => transferring dockerfile: 334B
```

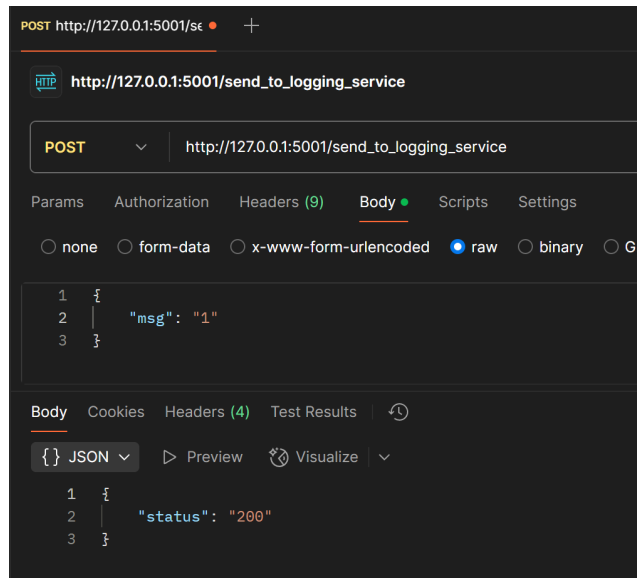
Result (all container running):

hm3				C:\Users\irunk\UCU\APZ\HM3			
hm3-facade_servic...	hm3-facade_service	5001:5001					2025-03-12 15:04:46 hazelcast_node_2
							2025-03-12 15:04:46 hazelcast_node_2
							2025-03-12 15:04:46 hazelcast_node_2
							2025-03-12 15:04:46 hazelcast_node_2
							2025-03-12 15:04:46 hazelcast_node_2
hazelcast_manage...	hazelcast/manageme	8080:8080					2025-03-12 15:04:46 hazelcast_node_2
							2025-03-12 15:04:46 hazelcast_node_2
							2025-03-12 15:04:46 hazelcast_node_2
							2025-03-12 15:04:46 hazelcast_node_2
							2025-03-12 15:04:47 hazelcast_node_2
hm3-config_servic...	hm3-config_service	5006:5006					ized new cluster connection between /172.20.0.8:5701 and
							2025-03-12 15:04:47 hazelcast_node_2
							ing to /172.20.0.7:5701, timeout: 10000, bind-any: true
							2025-03-12 15:04:47 hazelcast_management_center
							0.8]:5701 - 0eb5e84e-054a-4bbf-bce4-08e17af0f5f5
hm3-message_serv...	hm3-message_service	5005:5005					2025-03-12 15:04:47 hazelcast_node_3
							2025-03-12 15:04:47 hazelcast_node_3
							2025-03-12 15:04:47 hazelcast_node_3
							2025-03-12 15:04:47 hazelcast_node_3
							2025-03-12 15:04:47 hazelcast_node_3
hazelcast_node_1	hazelcast/hazelcast:5						2025-03-12 15:04:47 hazelcast_node_3
							2025-03-12 15:04:47 hazelcast_node_3
							2025-03-12 15:04:47 hazelcast_node_3
							2025-03-12 15:04:47 hazelcast_node_3
							2025-03-12 15:04:47 hazelcast_node_3
hm3-logging_servic...	hm3-logging_service_1	5003:5003					zed new cluster connection between /172.20.0.7:42997 and
							2025-03-12 15:04:47 hazelcast_node_3
							zed new cluster connection between /172.20.0.7:5701 and
							2025-03-12 15:04:47 hazelcast_node_2
							ized new cluster connection between /172.20.0.8:54943 and
hm3-logging_servic...	hm3-logging_service_2	5004:5004					2025-03-12 15:04:47 hazelcast_node_1
							ng cluster data. Migration tasks count: 271
							2025-03-12 15:04:47 hazelcast_node_2
							8]:5701 [dev] [5.5.0] Received auth from Connection[id=
							live=true, connectionType=PYH, planeIndex=-1], successf
hazelcast_node_2	hazelcast/hazelcast:5						2025-03-12 15:04:47 hazelcast_node_2
							dev] [5.5.0] Received auth from Connection[id=5, /172.2
							ue, connectionType=MCJVM, planeIndex=-1], successfully
							2025-03-12 15:04:47 hazelcast_node_2
							dev] [5.5.0] Received auth from Connection[id=6, /172.2
hazelcast_node_3	hazelcast/hazelcast:5						, connectionType=PYH, planeIndex=-1], successfully auth
							2025-03-12 15:04:47 hazelcast_node_2
							8]:5701 [dev] [5.5.0] Received auth from Connection[id=
							live=true, connectionType=PYH, planeIndex=-1], successf
							2025-03-12 15:04:48 hazelcast_node_2
hm3-logging_servic...	hm3-logging_service_3	5002:5002					2025-03-12 15:04:48 hazelcast_node_2
							2025-03-12 15:04:48 hazelcast_node_2
							2025-03-12 15:04:48 hazelcast_node_2
							2025-03-12 15:04:48 hazelcast_node_2
							2025-03-12 15:04:48 hazelcast_node_2

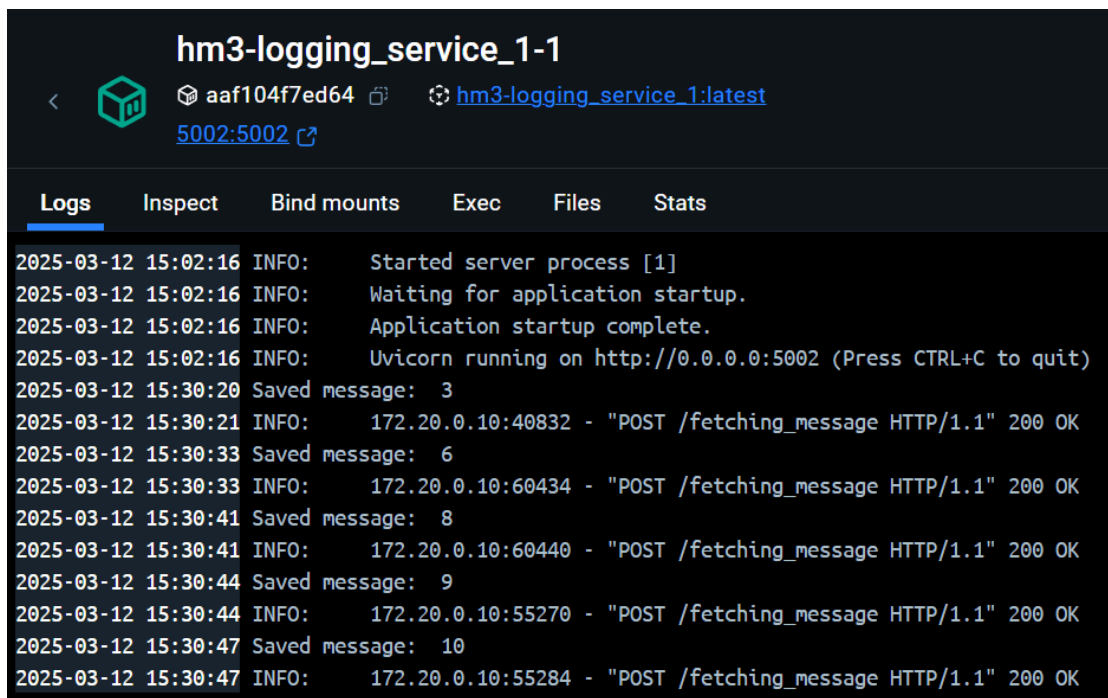
Also started a bash file:

```
PS C:\Users\irunk\UCU\APZ\HM3> bash stop_corresponding_hazelcast.sh
Analysis of the container events... (Logging Service fails -> Hazelcast Node fails, Logging Service restarts -> Hazelcast Node restarts)
-----
```

I submitted 10 messages using *Postman*:



We can take a look into logs to understand which logging_service_# was used to write down a message (because they are chosen randomly):



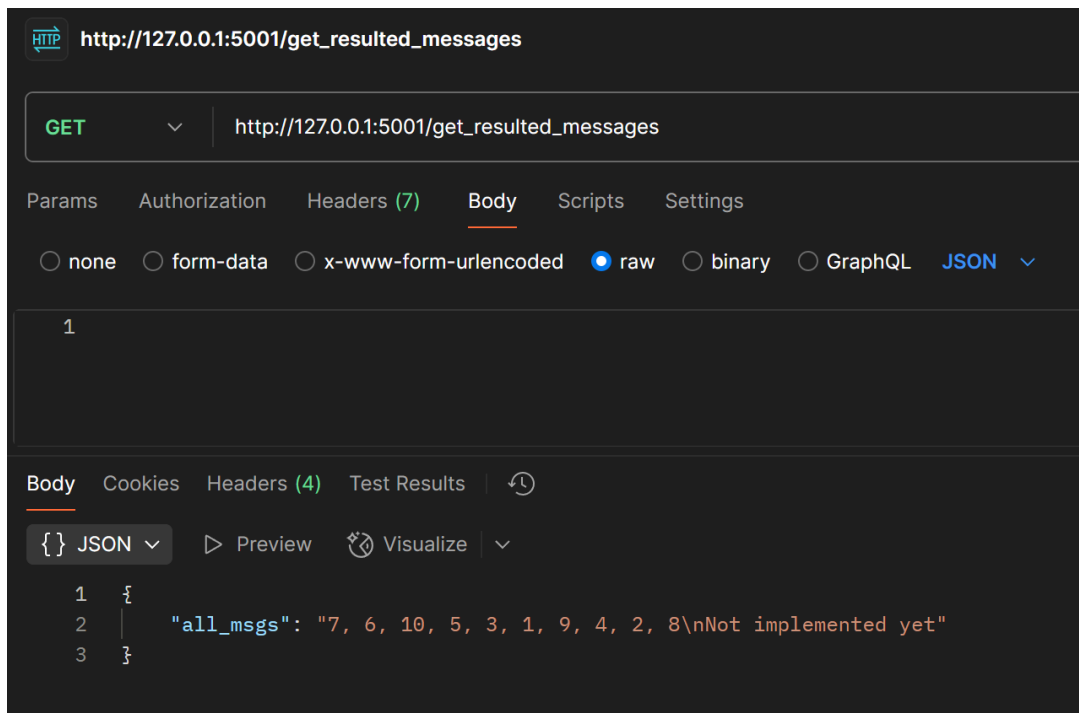
(For the second service)

```
2025-03-12 15:04:38 INFO:      Unicorn running on http://0.0.0.0:5003 (Press CTRL+C to quit)
2025-03-12 15:30:37 Saved message: 7
2025-03-12 15:30:37 INFO:      172.20.0.10:49718 - "POST /fetching_message HTTP/1.1" 200 OK
```

(For the third service)




```
2025-03-12 15:02:16 INFO:      Application startup complete.
2025-03-12 15:02:16 INFO:      Unicorn running on http://0.0.0.0:5004 (Press CTRL+C to quit)
2025-03-12 15:29:36 Saved message: 1
2025-03-12 15:29:36 INFO:      172.20.0.10:46142 - "POST /fetching_message HTTP/1.1" 200 OK
2025-03-12 15:30:17 Saved message: 2
2025-03-12 15:30:17 INFO:      172.20.0.10:47894 - "POST /fetching_message HTTP/1.1" 200 OK
2025-03-12 15:30:24 Saved message: 4
2025-03-12 15:30:24 INFO:      172.20.0.10:50860 - "POST /fetching_message HTTP/1.1" 200 OK
2025-03-12 15:30:29 Saved message: 5
2025-03-12 15:30:29 INFO:      172.20.0.10:50874 - "POST /fetching_message HTTP/1.1" 200 OK
```

We can try to get all messages (use GET) to understand if all of them are in the map:



As we can see, all messages are saved.

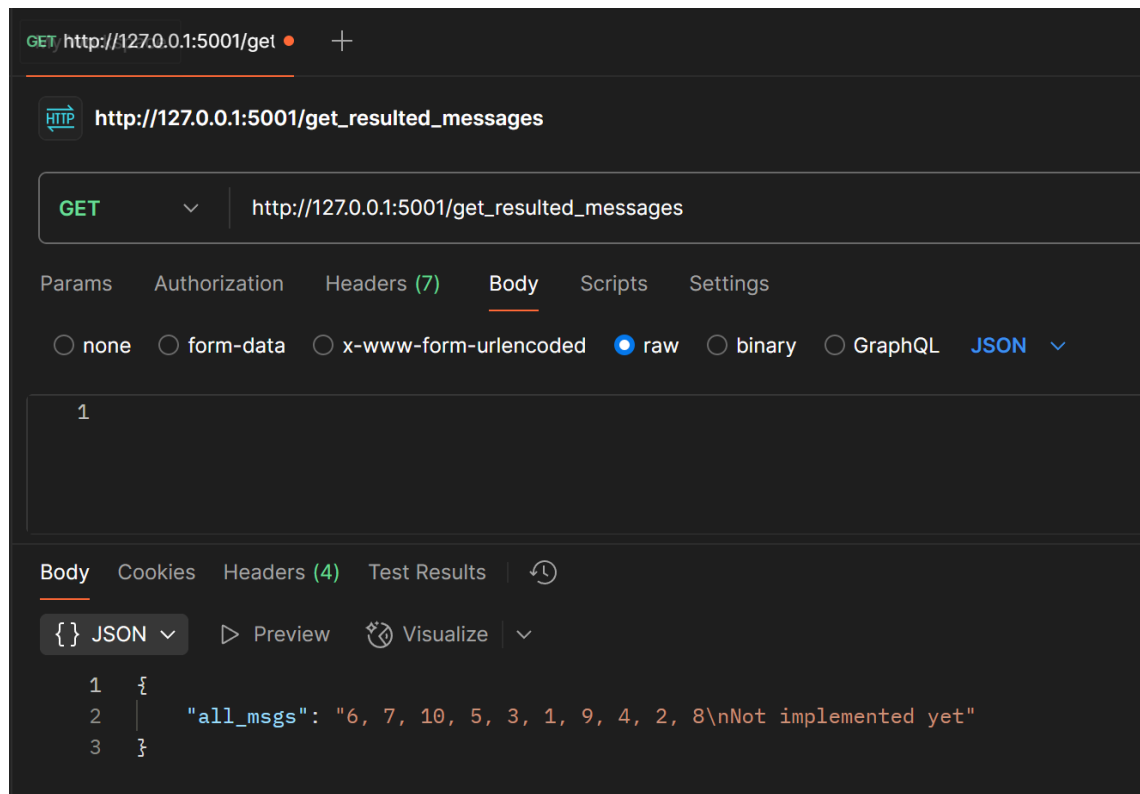
We can try to disconnect one logging_service #:

 hm3-facade_servic... hm3-facade_service 5001:5001	 :		ot connect to: /172.20.0.9:5701. Reason: IOException[Connection refused to address /172.20.0.9:5701] 2025-03-12 15:39:36 hazelcast_node_2 2025-03-12 13:39:36,527 [WARN] [hz.crazy_rosalind.cached.thread-8] [c.h.i.s .5.0] Removing connection to endpoint [172.20.0.9]:5701 Cause => java.io.IOException [Connection refused to address /172.20.0.9 2025-03-12 15:39:36 hazelcast_node_3 2025-03-12 13:39:36,541 [INFO] [hz.zealous_hertz.cached.thread-6] [c.h.t.Tr tting/rolling-back live transactions of [172.20.0.9]:5701, UUID: 68925a43-8ffa-463e-800b-a6a4d7679b90 2025-03-12 15:39:36 hazelcast_node_3 2025-03-12 13:39:36,547 [INFO] [hz.zealous_hertz.cached.thread-9] [c.h.i.c. 2025-03-12 15:39:36 hazelcast_node_3 2025-03-12 15:39:36 hazelcast_node_3 Members {size:2, ver:8} [2025-03-12 15:39:36 hazelcast_node_3 Member [172.20.0.7]:5701 - b13ca7df-2721-4da4-b8fe-2edf2f9623dc this 2025-03-12 15:39:36 hazelcast_node_3 Member [172.20.0.8]:5701 - 0eb5e84e-054a-4bbf-bce4-08e17af0f5f5 2025-03-12 15:39:36 hazelcast_node_3] 2025-03-12 15:39:36 hazelcast_node_3 2025-03-12 15:39:36 hazelcast_management_center 2025-03-12 13:39:36,546 [INFO] [MC-Client-dev.event-4] [c.h.w.s.MCClientMan -463e-800b-a6a4d7679b90 2025-03-12 15:39:36 hazelcast_node_3 2025-03-12 13:39:36,554 [INFO] [hz.zealous_hertz.cached.thread-9] [c.h.i.c. p is claimed with: MembersView(version=8, members=[MemberInfo{address=[172.20.0.7]:5701, uuid=b13ca7df-2721-4da4-b8fe-2edf2f962 =3}, MemberInfo{address=[172.20.0.8]:5701, uuid=0eb5e84e-054a-4bbf-bce4-08e17af0f5f5, cpMemberUUID=null, liteMember=false, memb 2025-03-12 15:39:36 hazelcast_node_2 2025-03-12 13:39:36,572 [INFO] [hz.crazy_rosalind.priority-generic-operatio :5701 [dev] [5.5.0] Storing snapshot of partition assignments while removing UUID 68925a43-8ffa-463e-800b-a6a4d7679b90 2025-03-12 15:39:36 hazelcast_node_2 2025-03-12 13:39:36,625 [INFO] [hz.crazy_rosalind.priority-generic-operatio 5.5.0] 2025-03-12 15:39:36 hazelcast_node_2 2025-03-12 15:39:36 hazelcast_node_2 Members {size:2, ver:8} [2025-03-12 15:39:36 hazelcast_node_2 Member [172.20.0.7]:5701 - b13ca7df-2721-4da4-b8fe-2edf2f9623dc 2025-03-12 15:39:36 hazelcast_node_2 Member [172.20.0.8]:5701 - 0eb5e84e-054a-4bbf-bce4-08e17af0f5f5 this 2025-03-12 15:39:36 hazelcast_node_2] 2025-03-12 15:39:36 hazelcast_node_2 2025-03-12 15:39:36 hazelcast_node_3 2025-03-12 13:39:36,630 [INFO] [hz.zealous_hertz.migration] [c.h.i.p.Intern artition tables from cluster to determine the most recent one... Local stamp: 1901921962995119671 2025-03-12 15:39:36 hazelcast_node_3 2025-03-12 13:39:36,657 [INFO] [hz.zealous_hertz.migration] [c.h.i.p.Intern t partition table is determined. 2025-03-12 15:39:36 hazelcast_node_3 2025-03-12 13:39:36,658 [INFO] [hz.zealous_hertz.migration] [c.h.i.p.Intern he most recent of partition state... 2025-03-12 15:39:36 hazelcast_node_3 2025-03-12 13:39:36,660 [WARN] [hz.zealous_hertz.migration] [c.h.i.p.Intern unknown addresses are found in partition table sent from master[[172.20.0.7]:5701]. (Probably they have recently joined or left 2025-03-12 15:39:36 hazelcast_node_3 [172.20.0.9]:5701 - 68925a43-8ffa-463e-800b-a6a4d7679b90 2025-03-12 15:39:36 hazelcast_node_3] 2025-03-12 15:39:36 hazelcast_node_2 2025-03-12 13:39:36,664 [INFO] [hz.crazy_rosalind.cached.thread-7] [c.h.t.T itting/rolling-back live transactions of [172.20.0.9]:5701, UUID: 68925a43-8ffa-463e-800b-a6a4d7679b90 2025-03-12 15:39:36 hazelcast_node_2 2025-03-12 13:39:36,703 [WARN] [hz.crazy_rosalind.priority-generic-operatio 01 [dev] [5.5.0] Following unknown addresses are found in partition table sent from master[[172.20.0.7]:5701]. (Probably they h 2025-03-12 15:39:36 hazelcast_node_2 [172.20.0.9]:5701 - 68925a43-8ffa-463e-800b-a6a4d7679b90 2025-03-12 15:39:36 hazelcast_node_2] 2036-01-01 00:00:00 hazelcast_node_1 exited with code 143 hazelcast node 3 2025-03-12T13:39:37.183380988Z 2025-03-12 13:39:37,173 [INFO] [hz.zealous_hertz.migration] [c.h
--	--	---	---

I stopped the first container of this service, and the node was automatically disconnected (from the logs, you can see two members in the cluster after that).

```
-----  
Container hm3-logging_service_1-1 failed -> Stopping hazelcast_node_1  
-----  
hazelcast_node_1
```

I also checked what happened with the data in the map:



As it can be seen, all the data remained the same, i.e., in the map it just regrouped.

Part 2:

- Create a config file with all available ports
- facade_service should make a request to the config_service, which returns *urls* to all available ports
- facade_service randomly chooses a port and tries to send a message

Implementation:

I added a new *FastAPI* application, which has an endpoint “.../all_ports”:

```
@app.get("/all_ports")
def handle() -> AvailableAPI:
    """
    Returns a static text.
    """

    all_appis = {
        "log_s": [f"http://" + "logging_service_" + str(port-5001) + ":" + str(port) for port in LOGGING_SERVICE_PORTS],
        "mes_s": f"http://" + "message_service" + ":" + str(MESSAGES_SERVICE)
    }

    return all_appis
```

In facade_service, I am making a request and then sorting all available ports (to make it random):

```
response = await client.get(url=f"http://config_service:5006/all_ports")
all_ports: AvailableAPI = AvailableAPI(**response.json())

log_ports = all_ports.log_s
random.shuffle(log_ports)

for port in log_ports:
    for num in range(5):
```

That is all, (no need on test examples to this part because the first part uses this code already)