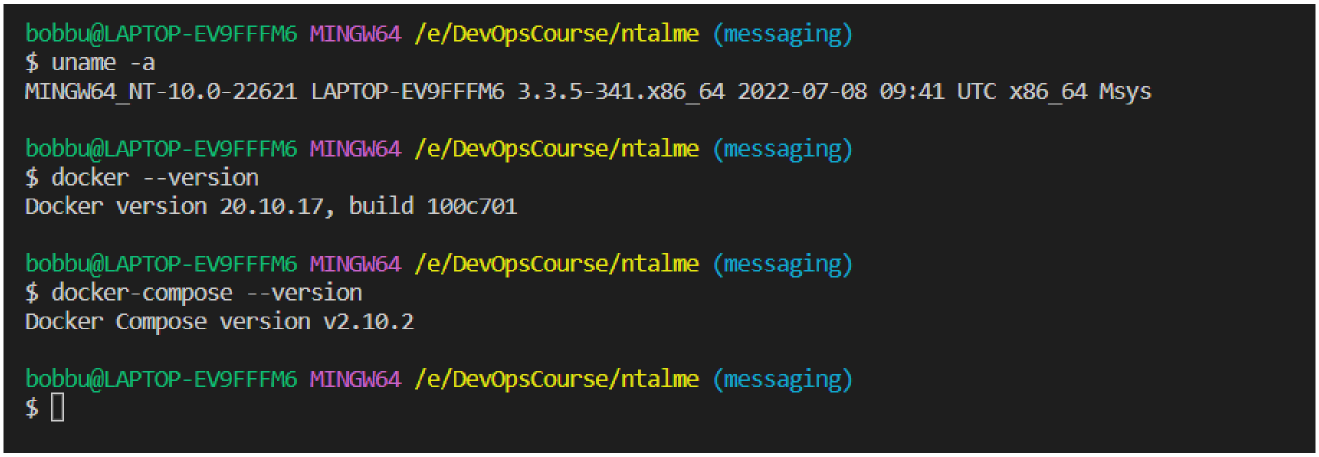
To run the application:

1. **Clone the repo: git clone -b messaging https://course-gitlab.tuni.fi/compse140-2022-2023/ntalme.git**
2. **Change directory to project folder and run: docker-compose build –-no-cache**
3. **Run: docker-compose up –d**
4. **Wait for about 30 seconds and run: curl localhost:8080**

Information about the host; output of: uname -a; docker --version; docker-compose --version:



```
bobbu@LAPTOP-EV9FFFM6 MINGW64 /e/DevOpsCourse/ntalme (messaging)
$ uname -a
MINGW64_NT-10.0-22621 LAPTOP-EV9FFFM6 3.3.5-341.x86_64 2022-07-08 09:41 UTC x86_64 Msys

bobbu@LAPTOP-EV9FFFM6 MINGW64 /e/DevOpsCourse/ntalme (messaging)
$ docker --version
Docker version 20.10.17, build 100c701

bobbu@LAPTOP-EV9FFFM6 MINGW64 /e/DevOpsCourse/ntalme (messaging)
$ docker-compose --version
Docker Compose version v2.10.2

bobbu@LAPTOP-EV9FFFM6 MINGW64 /e/DevOpsCourse/ntalme (messaging)
$ 
```

Benefits of the topic-based communication compared to request-response (HTTP):

For my observation, it was easy to send messages easily to servers and a lot less setup was required to send information between containers. Also, no server setup was required for the communication to be present. It also seemed quite fast.

Main learnings:

I have previously worked with RabbitMQ, but it was only limited to using direct queues and messages. Working with topics was quite new to me. I spent quite a lot of time learning how to utilize this and now I have a good understanding of both Docker and RabbitMQ.