

Juho Tapio  
K444476  
7.12.2022  
End report for DevOps course

## 1. Instructions for the teaching assistant

### Optional features

- /node-statistics

### Instructions for examiner to test the

You can run the software with **docker compose up --build**

The shutdown state update doesn't stop RabbitMQ. That has to be done with docker compose.

## 2. Description of the CI/CD pipeline

The test code has been put into a separate docker-compose file. You can find the pipeline code on the gitlab-ci file.

## 3. Example runs of the pipeline

```
345 CI_1      |      localPort: 41350,
346 CI_1      |      remoteAddress: '172.19.0.3',
347 CI_1      |      remotePort: 8083,
348 CI_1      |      remoteFamily: 'IPv4',
349 CI_1      |      timeout: undefined,
350 CI_1      |      bytesWritten: 525,
351 CI_1      |      bytesRead: 465
352 CI_1      |    }
353 CI_1      |  }
354 CI_1      | }
355 CI_1      |
356 CI_1      | Node.js v19.2.0
357 compsel140_CI_1 exited with code 1
358 Aborting on container exit...
✓ 360 Cleaning up project directory and file based variables
362 ERROR: Job failed: exit code 1
```

321 CI\_1 | Retrying connection

322 CI\_1 | Retrying connection

323 CI\_1 | Retrying connection

324 CI\_1 | Retrying connection

325 CI\_1 | Retrying connection

326 CI\_1 | PUT /state < INIT

327 CI\_1 | GET /state == RUNNING

328 CI\_1 | GET /messages

329 CI\_1 | 2022-12-07T13:56:16.241Z 1 MSG\_2 to compse140.o

330 CI\_1 | 2022-12-07T13:56:17.248Z 2 Got MSG\_2 to compse140.i

331 CI\_1 |

332 CI\_1 | PUT /state < SHUTDOWN

333 CI\_1 | GET /state fails successfully

334 compse140\_CI\_1 exited with code 0

335 Aborting on container exit...

✓ 337 Cleaning up project directory and file based variables 00:01

339 Job succeeded

Duration: 1 minute 57 seconds

Finished: 26 minutes ago

Queued: 0 seconds

Timeout: 1h (from project) ?

Runner: #12270835 (zxwgkjAP)

3-blue.shared.runners-

manager.gitlab.com/default

Commit 4242200d

Remove ports

✓ Pipeline #715956871 for project

start

## 4. Reflections

This has been one of the most interesting courses this far. CI/CD stuff is really interesting. My worst difficulty was learning RabbitMQ architecture. I didn't first understand why you would even need a message queue but now it makes more sense.

There's one thing I don't get... **PDF FILES ON GIT**. Why? Markdown is there for a reason. This seems to be a problem on multiple courses. How are you supposed to diff PDF files?

I spent about 15 hours coding the final stage. I have some previous experience from CI/CD and Docker Compose. Figuring out Docker-in-Docker and gitlab services took a lot of time that was very valuable for learning. I didn't code basically any of the optional requirements. My goal is to study 120 credits this year and graduate.

I think the SHUTDOWN state update part should have been done differently. I don't get why anyone would want one service to shutdown the whole system. However, if I wanted to implement this in a production environment I would use the deploy environment's SDKs and configuration (google cloud for example). Using docker compose commands from inside a container is a security risk and bad design.

RabbitMQ is the only container that cannot be shutdown with a command. Another solution for the shutdown procedure would be to create a new docker image with a script that shuts down the child process (RabbitMQ) when called. That sounds overly complicated and frankly ridiculous so I didn't implement it. I don't like the idea of shutting down a system with the system since you cannot start it the same way anymore.