

End Report

1 CONTENTS

1	Instructions	2
1.1	Optional features	2
1.2	Instructions for testing.....	2
1.2.1	Instructions to start the application.....	2
1.2.2	Instructions to test the application	2
1.2.3	Development platform information	2
2	CI/CD pipeline	3
2.1	Description	3
2.2	Example runs of the pipeline	4
2.2.1	Example of failed tests.....	4
2.2.2	Example of passed tests	5
3	Reflection.....	8
3.1	Main learnings and worst difficulties	8
3.2	Things I would do differently.....	9
3.3	Amount effort used.....	9
3.4	Usage of AI	9

1 INSTRUCTIONS

1.1 OPTIONAL FEATURES

- Statical analysis is implemented through linting and formatting for Node.js-based services, including tests, service2 and api-gateway. Linting is implemented with eslint and formatting with Prettier.
- Tests and statical analysis are run in the CI/CD pipeline.

1.2 INSTRUCTIONS FOR TESTING

1.2.1 Instructions to start the application

- Ensure Docker and Docker Compose are installed on your system.
- Clone the repository and navigate to the root folder of the project.
- Build the services using the following command:
 - `docker-compose build --no-cache`
- Start services using:
 - `docker-compose up`

1.2.2 Instructions to test the application

- Using curl-commands:
 - Log in to the application:
 - `curl http://localhost:8197 -u testuser:testpassword -H "Accept: text/plain"`
 - Example of GET request:
 - `curl http://localhost:8197/state -H "Accept: text/plain"`
 - Example of PUT request:
 - `curl http://localhost:8197/state -X PUT -d "PAUSED" -H "Content-Type: text/plain" -H "Accept: text/plain"`
- Run tests cases:
 - Restart the container to clear application data:
 - `docker-compose down && docker-compose up`
 - Build and run tests in a container:
 - `docker build -t tests-image ./tests --no-cache`
 - `docker run --rm tests-image`
 - Alternatively navigate to the tests folder and execute tests with:
 - `npm run test`
- Use of browser:
 - After logging in via curl, access the website in a browser. Buttons are provided to make requests: GET /request and PUT /state to shutdown services.

1.2.3 Development platform information

- Hardware
 - CPU: AMD Ryzen 5 4600H with Radeon Graphics (3.00 GHz, 6 cores, 12 processors)
 - Memory: 24.0 GB DDR4 RAM, 3200 MHz
 - Storage: 476 GB SSD (SKHynix_HFM512GDHTNI-87A0B)

- GPU: NVIDIA GeForce GTX 1660 Ti, AMD Radeon™ Graphics
- Software
 - CPU Architecture: x64-based PC
 - Operating System: Windows 11 Home 23H2
 - Docker Version: 27.3.1
 - Docker Compose Version: 2.29.7
 - Node Version (local testing, linting and formatting): 20.18.0

2 CI/CD PIPELINE

2.1 DESCRIPTION

- Version management
 - Git is used for version control.
 - The “project” branch is dedicated to development and final deployment.
 - The pipeline triggers on every commit, with deployment restricted to commits made to the “project” branch.
 - Pipeline is implemented to GitLab CI/CD pipeline and run by GitLab runners.
- Static analysis
 - The pipeline starts with linting and formatting checks for Node based services tests, service2 and api-gateway with Eslint and Prettier.
 - This step ensures consistent code style and detects syntax errors.
- Build tools
 - Docker is used for containerizing services
 - Docker Compose orchestrates the multi-container setup.
- Testing
 - Frameworks: Mocha, Chai and Chai-http for unit testing.
 - Tests are executed in the pipeline in independent docker container outside of other services.
 - Test cases are divided into three sections based on the tested endpoint.
 - Test cases cover login, GET and PUT requests to /state, GET requests to /request and /run-log.
 - Tests are chained and all of them are run in one go. Because services state changes and logs and auth status are saved, If one of the tests fails, it can follow that some of the next tests can also fail even though they would work correctly with the correct service state.
 - The last test of testing SHUTDOWN will shutdown all of the services.
- Packing
 - Services are packed using:
 - docker-compose build --no-cache
- Deployment
 - Automated deployment in “project” branch via Docker Compose using:
 - docker-compose up -d
- Monitoring

- Test logs are accessible in the GitLab pipeline. No additional monitoring tools are implemented.

2.2 EXAMPLE RUNS OF THE PIPELINE

2.2.1 Example of failed tests

Updated tests and created new one to endpoints /request and /run-log

Retry

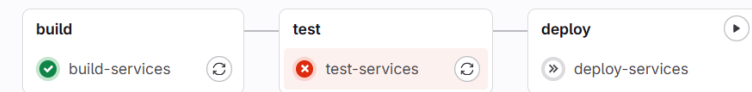
Delete

 Failed **Santeri Ora** created pipeline for commit `28e51df7` 1 day ago, finished 1 day ago

For **project**

latest 4 jobs 1 minute 57 seconds, queued for 1 seconds

Pipeline Jobs 4 Failed Jobs 1 Tests 0

Santeri Ora / COMP.SE.140 / Jobs / #1805

test-services

 Failed Started 1 day ago by Santeri Ora

Search visible log output

[illegible]


Duration: 1 minute 31 seconds

Finished: 1 day ago

Queued: 0 seconds

Timeout: 1h (from project) ?

Runner: #16 (KuMNs28s)


Commit 28e51df7 

Updated tests and created new one to endpoints /request and /run-log

Pipeline #567 Failed for project

test

Related jobs

→  test-services

- test-services

Search visible log output

```

47 41 packages are looking for funding
48   run `npm fund` for details
49 found 0 vulnerabilities
50 > tests@1.0.0 test
51 > mocha *.test.js
52 API route tests
53   /request endpoint tests
54     1) Should return the correct data structure as plain text
55   /run-log endpoint test
56     ✓ Should return INIT->RUNNING after login
57     ✓ Should change state to INIT and log out (2527ms)
58   /state endpoint tests
59     ✓ Should return INIT before login
60     ✓ Should return RUNNING when logged in
61     ✓ Should change state to SHUTDOWN
62     ✓ Should not repond
63 6 passing (23s)
64 1 failing
65 1) API route tests
66     /request endpoint tests
67       Should return the correct data structure as plain text:
68     Uncaught SyntaxError: Unexpected token 'R', "Request handled" is not valid JSON
69       at JSON.parse (<anonymous>)
70       at file:///C:/GitLab-Runner/builds/t3_KuMNs2/0/txsaor/comp.se.140/tests/api.test.js:40:43
71       at Request.callback (node_modules\superagent\lib\node\index.js:837:12)
72       at IncomingMessage.<anonymous> (node_modules\superagent\lib\node\index.js:1078:18)
73       at IncomingMessage.emit (node:events:531:35)
74       at endReadableNT (node:internal\streams/readable:1696:12)
75       at process.processTicksAndRejections (node:internal/process/task_queues:82:21)
76 Cleaning up project directory and file based variables
77 ERROR: Job failed: exit status 1

```

00:01

Duration: 1 minute 31 seconds
 Finished: 1 day ago
 Queued: 0 seconds
 Timeout: 1h (from project) [?](#)
 Runner: #16 (KuMNs28s)

Commit 28e51df7
 Updated tests and created new one to endpoints /request and /run-log

Pipeline #567 Failed for project

test ▼

Related jobs

→ test-services

test-services

2.2.2 Example of passed tests

Santeri Ora / COMP.SE.140 / Pipelines / #674

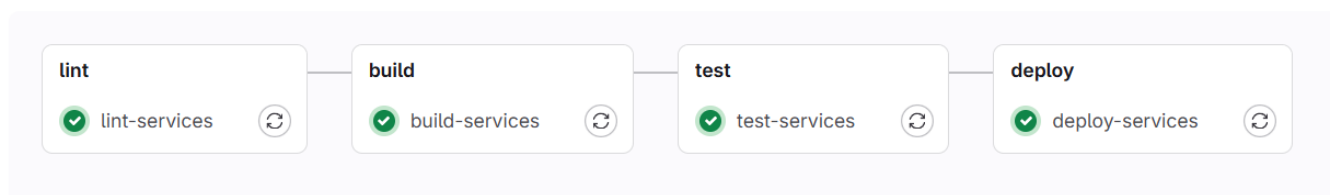
Refactored lint and format to root

Passed Santeri Ora created pipeline for commit 314a29e6 4 hours ago, finished 4 hours ago

For project

latest 4 jobs 2 minutes 43 seconds, queued for 1 seconds

Pipeline Jobs 4 Tests 0



test-services

✓ Passed Started 4 hours ago by Santeri Ora

Search visible log output

```
1 Running with gitlab-runner 17.5.3 (12030cf4)
2   on Bash t3_KuMNs2, system ID: s_74cf6db5f4f3
3   ✓ Preparing the "shell" executor
4   Using Shell (bash) executor...
5   ✓ Preparing environment
6   Running on DESKTOP-RSQFGQN...
7   ✓ Getting source from Git repository
8   Fetching changes with git depth set to 20...
9   Reinitialized existing Git repository in C:/GitLab-Runner/builds/t3_KuMNs2/8/txsaor/comp.se.140/.git/
10  Checking out 314a29e6 as detached HEAD (ref is project)...
11  Skipping Git submodules setup
12  ✓ Executing "step_script" stage of the job script
13  $ docker-compose up -d
14  Container compse140-nginx-1 Stopping
15  Container compse140-service1-2 Stopping
16  Container compse140-service1-3 Stopping
17  Container compse140-service1-1 Stopping
18  Container compse140-service1-1 Stopped
19  Container compse140-nginx-1 Stopped
20  Container compse140-api-gateway-1 Recreate
21  Container compse140-service1-2 Stopped
22  Container compse140-service1-3 Stopped
23  Container compse140-service2-1 Recreate
24  Container compse140-service2-1 Recreated
25  Container compse140-nginx-1 Stopping
```

Search visible log output

```
26  Container compse140-nginx-1 Stopped
27  Container compse140-service1-3 Recreate
28  Container compse140-nginx-1 Stopping
29  Container compse140-nginx-1 Stopped
30  Container compse140-nginx-1 Stopping
31  Container compse140-service1-1 Recreate
32  Container compse140-nginx-1 Stopped
33  Container compse140-service1-2 Recreate
34  Container compse140-api-gateway-1 Recreated
35  Container compse140-service1-2 Recreated
36  Container compse140-service1-3 Recreated
37  Container compse140-service1-1 Recreated
38  Container compse140-nginx-1 Recreate
39  Container compse140-nginx-1 Recreated
40  Container compse140-service2-1 Starting
41  Container compse140-api-gateway-1 Starting
42  Container compse140-service2-1 Started
43  Container compse140-api-gateway-1 Started
44  Container compse140-service1-2 Starting
45  Container compse140-service1-2 Started
46  Container compse140-service1-1 Starting
47  Container compse140-service1-1 Started
48  Container compse140-service1-3 Starting
49  Container compse140-service1-3 Started
50  Container compse140-nginx-1 Starting
51  Container compse140-nginx-1 Started
52  $ docker build -t tests-image ./tests --no-cache
53  #0 building with "desktop-linux" instance using docker driver
54  #1 [internal] load build definition from Dockerfile
55  #1 transferring dockerfile: 187B done
```



Duration: 1 minute 1 second

Finished: 4 hours ago

Queued: 2 seconds

Timeout: 1h (from project) ?

Runner: #16 (KuMNs28s)

Commit 314a29e6

Refactored lint and format to root

Pipeline #674 ✓ Passed for project

test

Related jobs

→ ✓ test-services

Duration: 1 minute 1 second

Finished: 4 hours ago

Queued: 2 seconds

Timeout: 1h (from project) ?

Runner: #16 (KuMNs28s)

Commit 314a29e6

Refactored lint and format to root

Pipeline #674 ✓ Passed for project

test

Related jobs

→ ✓ test-services

Search visible log output

```
56 #1 DONE 0.0s
57 #2 [internal] load metadata for docker.io/library/node:20.18.0
58 #2 DONE 0.5s
59 #3 [internal] load .dockerignore
60 #3 transferring context: 2B done
61 #3 DONE 0.0s
62 #4 [1/4] FROM docker.io/library/node:20.18.0@sha256:a7a3b7ec6de4b11bb2d673b31de9d28c6da09c557ee65453672c8e4f754c23fc
63 #4 DONE 0.0s
64 #5 [2/4] WORKDIR /tests
65 #5 CACHED
66 #6 [internal] load build context
67 #6 transferring context: 60.44kB 0.5s done
68 #6 DONE 0.6s
69 #7 [3/4] COPY . .
70 #7 DONE 0.6s
71 #8 [4/4] RUN npm install
72 #8 6.684 npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
73 #8 6.882 npm warn deprecated glob@8.1.0: Glob versions prior to v9 are no longer supported
74 #8 8.062
75 #8 8.062 added 119 packages, and audited 120 packages in 7s
76 #8 8.062
77 #8 8.062 41 packages are looking for funding
78 #8 8.062   run `npm fund` for details
79 #8 8.068
80 #8 8.068 found 0 vulnerabilities
81 #8 8.069 npm notice
82 #8 8.069 npm notice New minor version of npm available! 10.8.2 -> 10.9.1
83 #8 8.069 npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.9.1
84 #8 8.069 npm notice To update run: npm install -g npm@10.9.1
```

Duration: 1 minute 1 second
Finished: 4 hours ago
Queued: 2 seconds
Timeout: 1h (from project) [?](#)
Runner: #16 (KuMNs28s)

Commit 314a29e6 [🔗](#)

Refactored lint and format to root

Pipeline #674 Passed for project [🔗](#)

test

Related jobs

→ test-services

Search visible log output

```
85 #8 8.069 npm notice
86 #8 DONE 8.3s
87 #9 exporting to image
88 #9 exporting layers
89 #9 exporting layers 0.5s done
90 #9 writing image sha256:dc30856413470788209a7497e968012e67ab082dfafd9ae1586e096f51d61037 done
91 #9 naming to docker.io/library/tests-image 0.0s done
92 #9 DONE 0.5s
93 View build details: docker-desktop:///dashboard/build/desktop-linux/desktop-linux/tqbtlylna05zfd26nyn4vd7w5q
94 $ sleep 10
95 $ docker run --rm tests-image
96 > tests@1.0.0 test
97 > mocha *.test.js
98 API route tests
99   /request endpoint tests
100     ✓ Should return the correct data structure as plain text (2752ms)
101   /run-log endpoint test
102     ✓ Should return INIT->RUNNING after login
103     ✓ Should change state to INIT and log out (2520ms)
104   /state endpoint tests
105     ✓ Should return INIT before login
106     ✓ Should return RUNNING when logged in
107     ✓ Should change state to SHUTDOWN
108     ✓ Should not respond
109   7 passing (23s)
110 $ docker-compose down
111 Container compse140-nginx-1 Stopping
112 Container compse140-nginx-1 Stopped
113 Container compse140-nginx-1 Removing
114 Container compse140-nginx-1 Removed
```

Duration: 1 minute 1 second
Finished: 4 hours ago
Queued: 2 seconds
Timeout: 1h (from project) [?](#)
Runner: #16 (KuMNs28s)

Commit 314a29e6 [🔗](#)

Refactored lint and format to root

Pipeline #674 Passed for project [🔗](#)

test

Related jobs

→ test-services

Search visible log output

```

110 $ docker-compose down
111 Container compse140-nginx-1 Stopping
112 Container compse140-nginx-1 Stopped
113 Container compse140-nginx-1 Removing
114 Container compse140-nginx-1 Removed
115 Container compse140-service1-3 Stopping
116 Container compse140-service1-2 Stopping
117 Container compse140-service1-1 Stopping
118 Container compse140-api-gateway-1 Stopping
119 Container compse140-service1-3 Stopped
120 Container compse140-service1-3 Removing
121 Container compse140-api-gateway-1 Stopped
122 Container compse140-api-gateway-1 Removing
123 Container compse140-service1-2 Stopped
124 Container compse140-service1-2 Removing
125 Container compse140-service1-1 Stopped
126 Container compse140-service1-1 Removing
127 Container compse140-service1-3 Removed
128 Container compse140-service1-1 Removed
129 Container compse140-service1-2 Removed
130 Container compse140-service2-1 Stopping
131 Container compse140-service2-1 Stopped
132 Container compse140-service2-1 Removing
133 Container compse140-api-gateway-1 Removed
134 Container compse140-service2-1 Removed
135 Network compse140_loadbalancing Removing
136 Network compse140_loadbalancing Removed
137 Cleaning up project directory and file based variables
138 Job succeeded

```

Duration: 1 minute 1 second
 Finished: 4 hours ago
 Queued: 2 seconds
 Timeout: 1h (from project) [?](#)
 Runner: #16 (KuMNs28s)

Commit [314a29e6](#)
 Refactored lint and format to root

Pipeline #674 ✔ Passed for project

Related jobs
 → ✔ test-services

3 REFLECTION

3.1 MAIN LEARNINGS AND WORST DIFFICULTIES

I have had experience with Docker and using multiple services earlier. Although the final project and earlier exercises provided some new experiences with Docker, especially with more advanced usage like using entrypoints and creating replicas of services. The learnings that I gained were more about the new technologies used in the project, such as NGINX and the GitLab CI/CD pipeline with runners. Most of the challenges also came from these areas. By learning from the internet and experimenting, I created authentication, rate limiting, and state handling in NGINX. In earlier exercises, the NGINX implementation was very simple, but for the project, I had to rethink the whole structure and learn even more about NGINX and its capabilities.

Most of the challenges came from small details and implementing new things that I had never done before. There weren't any major issues, but I encountered some blockers where I got stuck and spent most of the project time. The first issue was shutting down the services inside Docker. I managed to handle that with entrypoints, and it worked well on my computer and in the GitLab pipeline when running tests. I received feedback from the NGINX exercise that the usage of the whole service required removing the entrypoint, and after that, shutting down the services didn't work. For this final project, I couldn't find the reason for that because the implementation works without any problems in Docker on my computer and also in GitLab runners. I also containerized the tests to get detached results.

Especially for this final project, most of the difficulties came from saving the authentication state and the services state to NGINX. I had to redo the authentication logic and switch the whole NGINX image to allow the use of Lua blocks. After all, I got NGINX working as intended, but I am still not very satisfied with the implementation. The logic is pretty messy, and due to time constraints, I couldn't improve it to the best of my ability.

I also encountered some issues running GitLab runners on my computer, but I solved that after one day. Overall, these worst difficulties that I mentioned took up 80% of the whole time I used for the project. All other tasks were very simple or I was already familiar with them.

3.2 THINGS I WOULD DO DIFFERENTLY

Things that I would do differently would be testing and error handling. I would perform more comprehensive testing to ensure there are no bugs and unwanted behaviors in NGINX because it was a very new technology for me, and as I already mentioned, I am not fully satisfied with the implementation. The other thing I would do differently is more advanced error handling. For the earlier exercises, I tried to handle errors well and show correct messages to users. With the final project I realized that proper and more detailed error handling would take too long, so I just focused more on positive usage and correct responses of the services.

3.3 AMOUNT EFFORT USED

50 hours

3.4 USAGE OF AI

- GitHub Copilot (ChatGPT) was used for syntax fixes. I also attempted to solve advanced errors and seek assistance, but most of the responses were incorrect and ended up being a waste of time.
- ChatGPT was used to review the End Report's English language and provide corrections.