

Instructions for the teaching assistant

Implemented optional features

- Testing of individual services
- Monitoring and logging for troubleshooting

Instructions for examiner to test the system.

Platform I used:

What	Version/Specs
OracleVirtualbox	ubuntu-24.04.1-desktop-amd64.iso
PC hardware	I9-9900K Coffee Lake Refresh architecture with Socket 1151. Max 8CPU for VM I think 25gb of ssd virtually partitioned and shared with my main computer(bidirectional) 8GB ram allocated
Docker	v 27.3.1, build ce12230
Docker-compose	v2.17.2

How to test the system:

docker build and up from the git just as in the requirements.

When the containers are up you can run the tests: docker-compose run tests npm test

Note: The test script is configured for the CI pipeline so it will shut down the containers after tests are run.

If you want to test the functionality yourself you can curl the apigateway for example:

```
curl -u admin:admin localhost:8197/state -X PUT -d "RUNNING"
```

REMEMBER THE CREDENTIALS: -u admin:admin for the API GATEWAY

The same credentials are used for the <http://localhost:8198/> page

If you want to test the monitoring/logging optional feature you can either curl the apigateway for /monitor: curl localhost:8197/monitor -X GET

OR

You can open the webpage and press show logs and then the logs are visible on the website.

Description of the CI/CD pipeline

When pushing to git the following pipeline will execute:

- Build
 - Builds the docker-compose locally. Nothing fancy.
- Test

- Puts down all containers and removes them. Then builds and compose up the containers and then runs the tests. After tests shuts down the containers.
- Deploy:
 - Builds the containers and compose up locally. Nothing fancy.

Example runs of the pipeline Include some kind of log of both failing test and passing.

Failing test:

```

299 $ docker-compose run tests npm test
300 Container shutdown-service Running
301 Container compse140-project-service1-1 Running
302 Container compse140-project-service1-2 Running
303 Container compse140-project-service1-3 Running
304 Container nginx Running
305 > apigatewayTest@1.0.0 test
306 > mocha singleServiceTest.js apigatewayTest.js --exit
307 GET service1
308   ✓ should return the service information from service1 and 2 (365ms)
309 GET nginx
310   ✓ should return the main html nginx
311 GET /state
312   ✓ should return the current state (52ms)
313 PAUSED State Behavior with GET /state
314   ✓ should not process requests when in PAUSED state (GET /state)
315 INIT State Behavior with GET /state
316   ✓ should not process requests when in PAUSED state (GET /state)
317 PAUSED State Behavior with GET /request
318   ✓ should not process requests when in PAUSED state (GET /request)
319 RUNNING State Behavior with /request
320   ✓ should process requests normally when in RUNNING state (166ms)
321 GET /run-log
322   ✓ should return state transition log
323 GET /monitor
324   1) should return monitor logs
325 SHUTDOWN State Behavior when state RUNNING
326   ✓ should stop all containers when in SHUTDOWN state
327 9 passing (780ms)
328 1 failing
329 1) GET /monitor
330     should return monitor logs:
331     AssertionError [ERR_ASSERTION]: Expected status code to be 200
332     + expected - actual
333     -404
334     +200
335
336     at Context.<anonymous> (apigatewayTest.js:125:12)
337     at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
338 Cleaning up project directory and file based variables
339 ERROR: Job failed: exit code 1

```

Passing test:

```
334 $ docker-compose run tests npm test
335 time="2025-01-18T15:52:02Z" level=warning msg="Found orphan containers ([compse
336 Container shutdown-service Running
337 Container compse140-project-service1-1 Running
338 Container compse140-project-service1-2 Running
339 Container compse140-project-service1-3 Running
340 Container nginx Running
341 > apigatewayTest@1.0.0 test
342 > mocha singleServiceTest.js apigatewayTest.js --exit
343 GET service1
344   ✓ should return the service information from service1 and 2 (374ms)
345 GET nginx
346   ✓ should return the main html nginx
347 GET /state
348   ✓ should return the current state (53ms)
349 PAUSED State Behavior with GET /state
350   ✓ should not process requests when in PAUSED state (GET /state)
351 INIT State Behavior with GET /state
352   ✓ should not process requests when in PAUSED state (GET /state)
353 PAUSED State Behavior with GET /request
354   ✓ should not process requests when in PAUSED state (GET /request)
355 RUNNING State Behavior with /request
356   ✓ should process requests normally when in RUNNING state (409ms)
357 GET /run-log
358   ✓ should return state transition log
359 GET /monitor
360   ✓ should return monitor logs
361 SHUTDOWN State Behavior when state RUNNING
362   ✓ should stop all containers when in SHUTDOWN state
363 10 passing (1s)
364 $ docker-compose down
365 Container compse140-project-service2-1 Stopping
366 Container tests Stopping
367 Container compse140-project-apigateway-1 Stopping
368 Container tests Stopped
369 Container tests Removing
370 Container tests Removed
371 Container nginx Stopping
372 Container nginx Stopped
373 Container nginx Removing
374 Container nginx Removed
375 Container shutdown-service Stopping
376 Container compse140-project-service2-1 Stopped
377 Container compse140-project-service2-1 Removing
378 Container compse140-project-service2-1 Removed
379 Container compse140-project-apigateway-1 Stopped
380 Container compse140-project-apigateway-1 Removing
381 Container compse140-project-apigateway-1 Removed
382 Container compse140-project-service1-3 Stopping
383 Container compse140-project-service1-1 Stopping
384 Container compse140-project-service1-2 Stopping
385 Container shutdown-service Stopped
386 Container shutdown-service Removing
387 Container shutdown-service Removed
388 Container compse140-project-service1-3 Stopped
389 Container compse140-project-service1-3 Removing
390 Container compse140-project-service1-2 Stopped
391 Container compse140-project-service1-2 Removing
392 Container compse140-project-service1-1 Stopped
393 Container compse140-project-service1-1 Removing
394 Container compse140-project-service1-3 Removed
395 Container compse140-project-service1-2 Removed
396 Container compse140-project-service1-1 Removed
397 Network compse140-project_loadbalancing Removing
398 Network compse140-project_service-bridge Removing
399 Network compse140-project_loadbalancing Removed
400 Network compse140-project_service-bridge Removed
401 Cleaning up project directory and file based variables
402 Job succeeded
```

Reflections

My worst difficulties were with understanding the networking between the containers where the web servers were running and nginx. It took me a long time to read through materials to get things to work. Especially with the CI pipeline. I had a problem where my tests passed locally but failed in the CI due to odd networking errors but that was fixed by running the tests in their own container and using correct urls...

If I started to do the project again, I would not use a VM with a shared folder/storage with my main pc because it forced me to not be able to do the optional static analysis. On the design side I would put

the gitlab-runner into its own container too instead of locally running it in the VM but now I do not have the time to “make it right” while doing 5 courses + masters thesis..

Amount of effort (hours) used: 50 (really rough approximation)