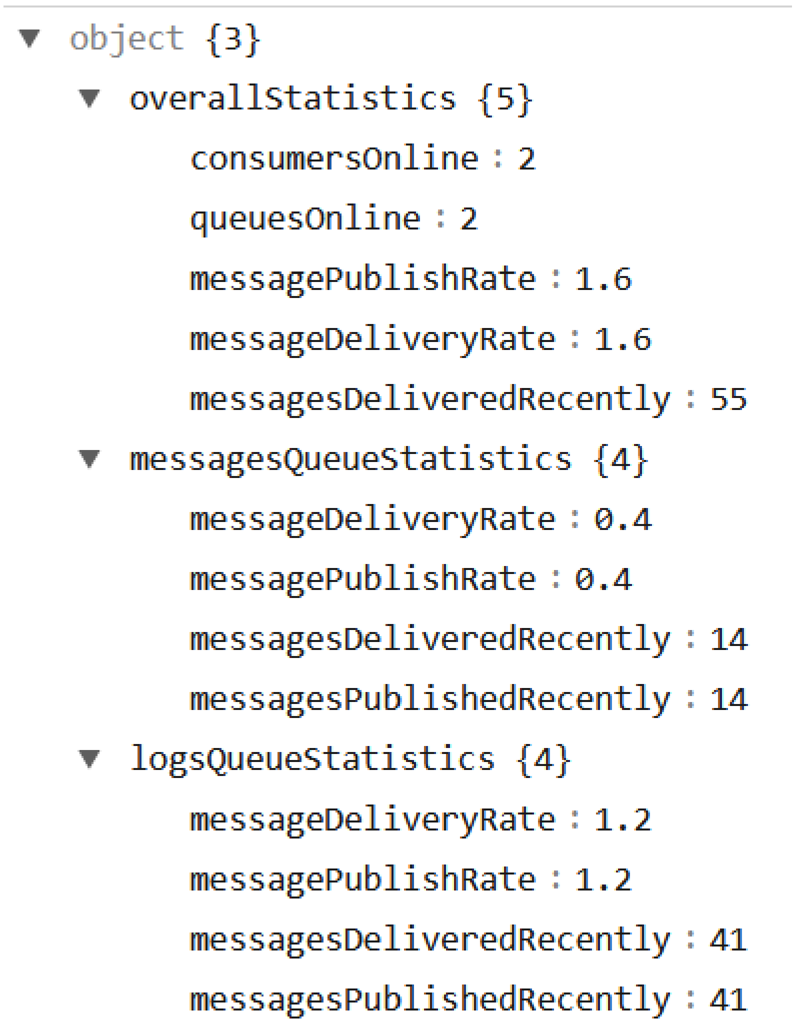
Joonas Pelttari

Instructions for the teaching assistant

Implemented optional features:

GET /mqstatistic
Returns a JSON that looks like Figure 1 (after copy-pasting it into some beautify tool)



```
▼ object {3}
    ▼ overallStatistics {5}
        consumersOnline : 2
        queuesOnline : 2
        messagePublishRate : 1.6
        messageDeliveryRate : 1.6
        messagesDeliveredRecently : 55
    ▼ messagesQueueStatistics {4}
        messageDeliveryRate : 0.4
        messagePublishRate : 0.4
        messagesDeliveredRecently : 14
        messagesPublishedRecently : 14
    ▼ logsQueueStatistics {4}
        messageDeliveryRate : 1.2
        messagePublishRate : 1.2
        messagesDeliveredRecently : 41
        messagesPublishedRecently : 41
```

Figure 1

Static analysis step. I am not sure if it was good enough, but I added a simple *dotnet format* step to the CI/CD pipeline. I didn't fix all the problems it warns about, so I marked that step as *"allow-failure: true"* so the pipeline doesn't fail but just marks that step as a warning.

Instructions for examiner to test the system.

git clone -b project git@github.com:JoonasPel/devops.git
cd <created folder>
docker-compose build –no-cache
docker-compose up -d

Project is also in the gitlab provided by the course, and its URL could be also used but in there the branch is named "main"!
git@compse140.devops-gitlab.rd.tuni.fi:joonas.pelttari/joonas.pelttari_private_project.git

After running the above commands, the project should be running (soon) and then it can be tested with the API defined in project instructions so commands like these should work:
(don't copy-paste them. At least in my system some notations like – and " change somehow)

Get messages:
curl localhost:8083/messages

Change state:
Available options are "PAUSED", "RUNNING" and "INIT". Unfortunately I was not able to make the "SHUTDOWN" work, so it does not exist.
curl localhost:8083/state -X PUT -d "PAUSED"

Get state:
curl localhost:8083/state

Get run-log:
curl localhost:8083/run-log

Get /mqstatistic (optional):
curl localhost:8083/mqstatistic

You can also go to the testing folder ./tests/APIgateway.Tests and run *dotnet test*

After:
docker-compose down

The actual CI/CD pipeline and gitlab runners can't really be tested because they are on my machine, but their logs can be seen in the gitlab provided by the course. If there is access to see them, I am not actually sure. I provide a photo below where TDD approach can be seen.
https://compse140.devops-gitlab.rd.tuni.fi/joonas.pelttari/joonas.pelttari_private_project

Dec 06, 2023

Implement API: GET /run-log Joonas Pelttari authored 22 hours ago	✓	`640a824f`
fix bug in test API: GET /run-log Joonas Pelttari authored 23 hours ago	✗	`da348b1c`
Implement tests for API: GET /run-log Joonas Pelttari authored 23 hours ago	✗	`5fbb5890`
Fix bug in API: GET /state test Joonas Pelttari authored 1 day ago	✓	`1583a150`
Implement API: GET+PUT /state Joonas Pelttari authored 1 day ago	✗	`70990f0d`
service1 listens for state changes and acts according Joonas Pelttari authored 1 day ago	✗	`8e9ee7ea`
Implement tests for API: GET+PUT /state Joonas Pelttari authored 1 day ago	✗	`fd66b4a0`
use baseUrl correctly Joonas Pelttari authored 1 day ago	✓	`f56dabea`

2. Description of the CI/CD pipeline

• Version management; use of branches etc:
CI/CD pipeline starts when code is pushed to main branch in Gitlab. I used my Virtualbox+Ubuntu as the Gitlab Runner and Shell as executor.

• Linting:
The first the step goes into APIgateway folder and runs some simple format check of the C# code. This step is marked as allow-failure: true because all the problems/warnings given are not fixed.

• Building tools:
The project is built in the next step with docker-compose build. I use sudo in the CI/CD pipeline but in my Virtualbox+Ubuntu I allowed the "gitlab-runner" user to use *sudo* with and only with docker-compose -command.

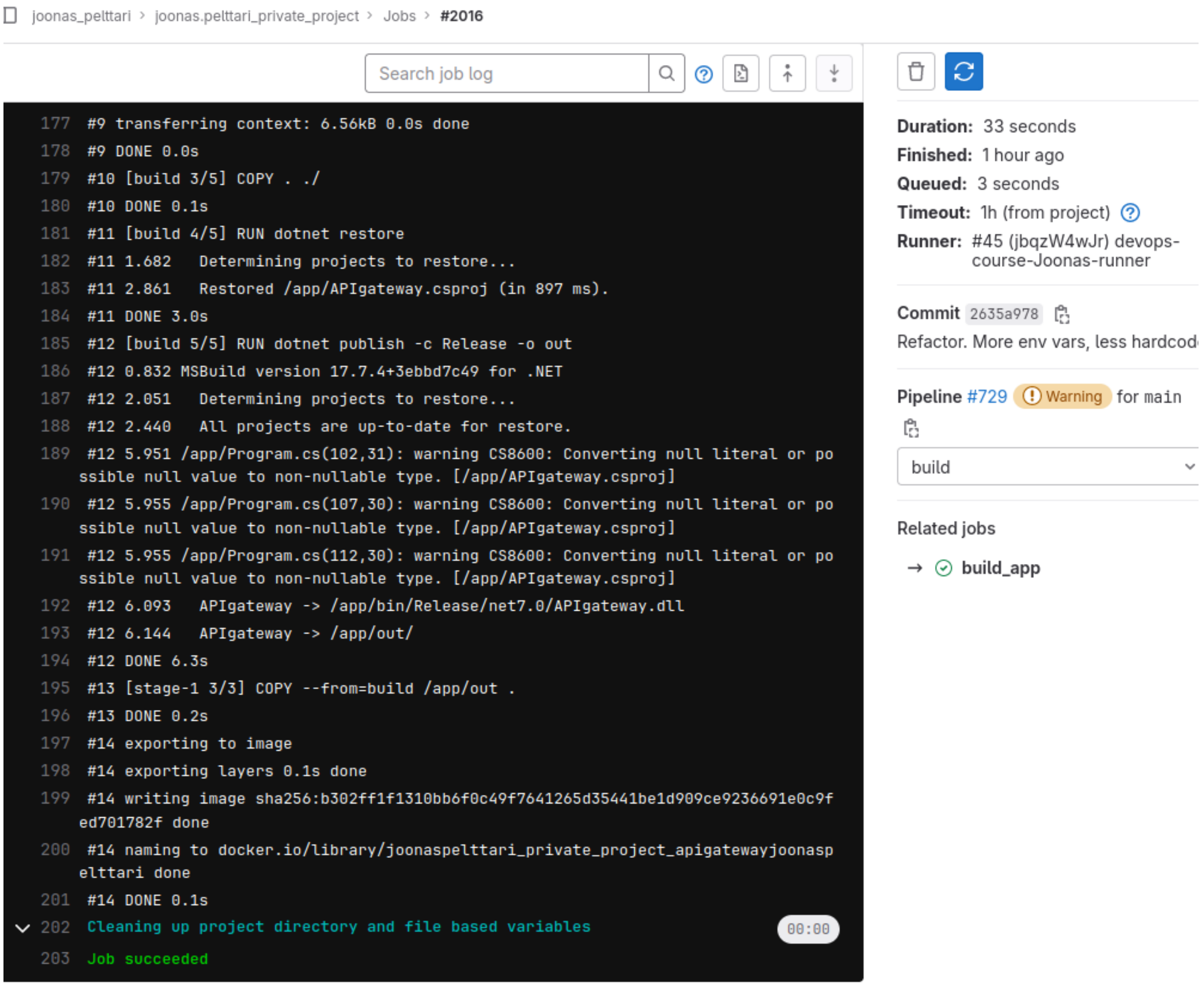
• Testing; tools and test cases:
For testing I used NUnit and RestSharp(.NET http library). Test-cases run some basic tests to the APIgateway testing all the API paths GET /messages, PUT /state, GET /state, GET /run-log, GET /mqstatistic. These were done with Test Driven Development approach. This test step is the third step and in here pipeline runs docker-compose up, then tests and afterwards closes the app with docker-compose down regardless of the test result.

• Deployment:
Deploying is just done to the same Virtualbox+Ubuntu with docker-compose up.

3. Example runs of the pipeline

Some logs of passing the pipeline:

Build step:



joonas_pelttari > joonas.pelttari_private_project > Jobs > #2016

```
#9 transferring context: 6.56kB 0.0s done
#9 DONE 0.0s
#10 [build 3/5] COPY . ./
#10 DONE 0.1s
#11 [build 4/5] RUN dotnet restore
#11 1.682   Determining projects to restore...
#11 2.861   Restored /app/APIgateway.csproj (in 897 ms).
#11 DONE 3.0s
#12 [build 5/5] RUN dotnet publish -c Release -o out
#12 0.832 MSBuild version 17.7.4+3ebbd7c49 for .NET
#12 2.051   Determining projects to restore...
#12 2.440   All projects are up-to-date for restore.
#12 5.951 /app/Program.cs(102,31): warning CS8600: Converting null literal or po
ssible null value to non-nullable type. [/app/APIgateway.csproj]
#12 5.955 /app/Program.cs(107,30): warning CS8600: Converting null literal or po
ssible null value to non-nullable type. [/app/APIgateway.csproj]
#12 5.955 /app/Program.cs(112,30): warning CS8600: Converting null literal or po
ssible null value to non-nullable type. [/app/APIgateway.csproj]
#12 6.093   APIgateway -> /app/bin/Release/net7.0/APIgateway.dll
#12 6.144   APIgateway -> /app/out/
#12 DONE 6.3s
#13 [stage-1 3/3] COPY --from=build /app/out .
#13 DONE 0.2s
#14 exporting to image
#14 exporting layers 0.1s done
#14 writing image sha256:b302ff1f1310bb6f0c49f7641265d35441be1d909ce9236691e0c9f
ed701782f done
#14 naming to docker.io/library/joonaspelttari_private_project_apigatewayjoonasp
elttari done
#14 DONE 0.1s
Cleaning up project directory and file based variables                00:00
Job succeeded
```

Duration: 33 seconds
Finished: 1 hour ago
Queued: 3 seconds
Timeout: 1h (from project)
Runner: #45 (jbqzW4wJr) devops-course-Joonas-runner

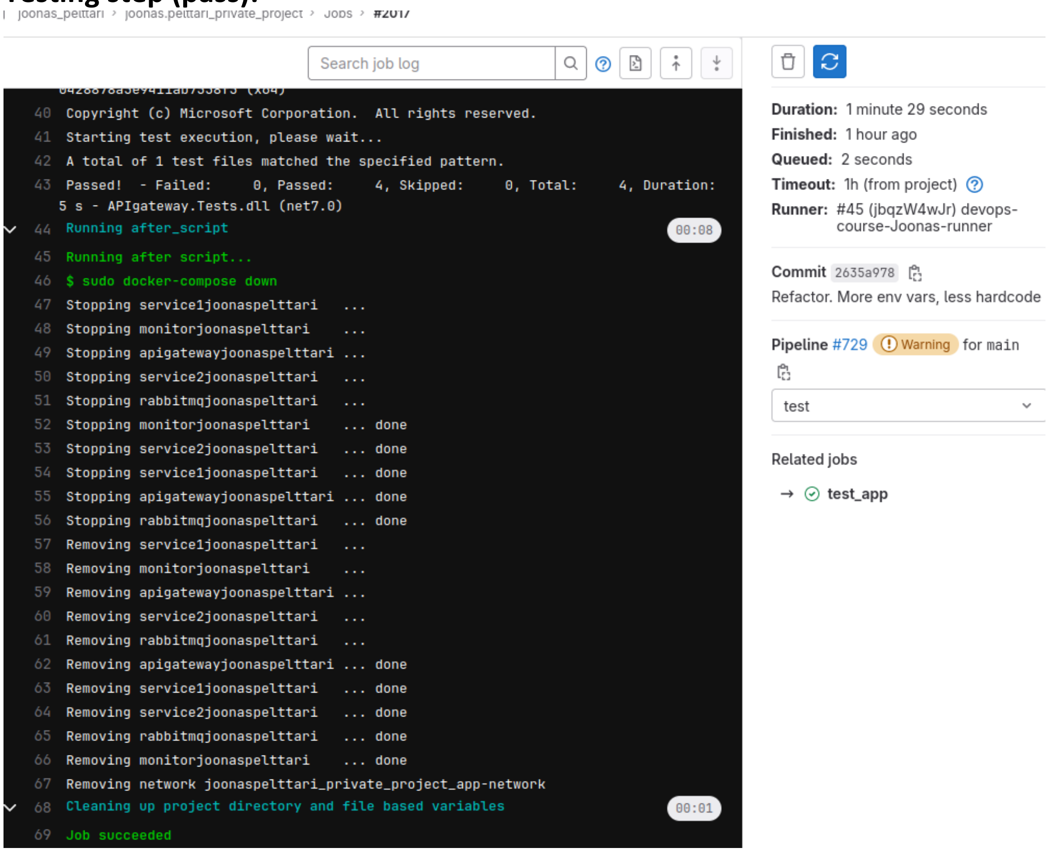
Commit 2635a978
Refactor. More env vars, less hardcod

Pipeline #729 Warning for main

Related jobs

→ build_app

Testing step (pass):


joonas_pelttari › joonas.pelttari_private_project › Jobs › #2017

```
Copyright (c) Microsoft Corporation.  All rights reserved.
Starting test execution, please wait...
A total of 1 test files matched the specified pattern.
Passed!  - Failed:     0, Passed:     4, Skipped:     0, Total:     4, Duration:
5 s - APIgateway.Tests.dll (net7.0)
Running after_script                                                          00:08
Running after script...
$ sudo docker-compose down
Stopping service1joonaspelttari   ...
Stopping monitorjoonaspelttari    ...
Stopping apigatewayjoonaspelttari ...
Stopping service2joonaspelttari   ...
Stopping rabbitmqjoonaspelttari   ...
Stopping monitorjoonaspelttari    ... done
Stopping service2joonaspelttari   ... done
Stopping service1joonaspelttari   ... done
Stopping apigatewayjoonaspelttari ... done
Stopping rabbitmqjoonaspelttari   ... done
Removing service1joonaspelttari   ...
Removing monitorjoonaspelttari    ...
Removing apigatewayjoonaspelttari ...
Removing service2joonaspelttari   ...
Removing rabbitmqjoonaspelttari   ...
Removing apigatewayjoonaspelttari ... done
Removing service1joonaspelttari   ... done
Removing service2joonaspelttari   ... done
Removing rabbitmqjoonaspelttari   ... done
Removing monitorjoonaspelttari    ... done
Removing network joonaspelttari_private_project_app-network
Cleaning up project directory and file based variables                        00:01
Job succeeded
```

Duration: 1 minute 29 seconds
Finished: 1 hour ago
Queued: 2 seconds
Timeout: 1h (from project)
Runner: #45 (jbqzW4wJr) devops-course-Joonas-runner

Commit 2635a978
Refactor. More env vars, less hardcode

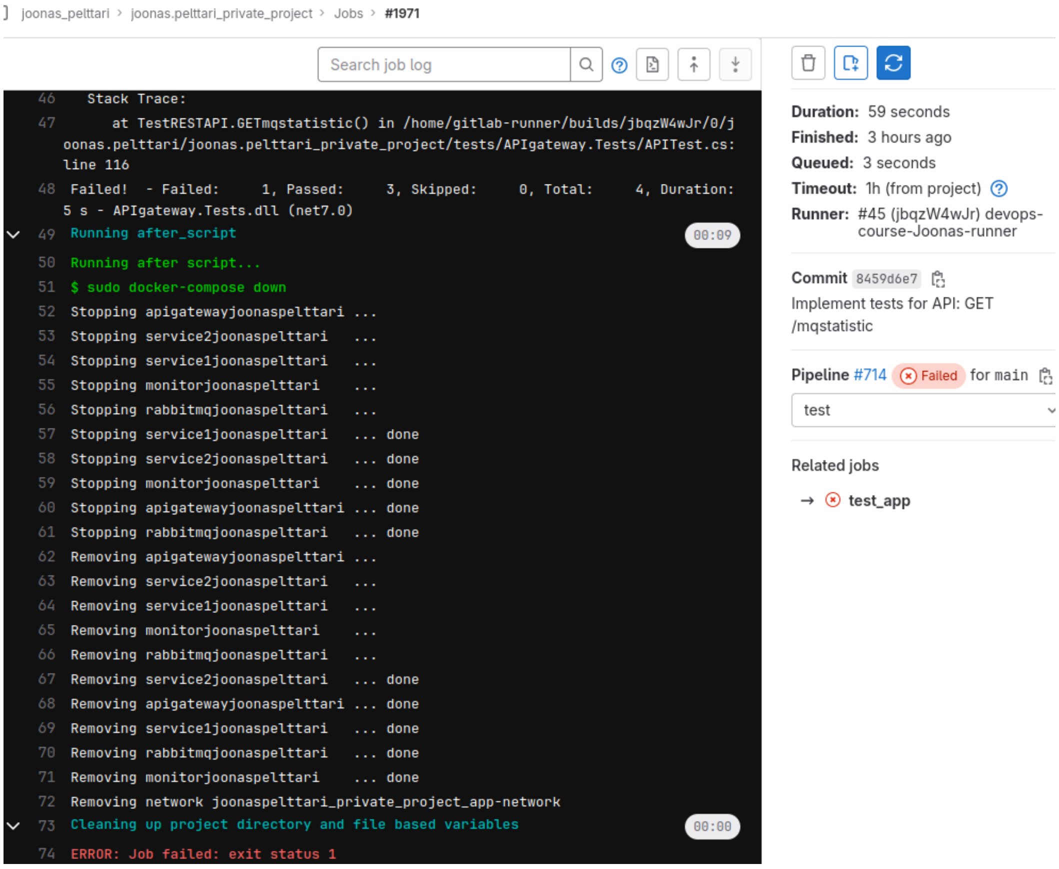
Pipeline #729 Warning for main

test

Related jobs
→ ✓ test_app

Example of a failing pipeline logs (test step failed):

Fail happened because tests for GET /mqstatistic were implemented, actual feature not.


joonas_pelttari › joonas.pelttari_private_project › Jobs › #1971

```
  Stack Trace:
     at TestRESTAPI.GETmqstatistic() in /home/gitlab-runner/builds/jbqzW4wJr/0/j
oonas.pelttari/joonas.pelttari_private_project/tests/APIgateway.Tests/APITest.cs:
line 116
Failed!  - Failed:     1, Passed:     3, Skipped:     0, Total:     4, Duration:
5 s - APIgateway.Tests.dll (net7.0)
Running after_script                                                          00:09
Running after script...
$ sudo docker-compose down
Stopping apigatewayjoonaspelttari ...
Stopping service2joonaspelttari   ...
Stopping service1joonaspelttari   ...
Stopping monitorjoonaspelttari    ...
Stopping rabbitmqjoonaspelttari   ...
Stopping service1joonaspelttari   ... done
Stopping service2joonaspelttari   ... done
Stopping monitorjoonaspelttari    ... done
Stopping apigatewayjoonaspelttari ... done
Stopping rabbitmqjoonaspelttari   ... done
Removing apigatewayjoonaspelttari ...
Removing service2joonaspelttari   ...
Removing service1joonaspelttari   ...
Removing monitorjoonaspelttari    ...
Removing rabbitmqjoonaspelttari   ...
Removing service2joonaspelttari   ... done
Removing apigatewayjoonaspelttari ... done
Removing service1joonaspelttari   ... done
Removing rabbitmqjoonaspelttari   ... done
Removing monitorjoonaspelttari    ... done
Removing network joonaspelttari_private_project_app-network
Cleaning up project directory and file based variables                        00:00
ERROR: Job failed: exit status 1
```

Duration: 59 seconds
Finished: 3 hours ago
Queued: 3 seconds
Timeout: 1h (from project)
Runner: #45 (jbqzW4wJr) devops-course-Joonas-runner

Commit 8459d6e7
Implement tests for API: GET /mqstatistic

Pipeline #714 Failed for main

test

Related jobs
→ ✗ test_app

4. Reflections

Main learnings and worst difficulties

Difficulties:

My worst difficulties were about the gitlab runner and things around it. Mostly getting the certificate to work, so the runner could be registered. And because I am not super familiar Linux user I had to play around the permissions also so gitlab runner could actually run the docker-compose commands. As described earlier, I have the gitlab runner -user the permission to use sudo with that one command only. I used sudo visudo for this.

I was not able to make the PUT /state "SHUTDOWN" work so this one was of course a big difficulty too. I tried to open a cli from the APIgateway (.NET code) and execute docker-compose down from there but without success.

One bug I had was that sometimes the app didn't start because a single container was unhealthy. I managed to understand that this might happen because of a permission problem with rabbitmq and some .erlang cookie. The problem happened to be that my healthcheck in docker-compose.yaml checking the port availability of rabbitmq (to start other containers after rabbitmq is ready = port available) had too low interval. It for some reason interrupted something in the rabbitmq when it was starting. Interval of 3s was the biggest that I got the problem with and after it never at all. I decided to use 10s. A better understanding of the actual problem would of course be a better solution.

I also had a lot of bugs with the PUT /state but I managed to get rid of them (mostly at least). Bugs like giving "RUNNING" or "INIT" multiple times in a row and then having multiple loggers in the service1 existed at first.

Learnings:

I learned a lot in this project and the main ones were probably about how to setup a gitlab runner and how to make pipeline with different steps. I believe with the current skills I would be able to just learn about some other step I want and include it into the pipeline. I think that the testing step was super good and including something like that in my future projects would be a good option.

Amount effort (hours) used

40-45 hours