

Instructions for the teaching assistant

Implemented optional features

None.

Instructions for examiner to test the system.

There isn't anything special. Running the course-expected command (below) should get you up and running.

```
docker compose build --no-cache
docker compose up -d
```

Notice that, `GET /request` and `PUT /state SHUTDOWN` do not work.

Data about the platform you used in development (hardware, CPU architecture, operating system, version of docker and docker-compose)

AMD 3000 series cpu, X86_64, Ubuntu running on host Windows 10 through WSL2, Docker version 27.5.1, build 9f9e405 (docker compose included)

Description of the CI/CD pipeline

The pipeline only includes the three basic steps.

- Building is done entirely through docker (compose). All build tools are installed within the containers.
- Tests run in a separate container. They use the Jest framework, which seemed like a sensible choice since my api is built using node+express.
- Deployment is simply a local deploy which runs `docker compose up -d`

Example runs of the pipeline

I've tried to include only the relevant parts in the logs.

Failing

```
Running with gitlab-runner 17.8.3 (690ce25c)
  on runnerer t3_3JcUxt, system ID: s_bd30cf973e70
Preparing the "shell" executor 00:00
Using Shell (bash) executor...
Preparing environment 00:00
Running on KONE...
Getting source from Git repository 00:01
Fetching changes with git depth set to 20...
Reinitialized existing Git repository in /home/jykae/repot/tuni-
devops/builds/t3_3JcUxt/0/jykae/devops/.git/
Checking out 8695fa47 as detached HEAD (ref is project)...
```

```

Skipping Git submodules setup
Downloading artifacts 00:00
Downloading artifacts for build-job (12709)...
Runtime platform                                arch=amd64 os=linux pid=9542
revision=690ce25c version=17.8.3
Downloading artifacts from coordinator... ok      host=compse140.devops-
gitlab.rd.tuni.fi id=12709 responseStatus=200 OK
... <snip> ...
$ docker compose run --rm tests npm test
Container devops-service2-1 Running
Container devops-service1-1 Running
Container devops-service1-2 Running
Container devops-service1-3 Running
Container devops-state-manager-1 Running
> api-tests@1.0.0 test
> jest
FAIL tests/state.test.js
  State Manager API Tests
    ✓ GET /state should succeed regardless of auth (30 ms)
    ✓ PUT /state without authorization should return 401 (4 ms)
    ✓ PUT /state with authorization should return 200 (30 ms)
    ✓ GET /state should return RUNNING after change (3 ms)
    ✗ GET /request should return text (5 ms)
    ✓ GET /run-log should return text (4 ms)
  • State Manager API Tests › GET /request should return text
    expect(received).toBe(expected) // Object.is equality
    Expected: 200
    Received: 404
      34 |     test('GET /request should return text', async () => {
      35 |         const response = await request(BASE_URL).get('/request');
    > 36 |         expect(response.statusCode).toBe(200);
          |                                     ^
      37 |         expect(response.type).toBe('text/plain');
      38 |     });
      39 |
      at Object.toBe (tests/state.test.js:36:37)
Test Suites: 1 failed, 1 total
Tests:      1 failed, 5 passed, 6 total
Snapshots:  0 total
Time:       0.49 s
Ran all test suites.
Running after_script 00:03
Running after script...
$ docker compose down
... <snip> ...
Cleaning up project directory and file based variables 00:00
ERROR: Job failed: exit status 1

```

Successful run

```

Running with gitlab-runner 17.8.3 (690ce25c)
  on runnerer t3_3JcUxt, system ID: s_bd30cf973e70
Preparing the "shell" executor 00:00
Using Shell (bash) executor...
Preparing environment 00:00
Running on KONE...
Getting source from Git repository 00:00
Fetching changes with git depth set to 20...
Reinitialized existing Git repository in /home/jykae/repot/tuni-
devops/builds/t3_3JcUxt/0/jykae/devops/.git/
Checking out 11e74dd7 as detached HEAD (ref is project)...
Skipping Git submodules setup
Downloading artifacts 00:01
Downloading artifacts for build-job (12902)...
Runtime platform                                arch=amd64 os=linux pid=20537
revision=690ce25c version=17.8.3
Downloading artifacts from coordinator... ok      host=compse140.devops-
gitlab.rd.tuni.fi id=12902 responseStatus=200 OK
... <snip> ...
$ docker compose run --rm tests npm test
Container devops-service2-1 Running
Container devops-service1-1 Running
Container devops-service1-2 Running
Container devops-service1-3 Running
Container devops-state-manager-1 Running
> api-tests@1.0.0 test
> jest
PASS tests/state.test.js
  State Manager API Tests
    ✓ GET /state should succeed regardless of auth (28 ms)
    ✓ PUT /state without authorization should return 401 (3 ms)
    ✓ PUT /state with authorization should return 200 (26 ms)
    ✓ GET /state should return RUNNING after change (3 ms)
    ✓ GET /run-log should return text (3 ms)
Test Suites: 1 passed, 1 total
Tests:      5 passed, 5 total
Snapshots:  0 total
Time:       0.397 s
Ran all test suites.
Running after_script 00:04
Running after script...
$ docker compose down
... <snip> ...
Cleaning up project directory and file based variables 00:00
Job succeeded

```

Reflections

By far my number one issue was not setting enough time aside for the project. This made the whole project a rush job where my decisions were mainly to use the first solution that worked. Furthermore, there are things I left unimplemented (shutdown and /request). If I started working on the project earlier I might have been able

to implement these and further develop the pipeline. This is entirely on me, the course side of things is in my opinion fine.

Amount effort (hours) used

15-20 hours, most of which was spent trying things out "offline" (that is, you won't see this activity in git logs).