

1. Instructions for the teaching assistant

Implemented optional features.

1. Implemented a static analysis step in the pipeline by using SonarQube.
2. Implemented *GET /mqstatistic* endpoint in the system
3. Implemented deployment to an external cloud (AWS)

Instructions for examiner to test the system.

1. To run the system's basic requirements,

1.1 Clone the project using below url.

```
git clone -b project https://course-gitlab.tuni.fi/comp.se.140-fall2023\_2023-2024/dcthra.git
```

1.2 Change directory to the project.

```
cd dcthra
```

1.3 Build the system using the below command.

```
docker-compose build --no-cache
```

1.4 Run the system using the below command.

```
docker-compose up -d
```

2. To test the system's basic requirements

2.1 Use curl/Postman to test the system.

Start the system using the below curl command.

```
curl localhost:8083/state -X PUT -d "INIT" -H "Content-Type: text/plain" -H "Accept: text/plain"
```

Pause the system using the below curl command.

```
curl localhost:8083/state -X PUT -d "PAUSED" -H "Content-Type: text/plain" -H "Accept: text/plain"
```

Resume the system using the below curl command.

```
curl localhost:8083/state -X PUT -d "RUNNING" -H "Content-Type: text/plain" -H "Accept: text/plain"
```

Shut down the system using the below curl command.

```
curl localhost:8083/state -X PUT -d "SHUTDOWN" -H "Content-Type: text/plain" -H "Accept: text/plain"
```

2.2 To test the GET `/mqstatistic` endpoint in the system,

```
curl --location 'localhost:8083/mqstatistic'
```

3. Test SonarQube Integration.

2.1 Run SonarQube docker container using the below command.

```
docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube
```

2.2 Login to the SonarQube by using default admin/admin credentials and generate a new user(admin) token by navigating to User > My Account > Security [1].

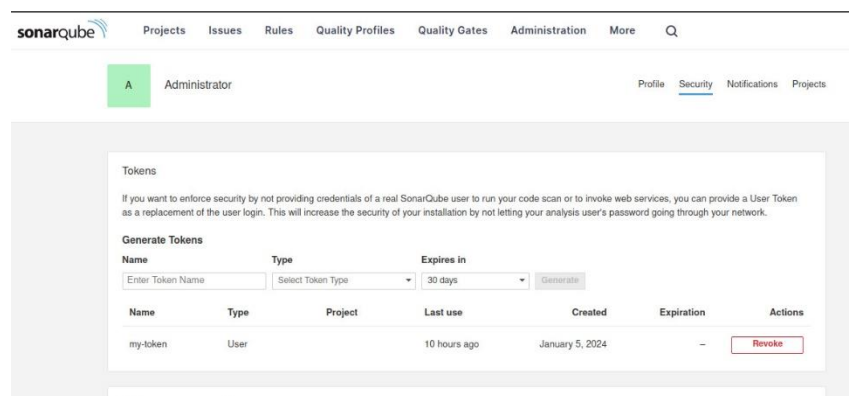


Figure: SonarQube Account page

2.3 Update the `-Dsonar.login= <new_token>` placeholder value with the previously generated token under the sonarqube-check stage in the `.gitlab-ci.yml` file.

```

tharindu_rathgamaguruge > tharindu.rathgamaguruge_private_project > Pipeline Editor
Browse templates Help
21 | - pip install coverage requests flask && coverage run -m unittest discover -s tests -v && coverage xml
22 |
23 |
24 | sonarqube-check:
25 |   stage: sonarqube-check
26 |   image:
27 |     name: sonarsource/sonar-scanner-cli:latest
28 |     entrypoint: [""]
29 |   variables:
30 |     SONAR_USER_HOME: "${CI_PROJECT_DIR}/.sonar"
31 |     GIT_DEPTH: "0"
32 |   cache:
33 |     key: "${CI_JOB_NAME}"
34 |     paths:
35 |       - .sonar/cache
36 |   script:
37 |     - sonar-scanner -Dsonar.projectKey=api_service -Dsonar.coverage.jacoco.xmlReportPaths=coverage.xml -Dsonar.projectName=api_service -Dsonar.sources=api_gateway_service -Dsonar.tests=tests -Dsonar.host.url=http://localhost:9000 -Dsonar.login=squ_c2f8edf892d2897eaea2036d6b88cdee9fc03993
38 |     allow_failure: true
39 |     only:
40 |       - project
41 |
42 | deploy:
43 |   stage: deploy
44 |   script:

```

Figure: sonarqube-check stage in gitlab-ci.yml

P.S- Here, I have used gitlab-runner with a specific tag. Therefore, you may need to change the tag according to your runner tag to test the system with sonarqube and gitlab-ci.

2.4 Add below configuration into the config.toml file in /etc/gitlab-runner location.
[runners.docker]

```

.....
.....
volumes = ["/var/run/docker.sock:/var/run/docker.sock", "/cache"]
.....

```

2. Description of the CI/CD pipeline

- VCS and Branches:

In this project Git was used as VCS and Gitlab as centralized VCS platform. Followed muti remote repositories management method. One repository to build, test and deploy the system more efficiently with Gitlab CI and Other gitlab to keep the final code.

```

tharindu-107455@107455-001L8:~/devopsProject/dcthra$ git remote -v
ci-origin https://tharindu.rathgamaguruge:glpat-Lx9dwbfxhMEXZD-HsnDR@compse140.devops-gitlab.rd.tuni.fi/tharindu.rathgamaguruge/tharindu.rathgamaguruge_private_project.git (fetch)
ci-origin https://tharindu.rathgamaguruge:glpat-Lx9dwbfxhMEXZD-HsnDR@compse140.devops-gitlab.rd.tuni.fi/tharindu.rathgamaguruge/tharindu.rathgamaguruge_private_project.git (push)
origin https://dcthra:glpat-tExTy-vmLn4NWDbqCHyP@course-gitlab.tuni.fi/comp.se.140-fall2023_2023-2024/dcthra.git (fetch)
origin https://dcthra:glpat-tExTy-vmLn4NWDbqCHyP@course-gitlab.tuni.fi/comp.se.140-fall2023_2023-2024/dcthra.git (push)
tharindu-107455@107455-001L8:~/devopsProject/dcthra$

```

Figure: remote repositories

Created project branch in both repositories from the exercise2 branch and used project branch to do changes during the implementations.

- Building tools

Used Python and JavaScript as main two programming languages in the project. npm and pip as package management tools. Due to the single script files, no build tools used to build the applications.

- Testing; tools and test cases

Testings mainly based on the api gateway service.

Test framework: - unittest (Python)

Test cases

1. To test GET /messages endpoint and expected response code.
user should be able to get expected response from the system.
2. To test PUT /state endpoint with
 - 2.1 by setting INIT as data
 - 2.2 by setting PAUSED as data.
 - 2.3 by setting RUNNING as data.
 - 2.4 by setting SHUTDOWN as data
 - 2.5 by setting dummy data (FAKE) as data

User should be able to get expected responses (status updates) according to the request.

3. To test GET /state endpoint and expected response.
User should be able to get current state according to the request.
4. To test GET /run-log endpoint and expected response.
User should be able to get the status change in the response
5. To test GET /mqstatistics endpoint and expected response.
user should be able to get expected response from the system.

- Packing

packaging done with docker.

- Deployment

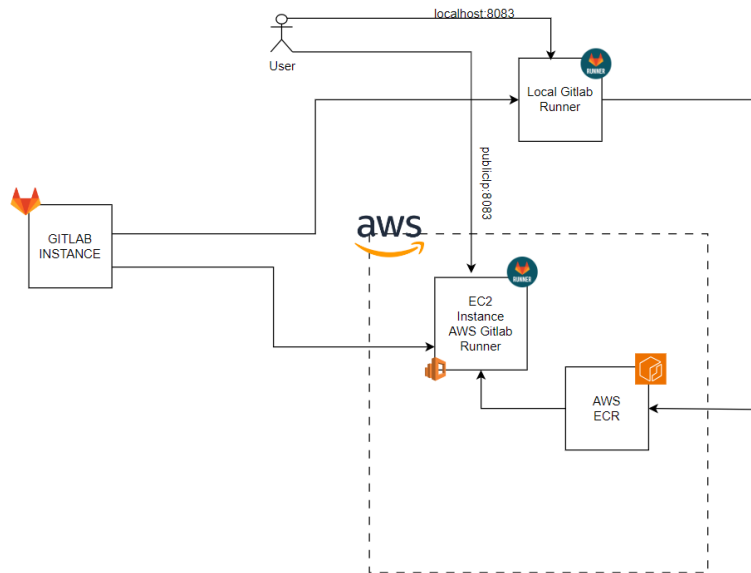


Figure: Deployment Architecture

1. Local deployment done with docker-compose using docker-compose up -d command.
2. AWS deployment
 - 2.1 Created an aws instance on the aws cloud.
 - 2.2 Created ECR(Elastic Container Registry) private repositories for service1, service2, api_gateway_service and monitoring_service (To ensure redundancy and high availability)
 - 2.3 Generated access key and secret key with granting permission for ec2 instances and ecr access.
 - 2.4 Added secret key, access key data in gitlab CI/CD variables (To ensure security)

Variables

Variables store information, like passwords and secret keys, that you can use in job scripts. Each project can define a maximum of 8000 variables. [Learn more.](#)

Variables can have several attributes. [Learn more.](#)

- Protected: Only exposed to protected branches or protected tags.
- Masked: Hidden in job logs. Must match masking requirements.
- Expanded: Variables with `!` will be treated as the start of a reference to another variable.

CI/CD Variables < 4	Value	Environments	Actions
AWS_ACCESS_KEY_ID Expanded	*****	All (default)	Edit Delete
AWS_DEFAULT_REGION Expanded	*****	All (default)	Edit Delete
AWS_SECRET_ACCESS_KEY Expanded	*****	All (default)	Edit Delete
ECR_REPO_URL Expanded	*****	All (default)	Edit Delete

Figure: GitLab CI/CD Variables

- 2.5 Installed and registered gitlab runner on ec2 instance.
- 2.6 Added bash script to push images into ecr.
- 2.7 Added two stages to .gitlab-ci.yml to push the images to ecr and deploy docker containers using docker-compose to ec2 instance.

- Operating; monitoring

Did not implement

3. Example runs of the pipeline

3.1 Success Build Stage

tharindu_rathgamaguruge > tharindu.rathgamaguruge_private_project > Jobs > #3868

Search job log

146 #22 4.158 Downloading zipp-3.17.0-py3-none-any.whl (7.4 kB)

147 #22 4.227 Collecting MarkupSafe>=2.0

148 #22 4.238 Downloading MarkupSafe-2.1.3-cp39-cp39-musllinux_1_1_x86_64.whl (29 kB)

149 #22 4.417 Installing collected packages: zipp, urllib3, packaging, MarkupSafe, itsdangerous, idna, click, charset-normalizer, certifi, blinker, Werkzeug, requests, Jinja2, importlib-metadata, flask, docker

150 #22 5.440 Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.3 Werkzeug-3.0.1 blinker-1.7.0 certifi-2023.11.17 charset-normalizer-3.3.2 click-8.1.7 docker-7.0.0 flask-3.0.0 idna-3.6 importlib-metadata-7.0.1 itsdangerous-2.1.2 packaging-23.2 requests-2.31.0 urllib3-2.1.0 zipp-3.17.0

151 #22 5.440 WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: <https://pip.pypa.io/warnings/venv>

152 #22 5.563

153 #22 5.563 [notice] A new release of pip is available: 23.0.1 -> 23.3.2

154 #22 5.563 [notice] To update, run: pip install --upgrade pip

155 #22 DONE 5.8s

156 #33 [api_gateway_service] exporting to image

157 #33 exporting layers 0.2s done

158 #33 writing image sha256:2d78108c5e04d999c557a33f6b033f2b963468b42f3b275edfc9419365478e85 done

159 #33 naming to docker.io/library/tharindurathgamaguruge_private_project-api_gateway_service done

160 #33 DONE 0.2s

161 Cleaning up project directory and file based variables

162 Job succeeded

00:01

Duration: 21 seconds

Finished: just now

Queued: 2 seconds

Timeout: 1h (from project)

Runner: #95 (pmxUAAeyY)

Tags: dcthra

Commit 88991def

"updated responses of service1 and unit tests"

Pipeline #1272 Passed for project

build

Related jobs

→ build

Figure: Success build stage

3.2 Passed test case scenario

tharindu_rathgamaguruge > tharindu.rathgamaguruge_private_project > Jobs > #3869

Search job log

12 Checking out 88991def as detached HEAD (ref is project) *Figure: Failed test case scenario*
13 Skipping Git submodules setup
14 Executing "step_script" stage of the job script
15 Using docker image sha256:6091c7bd89fd2789606b49815b2b9ea1a9142ee6e8762089ab3975afd6784a6c for docker:latest with digest docker@sha256:1b9844d846ce3a6a6af7013e999a373112c3c0450aca49e155ae444526a2c45e ...
16 \$ docker-compose run --rm api_gateway_service python -m unittest discover /app -v
17 Container tharindurathgamaguruge_private_project-rabbitmq-1 Created
18 Container tharindurathgamaguruge_private_project-rabbitmq-1 Starting
19 Container tharindurathgamaguruge_private_project-rabbitmq-1 Started
20 test_get_messages (test_api_gateway_service.TESTSAPIGatewayService) ... ok
21 test_get_run_log (test_api_gateway_service.TESTSAPIGatewayService) ... ok
22 test_get_state (test_api_gateway_service.TESTSAPIGatewayService) ... ok
23 test_set_fake_state (test_api_gateway_service.TESTSAPIGatewayService) ... ok
24 test_set_init_state (test_api_gateway_service.TESTSAPIGatewayService) ... ok
25 test_set_pause_state (test_api_gateway_service.TESTSAPIGatewayService) ... ok
26 test_set_running_state (test_api_gateway_service.TESTSAPIGatewayService) ... ok
27 -----
28 Ran 7 tests in 0.015s
29 OK
30 Cleaning up project directory and file based variables
31 Job succeeded

Duration: 5 seconds
Finished: 2 minutes ago
Queued: 3 seconds
Timeout: 1h (from project)
Runner: #95 (pmxUAAeyY)
Tags: dcthra

Commit 88991def
"updated responses of service1 and unit tests"

Pipeline #1272 Passed for project
test

Related jobs
→ test

Figure: Passed test case scenario

3.3 Failed Test case scenario

Search or go to...

Project
tharindu.rathgamaguruge...
Pinned
Issues
Merge requests
Manage
Plan
Code
Build
Pipelines
Jobs
Pipeline editor
Pipeline schedules
Artifacts
Secure
Help

Search job log

16 \$ docker-compose run --rm api_gateway_service python -m unittest discover /app -v
17 Container tharindurathgamaguruge_private_project-rabbitmq-1 Created
18 Container tharindurathgamaguruge_private_project-rabbitmq-1 Starting
19 Container tharindurathgamaguruge_private_project-rabbitmq-1 Started
20 test_get_messages (test_api_gateway_service.TESTSAPIGatewayService) ... ok
21 test_get_run_log (test_api_gateway_service.TESTSAPIGatewayService) ... ok
22 test_get_state (test_api_gateway_service.TESTSAPIGatewayService) ... ok
23 test_set_state_other_than_shutdown (test_api_gateway_service.TESTSAPIGatewayService) ... ok
24 test_shutdown (test_api_gateway_service.TESTSAPIGatewayService) ... FAIL
25 =====
26 FAIL: test_shutdown (test_api_gateway_service.TESTSAPIGatewayService)
27 -----
28 Traceback (most recent call last):
29 File "/usr/local/lib/python3.9/unittest/mock.py", line 1336, in patched
30 return func(*newargs, **newkeywargs)
31 File "/app/test_api_gateway_service.py", line 28, in test_shutdown
32 self.assertEqual(response.content_type, 'text/plain')
33 AssertionError: 'application/json' != 'text/plain'
34 - application/json
35 + text/plain
36 -----
37 Ran 5 tests in 0.019s
38 FAILED (failures=1)
39 Cleaning up project directory and file based variables
40 ERROR: Job failed: exit code 137

Duration: 15 seconds
Finished: 6 hours ago
Queued: 2 seconds
Timeout: 1h (from project)
Runner: #95 (pmxUAAeyY)
Tags: dcthra

Commit ac58ff4c
"updated services and api gateway with shutdown and added unit test for shutdown state endpoint"

Pipeline #1242 Failed for project
test

Related jobs
→ test

Figure: Failed test case scenario

3.4 SonarQube check stage

The screenshot shows a CI pipeline job for the SonarQube check stage. The job log on the left displays the following information:

```
113 INFO: ----- Run sensors on project
114 INFO: Sensor Analysis Warnings Import [csharp]
115 INFO: Sensor Analysis Warnings Import [csharp] (done) | time=0ms
116 INFO: Sensor Zero Coverage Sensor
117 INFO: Sensor Zero Coverage Sensor (done) | time=9ms
118 INFO: CPD Executor Calculating CPD for 2 files
119 INFO: CPD Executor CPD calculation finished (done) | time=9ms
120 INFO: Analysis report generated in 183ms, dir size=177.3 kB
121 INFO: Analysis report compressed in 36ms, zip size=36.4 kB
122 INFO: Analysis report uploaded in 33ms
123 INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?ids=api_service
124 INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
125 INFO: More about the report processing at http://localhost:9000/api/cg/task?id=AVzaDYF4lszETI98heBu
126 INFO: Analysis total time: 8.579 s
127 INFO: -----
128 INFO: EXECUTION SUCCESS
129 INFO: -----
130 INFO: Total time: 9.966s
131 INFO: Final Memory: 34M/128M
132 INFO: -----
133 Saving cache for successful job
134 Creating cache sonarqube-check-non_protected...
135 .sonar/cache: found 58 matching artifact files and directories
136 Archive is up to date!
137 Created cache
138 Cleaning up project directory and file based variables
139 Job succeeded
```

The right sidebar provides additional details:

- Duration: 18 seconds
- Finished: 1 hour ago
- Queued: 2 seconds
- Timeout: 1h (from project)
- Runner: #95 (pmxUAAeyY)
- Tags: [dcetra](#)
- Commit: [b39a3c6f](#)
- Update .gitlab-ci.yml file
- Pipeline #1359 Passed for project
- sonarqube-check
- Related jobs: [sonarqube-check](#)

Figure: SonarQube check stage

3.5 Images push to ECR

The screenshot shows a CI pipeline job for the AWS Image push stage. The job log on the left displays the following information:

```
86 Configure a credential helper to remove this warning. See
87 https://docs.docker.com/engine/reference/commandline/login/#credentials-store
88 Login Succeeded
89 The push refers to repository [767397724383.dkr.ecr.eu-north-1.amazonaws.com/service1]
90 7be78d72ef77: Preparing
91 af06b536393a: Preparing
92 c3af892e725f: Preparing
93 8806d03b46a0: Preparing
94 d25e51999115: Preparing
95 7b3bd0b27e31: Preparing
96 ec4d864ac810: Preparing
97 5af4f8f59b76: Preparing
98 7b3bd0b27e31: Waiting
99 ec4d864ac810: Waiting
100 5af4f8f59b76: Waiting
101 d25e51999115: Layer already exists
102 7be78d72ef77: Layer already exists
103 8806d03b46a0: Layer already exists
104 af06b536393a: Layer already exists
105 c3af892e725f: Layer already exists
106 7b3bd0b27e31: Layer already exists
107 ec4d864ac810: Layer already exists
108 5af4f8f59b76: Layer already exists
109 latest: digest: sha256:1a1afbcf382f0dcf87540e063bd623103adf0f5141a2adcc311a338b0c19c048 size: 1993
110 The push refers to repository [767397724383.dkr.ecr.eu-north-1.amazonaws.com/service2]
```

The right sidebar provides additional details:

- Duration: 13 seconds
- Finished: 51 minutes ago
- Queued: 0 seconds
- Timeout: 1h (from project)
- Runner: #95 (pmxUAAeyY)
- Tags: [dcetra](#)
- Commit: [b39a3c6f](#)
- Update .gitlab-ci.yml file
- Pipeline #1359 Passed for project
- aws-image-push
- Related jobs: [aws-image-push](#)

Figure: AWS Image push

3.6 Deploy stage.

tharindu_rathgamaguruge > tharindu.rathgamaguruge_private_project > Jobs > #3870

Search job log

with digest docker@sha256:1b9844d846ce3a6a6af7813e999a373112c3c0450aca49e155ae444526a2c45e ...

```
16 $ docker-compose up -d
17 Container tharindurathgamaguruge_private_project-rabbitmq-1 Running
18 Container tharindurathgamaguruge_private_project-service1-1 Creating
19 Container tharindurathgamaguruge_private_project-monitoring_service-1 Creating
20 Container tharindurathgamaguruge_private_project-service2-1 Creating
21 Container tharindurathgamaguruge_private_project-api_gateway_service-1 Creating
22 Container tharindurathgamaguruge_private_project-monitoring_service-1 Created
23 Container tharindurathgamaguruge_private_project-service2-1 Created
24 Container tharindurathgamaguruge_private_project-service1-1 Created
25 Container tharindurathgamaguruge_private_project-api_gateway_service-1 Created
26 Container tharindurathgamaguruge_private_project-api_gateway_service-1 Starting
27 Container tharindurathgamaguruge_private_project-service2-1 Starting
28 Container tharindurathgamaguruge_private_project-service1-1 Starting
29 Container tharindurathgamaguruge_private_project-monitoring_service-1 Starting
30 Container tharindurathgamaguruge_private_project-service1-1 Started
31 Container tharindurathgamaguruge_private_project-monitoring_service-1 Started
32 Container tharindurathgamaguruge_private_project-service2-1 Started
33 Container tharindurathgamaguruge_private_project-api_gateway_service-1 Started
34 Cleaning up project directory and file based variables
35 Job succeeded
```

08:08

Duration: 5 seconds

Finished: 22 minutes ago

Queued: 2 seconds

Timeout: 1h (from project)

Runner: #95 (pmxUAAeyY)

Tags: dchtra

Commit 88991def

"updated responses of service1 and unit tests"

Pipeline #1272 Passed for project

deploy

Related jobs

→ deploy

Figure: Deployment to local machine stage

3.7 Deployment to AWS

tharindu_rathgamaguruge > tharindu.rathgamaguruge_private_project > Jobs > #4281

Search job log

d04b35da41fa Extracting [=====] 5.636MB/6.391MB
d04b35da41fa Extracting [=====] 6.226MB/6.391MB
d04b35da41fa Extracting [=====] 6.391MB/6.391MB
d04b35da41fa Pull complete
api_gateway_service Pulled
Network tharindurathgamaguruge_private_project-mynetwork Creating
Network tharindurathgamaguruge_private_project-mynetwork Created
Container tharindurathgamaguruge_private_project-rabbitmq-1 Creating
Container tharindurathgamaguruge_private_project-rabbitmq-1 Created
Container tharindurathgamaguruge_private_project-service2-1 Creating
Container tharindurathgamaguruge_private_project-api_gateway_service-1 Creating
Container tharindurathgamaguruge_private_project-service1-1 Creating
Container tharindurathgamaguruge_private_project-monitoring_service-1 Creating
Container tharindurathgamaguruge_private_project-service1-1 Created
Container tharindurathgamaguruge_private_project-api_gateway_service-1 Created
Container tharindurathgamaguruge_private_project-monitoring_service-1 Created
Container tharindurathgamaguruge_private_project-service2-1 Created
Container tharindurathgamaguruge_private_project-rabbitmq-1 Starting
Container tharindurathgamaguruge_private_project-rabbitmq-1 Started
Container tharindurathgamaguruge_private_project-service1-1 Starting
Container tharindurathgamaguruge_private_project-monitoring_service-1 Starting
Container tharindurathgamaguruge_private_project-service2-1 Starting
Container tharindurathgamaguruge_private_project-api_gateway_service-1 Starting
Container tharindurathgamaguruge_private_project-service1-1 Started
Container tharindurathgamaguruge_private_project-monitoring_service-1 Started
Container tharindurathgamaguruge_private_project-service2-1 Started
Container tharindurathgamaguruge_private_project-api_gateway_service-1 Started
176 Cleaning up project directory and file based variables
177 Job succeeded

08:08

Duration: 29 seconds

Finished: 51 minutes ago

Queued: 1 second

Timeout: 1h (from project)

Runner: #106 (mF2szrZKb)

Tags: dchtra-aws

Commit b39a3c6f

Update .gitlab-ci.yml file

Pipeline #1359 Passed for project

deploy

Related jobs

→ deploy

→ deploy-aws

Figure: Deployment to AWS instance stage

4. Reflections

Main learnings and worst difficulties

1. Shutdown the services

Firstly, I implemented shutdown method by writing process kill functions in each service and sending an API to gateway service. However, the test was failed due to service1(implemented using python) killed before sending a response back to the API gateway service. Furthermore, could not implement any method to stop the RabbitMQ service. Then I used subprocess library in python. From this we can stop the running docker containers in the host (We need to mount host /var/run/docker.sock to communicate with the host docker system). Also, I had to add volume mount into the config.toml

```
volumes = ["/var/run/docker.sock:/var/run/docker.sock", "/cache"]
```

2. AWS deployment implementation

AWS deployment implementation can be done through different approaches.

2.1 Directly ssh into the ec2 instance and clone the repository then run the services with docker compose. Here we need to build the services again in the cloud is a drawback of this implementation.

2.2 Pushing images into the ECR and running the services on EC2 instance using docker-compose. For this implementation, I used bash script to push images to the ECR. This implementation can be more automated with Ansible.

3. Test cases execute methods

Firstly, I executed test cases inside the api_gateway_service docker and after understanding the behavior in git CI executed test cases using python image in the test stage.

Amount effort (hours) used

Around 56 hours.

Reference Links

[1]- <https://docs.sonarsource.com/sonarqube/9.8/user-guide/user-account/generating-and-using-tokens/>