

COMP.SE.140

Instructions for the teaching assistant

Implemented optional features

- Static analysis check is done using go vet (but it does look horrible since ./... didn't work in container for some reason)

Instructions for examiner to test the system.

0. Make sure to clean any other compose140 exercises from containers and images first.
1. Clone the repo: `git clone -b project https://github.com/KaluTheKova/COMP.SE.140.git`
2. Start the system with `docker compose up -d` (or `docker-compose up` if using Docker Compose standalone)
3. Run commands in any order you wish:
 - `curl localhost:8083/state -X PUT -d "INIT"`
 - `curl localhost:8083/state -X PUT -d "PAUSED"`
 - `curl localhost:8083/state -X PUT -d "RUNNING"`
 - `curl localhost:8083/state -X PUT -d "SHUTDOWN"`
 - `curl localhost:8083/state`
 - `curl localhost:8083/run-log`
 - `curl localhost:8083/messages`

NOTE: As per exercise instructions, "INIT" is logged into state log as "RUNNING".

NOTE: Unit tests are not located in "tests" - folder, because Go requires tests to reside in the same folder as the file that is being tested.

Description of the CI/CD pipeline

Version management; use of branches etc

- Project pushes to two remotes simulatenously, Github and local Gitlab with CI/CD pipeline.
- Gitlab repo is locally run in the Docker, using Gitlab CE.
- Github repo: <https://github.com/KaluTheKova/COMP.SE.140/tree/project>

Building tools

- Built using go build
- Image: golang:alpine

Testing; tools and test cases

- Testing uses go's testing: go test

Packing

- Packaged as containers and run in a dind-container on the gitlab-runner.

Deployment

- Deployed using dind-container on the gitlab-runner.

Operating; monitoring

Operated using the following curl commands:

- curl localhost:8083/state -X PUT -d "INIT"
- curl localhost:8083/state -X PUT -d "PAUSED"
- curl localhost:8083/state -X PUT -d "RUNNING"
- curl localhost:8083/state -X PUT -d "SHUTDOWN"
- curl localhost:8083/state
- curl localhost:8083/run-log
- curl localhost:8083/messages

No monitoring implemented.

Example runs of the pipeline

Succeeding pipeline:

```
Running with gitlab-runner 15.5.1 (7178588d)
2 on privileged-runner k4L6VW4c
3Preparing the "docker" executor00:23
4Using Docker executor with image docker ...
5Starting service docker:dind ...
6Pulling docker image docker:dind ...
7Using docker image
sha256:0e865cd51cb00cea31ca715846714e36966d0a404bbc50490633e3c1416caa6d for
docker:dind with digest
docker@sha256:2e0135466bcb3398e7f3943b87aef5c036dbaf1683805b8bfe992a477f7269e9 ...
8Waiting for services to be up and running (timeout 30 seconds)...
9Pulling docker image docker ...
10Using docker image
sha256:0f8d12a73562adf6588be88e37974abd42168017f375a1e160ba08a7ee3ffaa9 for docker
with digest
docker@sha256:75026b00c823579421c1850c00def301a6126b3f3f684594e51114c997f76467 ...
12Preparing environment00:01
13Running on runner-k4l6vw4c-project-36-concurrent-0 via b597f1396b2f...
15Getting source from Git repository00:01
16Fetching changes with git depth set to 20...
17Reinitialized existing Git repository in /builds/gitlab-instance-
9d36923c/COMP.SE.140/.git/
18Checking out 46f993a0 as project...
19Skipping Git submodules setup
21Executing "step_script" stage of the job script02:01
22Using docker image
sha256:0f8d12a73562adf6588be88e37974abd42168017f375a1e160ba08a7ee3ffaa9 for docker
with digest
docker@sha256:75026b00c823579421c1850c00def301a6126b3f3f684594e51114c997f76467 ...
23$ docker-compose -H $DOCKER_HOST up --build -d
24#1 [compse140-obse internal] load build definition from Dockerfile
25#1 transferring dockerfile: 205B done
26#1 DONE 0.1s
27#2 [compse140-httpserve internal] load build definition from Dockerfile
28#2 ...
29#3 [compse140-rabbitmq internal] load build definition from Dockerfile
30#3 transferring dockerfile: 162B done
31#3 DONE 0.1s
32#2 [compse140-httpserve internal] load build definition from Dockerfile
33#2 transferring dockerfile: 213B done
34#2 DONE 0.2s
```

35#4 [compse140-gateway internal] load build definition from Dockerfile
36#4 transferring dockerfile: 211B done
37#4 DONE 0.2s
38#5 [compse140-gateway internal] load .dockerignore
39#5 transferring context: 2B done
40#5 DONE 0.2s
41#6 [compse140-imed internal] load build definition from Dockerfile
42#6 transferring dockerfile: 195B done
43#6 DONE 0.2s
44#7 [compse140-orig internal] load build definition from Dockerfile
45#7 transferring dockerfile: 195B done
46#7 DONE 0.3s
47#8 [compse140-rabbitmq internal] load .dockerignore
48#8 transferring context: 2B done
49#8 DONE 0.2s
50#9 [compse140-orig internal] load .dockerignore
51#9 transferring context: 2B done
52#9 DONE 0.2s
53#10 [compse140-imed internal] load .dockerignore
54#10 transferring context: 2B done
55#10 DONE 0.2s
56#11 [compse140-obse internal] load .dockerignore
57#11 transferring context: 2B done
58#11 DONE 0.2s
59#12 [compse140-gateway internal] load metadata for
docker.io/library/golang:alpine
60#12 ...
61#13 [compse140-httpserve internal] load .dockerignore
62#13 transferring context: 2B done
63#13 DONE 0.2s
64#12 [compse140-imed internal] load metadata for docker.io/library/golang:alpine
65#12 ...
66#14 [compse140-rabbitmq internal] load metadata for
docker.io/library/rabbitmq:3-management
67#14 DONE 2.5s
68#12 [compse140-obse internal] load metadata for docker.io/library/golang:alpine
69#12 DONE 2.6s
70#15 [compse140-orig internal] load build context
71#15 transferring context: 15.46kB done
72#15 DONE 0.0s
73#16 [compse140-obse 1/6] FROM
docker.io/library/golang:alpine@sha256:2381c1e5f8350a901597d633b2e517775eeac7a6682
be39225a93b22cfd0f8bb
74#16 resolve
docker.io/library/golang:alpine@sha256:2381c1e5f8350a901597d633b2e517775eeac7a6682
be39225a93b22cfd0f8bb 0.0s done
75#16 ...
76#17 [compse140-httpserve internal] load build context
77#17 transferring context: 2.36kB done
78#17 DONE 0.1s
79#18 [compse140-gateway internal] load build context
80#18 transferring context: 30.24kB done
81#18 DONE 0.1s
82#19 [compse140-imed internal] load build context
83#19 transferring context: 8.02kB done
84#19 DONE 0.2s
85#20 [compse140-obse internal] load build context
86#20 transferring context: 8.26kB done
87#20 DONE 0.2s

88#16 [compse140-obse 1/6] FROM
docker.io/library/golang:alpine@sha256:2381c1e5f8350a901597d633b2e517775eeac7a6682
be39225a93b22cfd0f8bb
89#16 sha256:2381c1e5f8350a901597d633b2e517775eeac7a6682be39225a93b22cfd0f8bb
1.65kB / 1.65kB done
90#16 sha256:3183b343d01a6e8c9bee7b7c4eb7208518e68dc0cdac8623e1575820342472ed
1.16kB / 1.16kB done
91#16 sha256:feb4bbda921c3f51bc84ae6f12414c53189d33e193e37d930b4a44076b7fb348
5.11kB / 5.11kB done
92#16 ...
93#21 [compse140-rabbitmq 1/2] FROM docker.io/library/rabbitmq:3-
management@sha256:65a1412faeae95d260c8fcab7e0aa138fd4842726713b8496a2be87b26facca3
94#21 resolve docker.io/library/rabbitmq:3-
management@sha256:65a1412faeae95d260c8fcab7e0aa138fd4842726713b8496a2be87b26facca3
0.0s done
95#21 sha256:65a1412faeae95d260c8fcab7e0aa138fd4842726713b8496a2be87b26facca3
1.42kB / 1.42kB done
96#21 sha256:e7b677df5e85506845468791bf8ff9267523ceda705a32008b4cbddba460db09
2.62kB / 2.62kB done
97#21 sha256:854c78c564838cd1e532115a0eda5dbdef260c164ac1849da3e3815ee0db67c9
8.52kB / 8.52kB done
98#21 sha256:03dea6b5e70e630bb5bf621528463788e46cf2571f98f3f591e10633f1b650d9
334.49kB / 334.49kB 1.3s done
99#21 sha256:cae1c228689d2f294439828746d67d06536e5cf331393d0529e18ce4c8f222f8
8.92kB / 8.92kB 1.3s done
100#21 sha256:846c0b181ffff0c667d9444f8378e8fcfa13116da8d308bf21673f7e4bea8d580
7.34MB / 28.58MB 5.4s
101#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
6.29MB / 51.10MB 5.4s
102#21 sha256:92e035229b114a43bd976252b210632f191d41ae606b4eafaf029f0d06c0f454
5.19kB / 5.19kB 1.7s done
103#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
2.10MB / 21.54MB 5.4s
104#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
4.19MB / 21.54MB 6.6s
105#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
9.44MB / 51.10MB 7.4s
106#21 sha256:846c0b181ffff0c667d9444f8378e8fcfa13116da8d308bf21673f7e4bea8d580
9.44MB / 28.58MB 8.1s
107#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
6.29MB / 21.54MB 8.5s
108#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
12.58MB / 51.10MB 9.4s
109#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
8.39MB / 21.54MB 10.0s
110#21 sha256:846c0b181ffff0c667d9444f8378e8fcfa13116da8d308bf21673f7e4bea8d580
11.53MB / 28.58MB 11.3s
111#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
15.73MB / 51.10MB 11.4s
112#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
10.49MB / 21.54MB 11.8s
113#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
18.87MB / 51.10MB 12.7s
114#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
22.02MB / 51.10MB 14.0s
115#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
12.58MB / 21.54MB 14.9s
116#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
25.17MB / 51.10MB 15.4s

117#21 sha256:846c0b181fff0c667d9444f8378e8fcfa13116da8d308bf21673f7e4bea8d580
13.63MB / 28.58MB 15.8s
118#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
28.31MB / 51.10MB 16.7s
119#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
14.68MB / 21.54MB 17.7s
120#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
31.46MB / 51.10MB 18.1s
121#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
34.60MB / 51.10MB 19.6s
122#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
16.78MB / 21.54MB 20.3s
123#21 sha256:846c0b181fff0c667d9444f8378e8fcfa13116da8d308bf21673f7e4bea8d580
14.68MB / 28.58MB 20.9s
124#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
37.75MB / 51.10MB 20.9s
125#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
40.89MB / 51.10MB 22.2s
126#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
44.04MB / 51.10MB 23.4s
127#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
18.87MB / 21.54MB 23.7s
128#21 sha256:846c0b181fff0c667d9444f8378e8fcfa13116da8d308bf21673f7e4bea8d580
16.78MB / 28.58MB 24.7s
129#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
47.19MB / 51.10MB 24.7s
130#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
50.33MB / 51.10MB 26.0s
131#21 sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38
51.10MB / 51.10MB 26.1s done
132#21 sha256:f0926bc87948c9673fbbbee4d4bffa0af175dfae14be1b5b8e6a3d0c0037099ef 0B
/ 274B 26.2s
133#21 sha256:f0926bc87948c9673fbbbee4d4bffa0af175dfae14be1b5b8e6a3d0c0037099ef
274B / 274B 26.6s
134#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
20.97MB / 21.54MB 26.7s
135#21 sha256:f0926bc87948c9673fbbbee4d4bffa0af175dfae14be1b5b8e6a3d0c0037099ef
274B / 274B 26.6s done
136#21 sha256:9ae105c01998d72f7a495f74a8674275001eb3ee74e232d458dce9091b2706c3 0B
/ 107B 26.7s
137#21 sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad
21.54MB / 21.54MB 26.9s done
138#21 sha256:9ae105c01998d72f7a495f74a8674275001eb3ee74e232d458dce9091b2706c3
107B / 107B 26.9s done
139#21 sha256:553e1f2f08f51bf343735e0e580955a4c851bd0ba8b2a2f0ed461538e37a15bb 0B
/ 501B 27.0s
140#21 sha256:46f5869ee28d3f9a1362b44150f944d7f8ea10bb2be4a01f1c9ee635619af4be 0B
/ 834B 27.0s
141#21 sha256:553e1f2f08f51bf343735e0e580955a4c851bd0ba8b2a2f0ed461538e37a15bb
501B / 501B 27.2s done
142#21 sha256:2ed5fb86c5042c27df38884a97dc40996ae0891099ef6365b2ac9b1d4ca8b081 0B
/ 9.90MB 27.3s
143#21 sha256:46f5869ee28d3f9a1362b44150f944d7f8ea10bb2be4a01f1c9ee635619af4be
834B / 834B 27.3s done
144#21 ...
145#16 [compse140-obse 1/6] FROM
docker.io/library/golang:alpine@sha256:2381c1e5f8350a901597d633b2e517775eeac7a6682
be39225a93b22cfd0f8bb

146#16 sha256:8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9 0B / 3.37MB 27.3s
147#16 sha256:a2f8637abd914a8a62416e027a351293d0472bc4b4f44383c6f425fd0e03861c 0B / 284.81kB 30.8s
148#16 sha256:8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9 1.05MB / 3.37MB 31.1s
149#16 sha256:a2f8637abd914a8a62416e027a351293d0472bc4b4f44383c6f425fd0e03861c 284.81kB / 284.81kB 31.3s done
150#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613 0B / 122.33MB 31.5s
151#16 sha256:8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9 2.10MB / 3.37MB 33.4s
152#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613 6.29MB / 122.33MB 34.8s
153#16 sha256:8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9 3.15MB / 3.37MB 36.2s
154#16 sha256:8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9 3.37MB / 3.37MB 36.6s done
155#16 extracting
sha256:8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9
156#16 sha256:4f26d270037d8578282b40f86c0a1816fc5d034d6213f7dcab8440056a589309 0B / 155B 36.7s
157#16 sha256:8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9 3.37MB / 3.37MB 36.6s done
158#16 extracting
sha256:8921db27df2831fa6eaa85321205a2470c669b855f3ec95d5a3c2b46de0442c9 0.1s done
159#16 extracting
sha256:a2f8637abd914a8a62416e027a351293d0472bc4b4f44383c6f425fd0e03861c 0.0s done
160#16 ...
161#21 [compse140-rabbitmq 1/2] FROM docker.io/library/rabbitmq:3-management@sha256:65a1412faeae95d260c8fcab7e0aa138fd4842726713b8496a2be87b26facca3
162#21 sha256:846c0b181fff0c667d9444f8378e8fcfa13116da8d308bf21673f7e4bea8d580 25.17MB / 28.58MB 37.5s
163#21 sha256:2ed5fb86c5042c27df38884a97dc40996ae0891099ef6365b2ac9b1d4ca8b081 9.90MB / 9.90MB 30.9s done
164#21 ...
165#16 [compse140-obse 1/6] FROM docker.io/library/golang:alpine@sha256:2381c1e5f8350a901597d633b2e517775eeac7a6682be39225a93b22cfd0f8bb
166#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613 12.58MB / 122.33MB 38.0s
167#16 sha256:4f26d270037d8578282b40f86c0a1816fc5d034d6213f7dcab8440056a589309 155B / 155B 38.7s done
168#16 sha256:4f26d270037d8578282b40f86c0a1816fc5d034d6213f7dcab8440056a589309 155B / 155B 38.7s done
169#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613 18.87MB / 122.33MB 40.2s
170#16 ...
171#21 [compse140-rabbitmq 1/2] FROM docker.io/library/rabbitmq:3-management@sha256:65a1412faeae95d260c8fcab7e0aa138fd4842726713b8496a2be87b26facca3
172#21 sha256:846c0b181fff0c667d9444f8378e8fcfa13116da8d308bf21673f7e4bea8d580 28.58MB / 28.58MB 39.1s done
173#21 extracting
sha256:846c0b181fff0c667d9444f8378e8fcfa13116da8d308bf21673f7e4bea8d580 0.4s done
174#21 extracting
sha256:03dea6b5e70e630bb5bf621528463788e46cf2571f98f3f591e10633f1b650d9 0.0s done
175#21 extracting
sha256:cae1c228689d2f294439828746d67d06536e5cf331393d0529e18ce4c8f222f8 done

176#21 extracting
sha256:3ddf939b446499c0937604d712f7e13340db9687e710d61c8b0b278700ba9e38 0.5s done
177#21 extracting
sha256:92e035229b114a43bd976252b210632f191d41ae606b4eafaf029f0d06c0f454 done
178#21 extracting
sha256:d5e103ef1475feb2f050483aa4216a95752e3c27c02404fcc0a815938da1c0ad 0.3s done
179#21 extracting
sha256:f0926bc87948c9673fbbee4d4bfff60af175dfae14be1b5b8e6a3d0c0037099ef done
180#21 extracting
sha256:9ae105c01998d72f7a495f74a8674275001eb3ee74e232d458dce9091b2706c3 done
181#21 extracting
sha256:553e1f2f08f51bf343735e0e580955a4c851bd0ba8b2a2f0ed461538e37a15bb done
182#21 extracting
sha256:46f5869ee28d3f9a1362b44150f944d7f8ea10bb2be4a01f1c9ee635619af4be done
183#21 extracting
sha256:2ed5fb86c5042c27df38884a97dc40996ae0891099ef6365b2ac9b1d4ca8b081 0.2s done
184#21 DONE 41.3s
185#16 [compse140-obse 1/6] FROM
docker.io/library/golang:alpine@sha256:2381c1e5f8350a901597d633b2e517775eeac7a6682
be39225a93b22cfd0f8bb
186#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
25.17MB / 122.33MB 42.0s
187#16 ...
188#22 [compse140-rabbitmq 2/2] RUN rabbitmq-plugins enable --offline
rabbitmq_mqtt rabbitmq_federation_management rabbitmq_stomp
189#0 0.640 Enabling plugins on node rabbit@buildkitsandbox:
190#0 0.640 rabbitmq_mqtt
191#0 0.640 rabbitmq_federation_management
192#0 0.640 rabbitmq_stomp
193#0 0.959 The following plugins have been configured:
194#0 0.959 rabbitmq_federation
195#0 0.959 rabbitmq_federation_management
196#0 0.959 rabbitmq_management
197#0 0.959 rabbitmq_management_agent
198#0 0.959 rabbitmq_mqtt
199#0 0.959 rabbitmq_prometheus
200#0 0.959 rabbitmq_stomp
201#0 0.959 rabbitmq_web_dispatch
202#0 0.959 Applying plugin configuration to rabbit@buildkitsandbox...
203#0 0.960 The following plugins have been enabled:
204#0 0.960 rabbitmq_federation
205#0 0.960 rabbitmq_federation_management
206#0 0.960 rabbitmq_mqtt
207#0 0.960 rabbitmq_stomp
208#0 0.960
209#0 0.960 set 8 plugins.
210#0 0.960 Offline change; changes will take effect at broker restart.
211#22 DONE 1.2s
212#16 [compse140-obse 1/6] FROM
docker.io/library/golang:alpine@sha256:2381c1e5f8350a901597d633b2e517775eeac7a6682
be39225a93b22cfd0f8bb
213#16 ...
214#23 [compse140-rabbitmq] exporting to image
215#23 exporting layers 0.0s done
216#23 writing image
sha256:afd3fa418a1741146c51637819fcfb75ef62a53952c605f88f3d59d243f7ab68 done
217#23 naming to docker.io/library/compse140-rabbitmq done
218#23 DONE 0.1s

219#16 [compse140-obse 1/6] FROM
docker.io/library/golang:alpine@sha256:2381c1e5f8350a901597d633b2e517775eeac7a6682
be39225a93b22cfd0f8bb
220#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
31.46MB / 122.33MB 43.8s
221#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
37.75MB / 122.33MB 45.6s
222#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
44.04MB / 122.33MB 47.4s
223#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
50.33MB / 122.33MB 49.3s
224#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
56.62MB / 122.33MB 51.0s
225#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
62.91MB / 122.33MB 52.8s
226#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
69.21MB / 122.33MB 54.7s
227#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
75.50MB / 122.33MB 56.5s
228#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
81.79MB / 122.33MB 58.5s
229#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
88.08MB / 122.33MB 60.2s
230#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
94.37MB / 122.33MB 62.0s
231#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
100.66MB / 122.33MB 63.9s
232#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
106.95MB / 122.33MB 65.6s
233#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
113.25MB / 122.33MB 67.5s
234#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
119.54MB / 122.33MB 69.2s
235#16 sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
122.33MB / 122.33MB 70.3s done
236#16 extracting
sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613
237#16 extracting
sha256:d48e7ca896ec974241d86439d34188186e085705e42d914d8b7757c27eea8613 1.7s done
238#16 extracting
sha256:4f26d270037d8578282b40f86c0a1816fc5d034d6213f7dcab8440056a589309 done
239#16 DONE 72.6s
240#24 [compse140-httpserve 2/6] WORKDIR /app
241#24 DONE 0.4s
242#25 [compse140-obse 3/6] COPY go.mod ./
243#25 DONE 0.1s
244#26 [compse140-imed 3/6] COPY go.mod ./
245#26 ...
246#27 [compse140-gateway 3/6] COPY go.mod ./
247#27 DONE 0.1s
248#28 [compse140-httpserve 3/6] COPY go.mod ./
249#28 DONE 0.2s
250#26 [compse140-imed 3/6] COPY go.mod ./
251#26 DONE 0.2s
252#29 [compse140-orig 3/6] COPY go.mod ./
253#29 DONE 0.2s
254#30 [compse140-orig 4/6] RUN go mod download
255#30 ...
256#31 [compse140-httpserve 4/6] RUN go mod download


```
257#0 1.069 go: no module dependencies to download
258#31 DONE 1.1s
259#32 [compse140-httpserve 5/6] COPY . /app
260#32 DONE 0.0s
261#33 [compse140-httpserve 6/6] RUN go build -o /app/httpserve
262#33 ...
263#34 [compse140-obse 4/6] RUN go mod download
264#34 DONE 1.9s
265#35 [compse140-obse 5/6] COPY . /app
266#35 DONE 0.1s
267#36 [compse140-obse 6/6] RUN go build -o /app/obse
268#36 ...
269#33 [compse140-httpserve 6/6] RUN go build -o /app/httpserve
270#33 DONE 1.3s
271#23 [compse140-httpserve] exporting to image
272#23 exporting layers 0.1s done
273#23 writing image
sha256:1f1162696d7e323bfbf44c1538d1ade9d45c4422691d4692a861fdb04c167978 done
274#23 naming to docker.io/library/compse140-httpserve done
275#23 DONE 0.2s
276#37 [compse140-imed 4/6] RUN go mod download
277#37 DONE 2.9s
278#38 [compse140-imed 5/6] COPY . ./
279#38 DONE 0.0s
280#39 [compse140-imed 6/6] RUN go build -o /imed
281#39 ...
282#36 [compse140-obse 6/6] RUN go build -o /app/obse
283#36 DONE 1.3s
284#23 [compse140-obse] exporting to image
285#23 exporting layers 0.1s done
286#23 writing image
sha256:a8abacc7c52c487bc6a8302071479f6bb0d3cf71ef00357a8a213a504d5b814d done
287#23 naming to docker.io/library/compse140-obse done
288#23 DONE 0.3s
289#40 [compse140-gateway 4/6] RUN go mod download
290#40 ...
291#39 [compse140-imed 6/6] RUN go build -o /imed
292#39 DONE 1.8s
293#23 [compse140-imed] exporting to image
294#23 exporting layers 0.1s done
295#23 writing image
sha256:3d73637e92e705f1289a04be3dab4e202ff800712a1a3366daa5f2a019ac5b2d done
296#23 naming to docker.io/library/compse140-imed done
297#23 DONE 0.4s
298#40 [compse140-gateway 4/6] RUN go mod download
299#40 ...
300#30 [compse140-orig 4/6] RUN go mod download
301#30 DONE 12.2s
302#41 [compse140-orig 5/6] COPY . ./
303#41 DONE 0.1s
304#42 [compse140-orig 6/6] RUN go build -o /orig
305#42 DONE 1.3s
306#40 [compse140-gateway 4/6] RUN go mod download
307#40 ...
308#23 [compse140-orig] exporting to image
309#23 exporting layers
310#23 exporting layers 0.7s done
311#23 writing image
sha256:98c10ffffbfcff4b586d4f4f0a385c1e9835633efc1ffeb34a3756d1b184a1fb9 done
```

```
312#23 naming to docker.io/library/compse140-orig done
313#23 DONE 1.1s
314#40 [compse140-gateway 4/6] RUN go mod download
315#40 DONE 22.4s
316#43 [compse140-gateway 5/6] COPY . /app
317#43 DONE 0.0s
318#44 [compse140-gateway 6/6] RUN go build -o /app/gateway
319#44 DONE 3.2s
320#23 [compse140-gateway] exporting to image
321#23 exporting layers
322#23 exporting layers 1.5s done
323#23 writing image
sha256:fbd342c467feb8244b2f5f69c2fdee0ec5888b63cc5ad44702812ad39105c620 done
324#23 naming to docker.io/library/compse140-gateway done
325#23 DONE 2.7s
326Network compse140_default Creating
327Network compse140_default Created
328Volume "compse140_message-storage" Creating
329Volume "compse140_message-storage" Created
330Container compse140-rabbitmq-1 Creating
331Container compse140-rabbitmq-1 Created
332Container compse140-imed-1 Creating
333Container compse140-obse-1 Creating
334Container compse140-httpserve-1 Creating
335Container compse140-gateway-1 Creating
336Container compse140-imed-1 Created
337Container compse140-httpserve-1 Created
338Container compse140-gateway-1 Created
339Container compse140-obse-1 Created
340Container compse140-orig-1 Creating
341Container compse140-orig-1 Created
342Container compse140-rabbitmq-1 Starting
343Container compse140-rabbitmq-1 Started
344Container compse140-gateway-1 Starting
345Container compse140-rabbitmq-1 Waiting
346Container compse140-rabbitmq-1 Waiting
347Container compse140-rabbitmq-1 Waiting
348Container compse140-gateway-1 Started
349Container compse140-rabbitmq-1 Healthy
350Container compse140-rabbitmq-1 Healthy
351Container compse140-rabbitmq-1 Healthy
352Container compse140-httpserve-1 Starting
353Container compse140-obse-1 Starting
354Container compse140-imed-1 Starting
355Container compse140-obse-1 Started
356Container compse140-httpserve-1 Started
357Container compse140-imed-1 Started
358Container compse140-orig-1 Starting
359Container compse140-orig-1 Started
360$ echo "Application successfully deployed."
361Application successfully deployed.
363Job succeeded
```

Failing pipeline:

Running with gitlab-runner 15.5.1 (7178588d)
_ on privileged-runner k4L6VW4c

```
3Preparing the "docker" executor00:46
4Using Docker executor with image golang:alpine ...
5Pulling docker image golang:alpine ...
6Using docker image
sha256:cae57157ceaa07c0ffffad419f2c4cedc683c31981f466a0fb9fd8cdd434e05d8 for
golang:alpine with digest
golang@sha256:a9b24b67dc83b3383d22a14941c2b2b2ca6a103d805cac6820fd1355943beaf1 ...
8Preparing environment00:02
9Running on runner-k416vw4c-project-36-concurrent-0 via b597f1396b2f...
11Getting source from Git repository00:02
12Fetching changes with git depth set to 20...
13Reinitialized existing Git repository in /builds/gitlab-instance-
9d36923c/COMP.SE.140/.git/
14Checking out caa18e8d as project...
15Skipping Git submodules setup
17Executing "step_script" stage of the job script00:47
18Using docker image
sha256:cae57157ceaa07c0ffffad419f2c4cedc683c31981f466a0fb9fd8cdd434e05d8 for
golang:alpine with digest
golang@sha256:a9b24b67dc83b3383d22a14941c2b2b2ca6a103d805cac6820fd1355943beaf1 ...
19$ echo "Waiting for 30 seconds..."
20Waiting for 30 seconds...
21$ sleep 30
22$ cd ./gateway
23$ CGO_ENABLED=0 go test
24go: downloading github.com/docker/docker v20.10.22+incompatible
25go: downloading github.com/gin-gonic/gin v1.8.1
26go: downloading github.com/stretchr/testify v1.8.1
27go: downloading github.com/davecgh/go-spew v1.1.1
28go: downloading github.com/pmezard/go-difflib v1.0.0
29go: downloading gopkg.in/yaml.v3 v3.0.1
30go: downloading github.com/gin-contrib/sse v0.1.0
31go: downloading golang.org/x/net v0.2.0
32go: downloading github.com/matttn/go-isatty v0.0.16
33go: downloading google.golang.org/protobuf v1.28.1
34go: downloading github.com/ugorji/go/codec v1.2.7
35go: downloading github.com/pelletier/go-toml/v2 v2.0.6
36go: downloading gopkg.in/yaml.v2 v2.4.0
37go: downloading github.com/go-playground/validator/v10 v10.11.1
38go: downloading golang.org/x/sys v0.2.0
39go: downloading github.com/leodido/go-urn v1.2.1
40go: downloading github.com/go-playground/universal-translator v0.18.0
41go: downloading golang.org/x/crypto v0.3.0
42go: downloading golang.org/x/text v0.4.0
43go: downloading github.com/go-playground/locales v0.14.0
44go: downloading github.com/docker/distribution v2.8.1+incompatible
45go: downloading github.com/opencontainers/go-digest v1.0.0
46go: downloading github.com/opencontainers/image-spec v1.0.2
47go: downloading github.com/docker/go-connections v0.4.0
48go: downloading github.com/pkg/errors v0.9.1
49go: downloading github.com/docker/go-units v0.5.0
50go: downloading github.com/sirupsen/logrus v1.9.0
51go: downloading github.com/gogo/protobuf v1.3.2
52# github.com/KaluTheKova/COMP.SE.140/httpserve
53./gateway.go:164:3: log.Panic call has possible formatting directive %s
54./gateway.go:173:3: log.Panic call has possible formatting directive %s
55FAIL github.com/KaluTheKova/COMP.SE.140/httpserve [build failed]
57ERROR: Job failed: exit code 2
```

Reflections

Main learnings and worst difficulties

Main learnings include a yearning for Jenkins automation server or any pipeline automation server where I could replay a pipeline and adjust the `gitlab-ci.yml` without having to always recommit.

I also learned that one can push into multiple repositories simultaneously. Previously, I had never needed this feature, but now it proved to be interesting to implement and use. So here I configured the project so that it pushes simultaneously to my Github and to my locally hosted Gitlab that ran the pipeline.

It was also quite interesting to setup Gitlab on my local docker. In real world jobs, DevOps engineer would be responsible for setting up a company's Gitlab and managing it, so this was quite an interesting thing to practise.

My worst difficulties were actually with RabbitMQ's Golang implementation. The library had some under-the-hood implementations that caused a lot of lost time and headache, such as somewhat unusual use of channels without documenting them.

Also, implementing local deployment was a somewhat pain. I would have liked to deploy to Heroku, but Heroku is no longer free starting November 28, 2022.

Instead of using HTTP request from Gateway to and respond on that, I wanted to try out using Docker CLI via Gateway's code. It was interesting to implement INIT, PAUSED, RUNNING and SHUTDOWN via that way. Using code to run container functions is also pretty DevOps thinking.

Of course, in a real world production application this might cause problems if the CLI library is reliant on Docker, since the production application would have to always run on Docker instead of being portable to another container engine. In a real world application, I probably would have used the HTTP method. In this case however, we can assume that Docker engine is used to run the application.

And naturally, I would have implemented error handling and basically everything more robustly.

Amount effort (hours) used
Around 50 I believe.