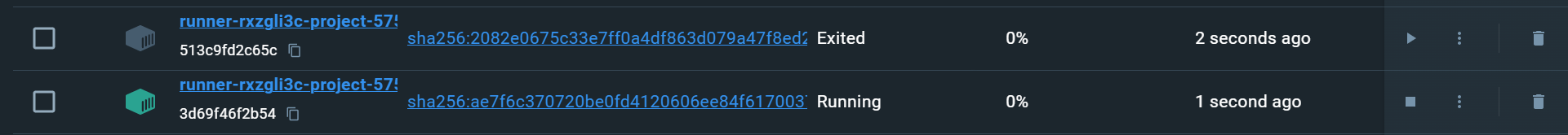
# CI/CD Setup in individual project advanced software

A screenshot of a computer

Description automatically generatedThe image below illustrates how the CI pipeline of HouseHunters is structured in Sem.6, along with explanations of each stage. Keep in minds that the ‘build for production’ and ‘deploy’ stages are not implemented yet.

Runner setup  
This semester a GitLab runner is used that runs on the developer’s machine. In order to make the pipeline more transferable the runner uses a Docker executor, which means that each pipeline job runs in an isolated docker container. This also gives developers the ability to specify what docker image they want to use for their job, which is a great abstraction from the PC that the runner is on.

Below is an image of what a CI/CD job looks in docker while it is being executed:

  
  
Pipeline stages  
  
Build

The build stage’s main purpose is to try to build each respective microservice’s code and check for any compilation errors. This would catch early problems with the code itself and prevent the pipeline from going any further.

Test  
The test stage runs controller, unit and integration tests on each microservice. The acceptance criteria is that for each microservice the code coverage is at least 80%. If the tests result in an insufficient coverage, the job fails.  
  
A black background with white text

Description automatically generated  
  
E2E

This stage uses docker compose and playwright to start the HouseHunters frontend and all microservices, along with a local MongoDB database and a RabbitMQ container.

After that, Playwright is used to execute end-to-end tests. An unsuccessful test fails the pipeline job and an artifact of the test is saved that can be inspected in trace.playwright.dev

A screen shot of a computer

Description automatically generated  
A screenshot of a computer

Description automatically generated

## Security

This stage uses the same docker compose setup from the E2E stage to execute a full scan from OWASP’s ZAP tool. Over 130 security tests are run and if a single on fails and is not configured to be ignored, the job fails.

A screen shot of a computer code

Description automatically generated