



UNIVERSITY OF AGDER

Faculty of Engineering and Science
Department of Information and Communication Technology

Assignment 6

CI/CD

Course: IKT206-G 26V
DevOps

Submitted by

Group 18

Christopher Sanden
Fabian Haukedal Johansen
Henrik Høie Corneliussen
Teodor Salvesen

April 8, 2026

Introduction

In this assignment, we worked with CI/CD by setting up a workflow that builds and pushes a Docker image to Docker Hub and then makes it possible to deploy updates more automatically. The main focus was on connecting the different parts from earlier assignments and making the setup function together in a practical way.

We first created a private repository in Docker Hub to store the image. After that, we set up a Gitea Actions workflow that logs in to Docker Hub, builds the image, and pushes it when changes are made. Finally, we made the necessary changes to the runner and the `docker-compose.yml` file so the correct image could be used and updated through Watchtower.

Contents

1	Docker Hub	1
2	Gitea Actions	1
3	Docker and Docker Compose	3
3.1	Runner changes	3
3.2	docker-compose.yml explanation	3
4	Conclusion	4

List of Figures

- 1 Showcasing where to create repository in Docker Hub 1

Listings

1 Docker Hub

For this section, we first registered accounts. The personal subscription tier is chosen by default, so no need to worry about that bit. Then we simply went to the repositories section and created a repository with the name ikt206g26v06 and set visibility to private.

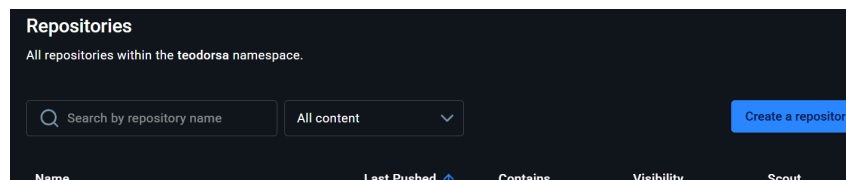


Figure 1: Showcasing where to create repository in Docker Hub

2 Gitea Actions

For the workflow, we based this on the one used in assignment 4. However, this one needed many changes to function properly as they are very different assignments.

```
1  # Name of workflow
2  name: assignment 6 workflow
3
4  # Defines when the workflow should run
5  on:
6    push:
7
8  # A job is a set of steps that run when the workflow is
   triggered
9  jobs:
10
11    # Creates a job named build-and-push
12    build-and-push:
13
14      # Specifies what runner environment to use
15      runs-on: ubuntu-latest
16
17      # Defines steps executed by the job
18      steps:
19
20        # Downloads the repository code
21        - name: Check out repository
22          uses: actions/checkout@v4
23
24        # Logs in to Docker Hub with username and token,
           which are saved secrets in gitea
```

```
25     - name: Login to docker hub
26       uses: docker/login-action@v3
27       with:
28         username: ${ secrets.DOCKER_HUB_USERNAME }
29         password: ${ secrets.DOCKER_HUB_TOKEN }
30
31     # Builds the Docker image from the Example folder,
32     # uses the Dockerfile in that folder, pushes the
33     # image to Docker Hub, and tags it as latest
34     - name: build and push image
35       uses: docker/build-push-action@v6
36       with:
37         context: ./Example
38         file: ./Example/Dockerfile
39         push: true
40         tags: docker.io/${ secrets.DOCKER_HUB_USERNAME
41           }/ikt206g26v06:latest
```

3 Docker and Docker Compose

3.1 Runner changes

Before talking about the changes to the docker compose file for this specific assignment, we first have to talk about the docker compose file for the runner. It is a simple fix, but since the runner created in an earlier assignment is repository specific, its token needs to be changed to ensure it creates the runner for the correct repository.

3.2 docker-compose.yml explanation

This file has remained mostly the same as in the previous assignment. The big addition to it is the watchtower and some small tweaks here and there. This code listing will only show what was changed and not the whole file, as that can be found in the previous assignment.

```
1     web:
2       image: docker.io/teodorsa/ikt206g26v06:latest
3       labels:
4         - "com.centurylinklabs.watchtower.enable=true"
5
6     watchtower:
7       image: containrrr/watchtower:latest
8       restart: always
9       environment:
10        DOCKER_API_VERSION: "1.40"
11       volumes:
12        - /var/run/docker.sock:/var/run/docker.sock
13        - /home/teodorsa/.docker/config.json:/config.json:ro
14       command: --interval 30 --label-enable
```

For the web section, the big change was changing it from using build to instead using an image. Specifically, the docker hub image. The other change is a label that defines this specific container as one that can be monitored and updated by the watchtower.

For the watchtower, it is set to use the latest image. Since only web has the label, that is the only container it is allowed to monitor or update. Monitors the docker hub image for updates. If it finds any, it will restart the web container with the new latest image now in use. DOCKER-API-VERSION is added just to ensure compatibility with the docker daemon. The first of the volumes gives access to docker itself, whereas the second one gives access to login credentials for docker hub. Finally, the "command" defines how often the watchtower should check the docker hub for updates to the image and which containers to give the updates to.

4 Conclusion

Overall, this assignment gave us a practical introduction to how CI/CD can be used together with Docker, Gitea Actions, and Docker Compose. By combining these tools, we created a setup where changes can be built automatically, pushed to Docker Hub, and then picked up by the running environment.

The assignment also showed that even though the overall structure can be based on earlier work, some changes are still needed to make everything function correctly for a new setup. In particular, the workflow configuration, runner token, and Watchtower additions were important for getting the full solution to work as intended.