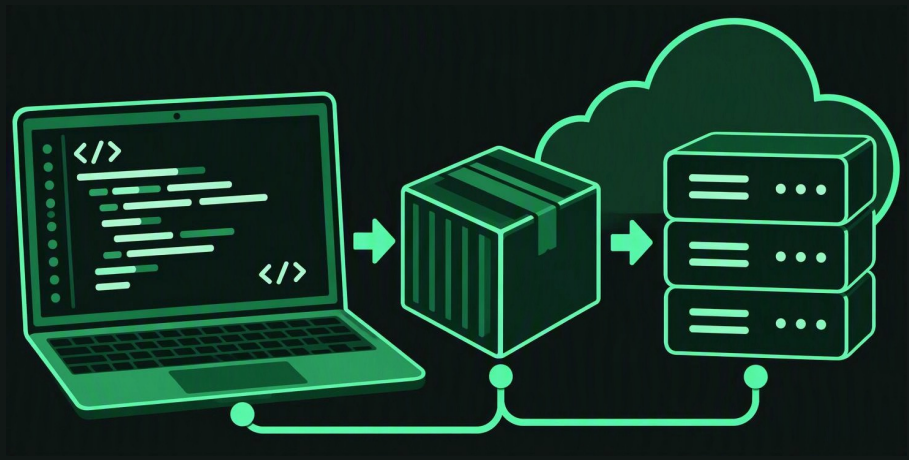
Deployment with CI/CD

3rd semester

A short introduction before the full-pipeline tutorial



CI: continuous integration

CI

Frequent code changes.
Automatic build and test checks.

Developers integrate small changes into a shared repository. Automation checks whether the project still builds and its tests pass.

In this lesson

GitHub Actions runs the build and tests.

CD: delivery or deployment?

Continuous delivery

A releasable version is ready.

A person decides when to release it to production.

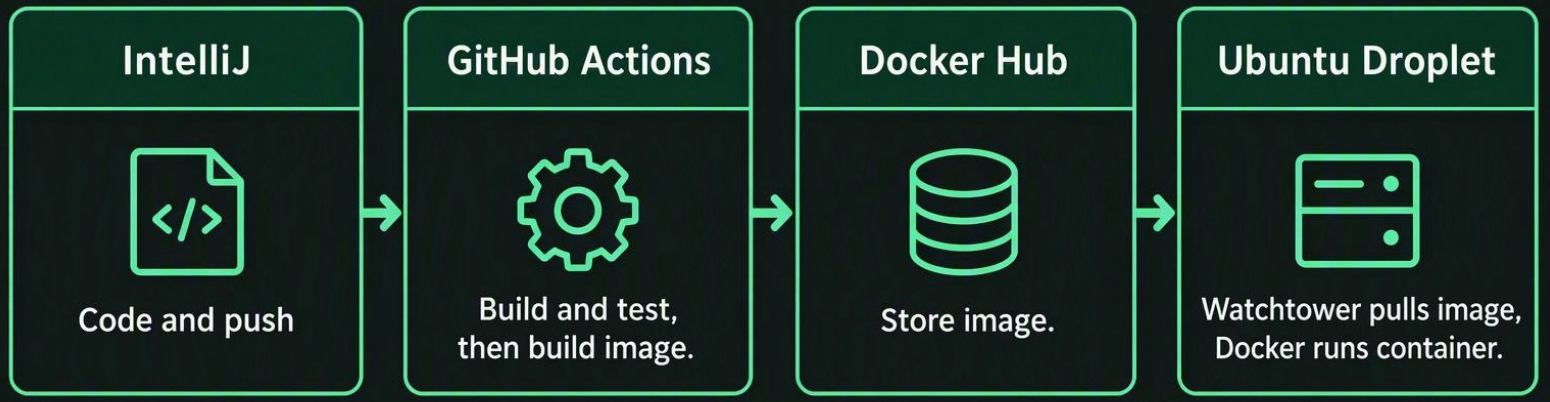
Continuous deployment

Passing changes go live automatically.

The automated pipeline also performs the production release.

The tutorial pipeline

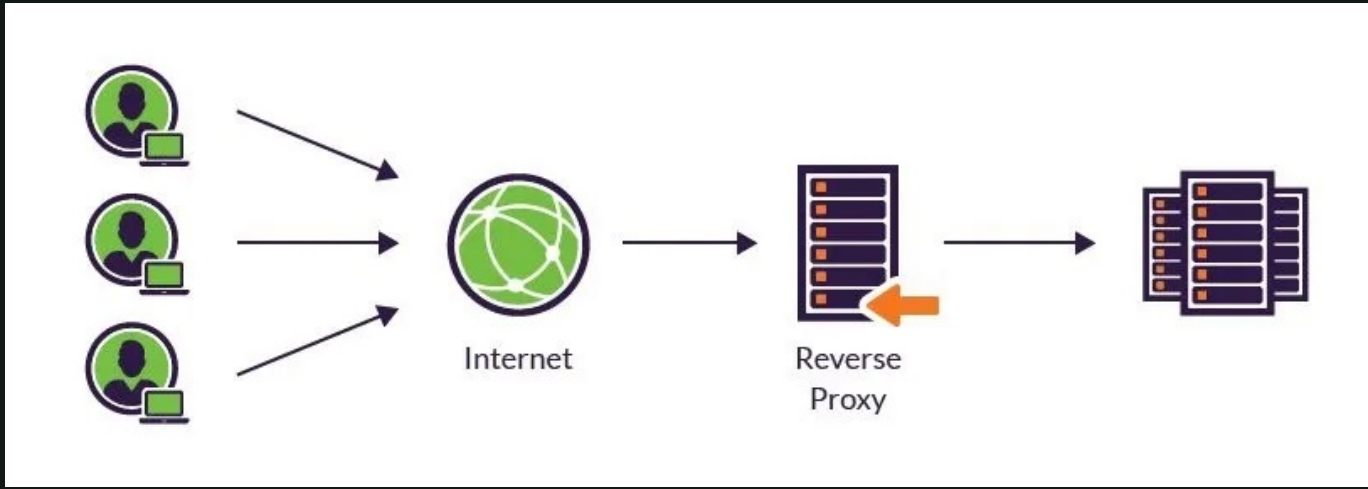
Code becomes an image, then a running application



Watchtower connects the published image to the running service.

Caddy as a reverse proxy

Caddy receives client requests, forwards them to your app,
and returns the app's response to the client.



Caddy also handles HTTPS certificates automatically for a configured domain.

Reverse proxy = Caddy. Backend servers = your application containers.

Source code, image and container

01 **Source code**

The Java project you edit in IntelliJ and version in GitHub.

02 **Docker image**

The packaged application that the pipeline publishes to Docker Hub.

03 **Container**

The running instance of that image on the Ubuntu Droplet.

Before the tutorial

Project and repository

Java / Javalin project
Maven configuration: pom.xml
Dockerfile
Workflow in .github/workflows/

Accounts and server

GitHub repository access
Docker Hub account
Ubuntu Droplet with Docker
Required credentials available

Credentials belong in the configured secrets, never in committed source code.

Checkpoints during the tutorial

1	**Repository**	Did the intended commit reach GitHub?
2	**GitHub Actions**	Did the build and test steps pass?
3	**Docker Hub**	Does the expected image exist?
4	**Ubuntu Droplet**	Does the updated application respond?

If something fails, inspect the first failed handoff before continuing.

The full-pipeline tutorial

One small change.
Visible in the deployed app.

Follow the tutorial, then make a small observable change and trace it through every stage.

3sem.kursusmaterialer.dk/toolbox/deployment-pipeline/full-pipeline/

Success: the app responds, and you can explain how the change arrived.