



AARHUS UNIVERSITET

# Microservices and DevOps

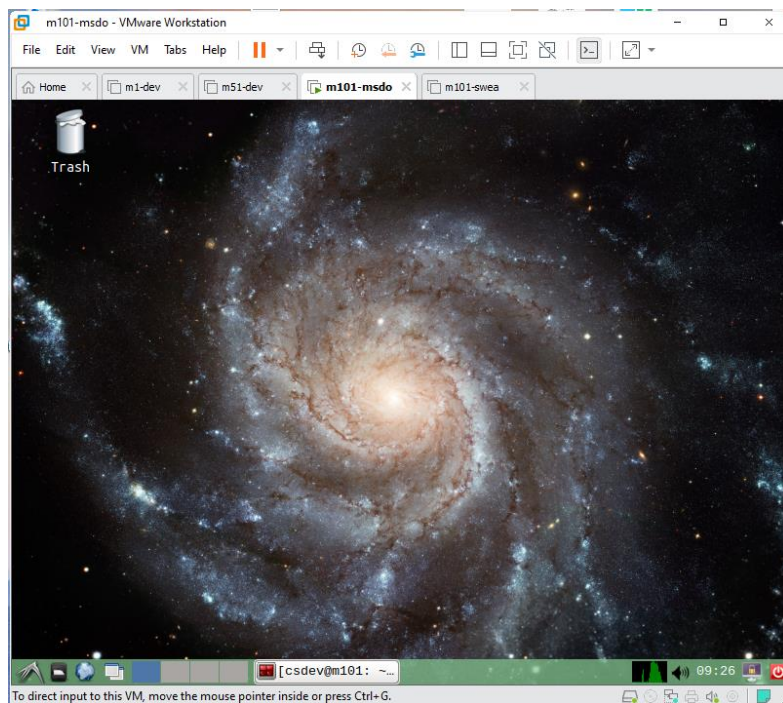
DevOps and Container Technology

Learning SkyCave

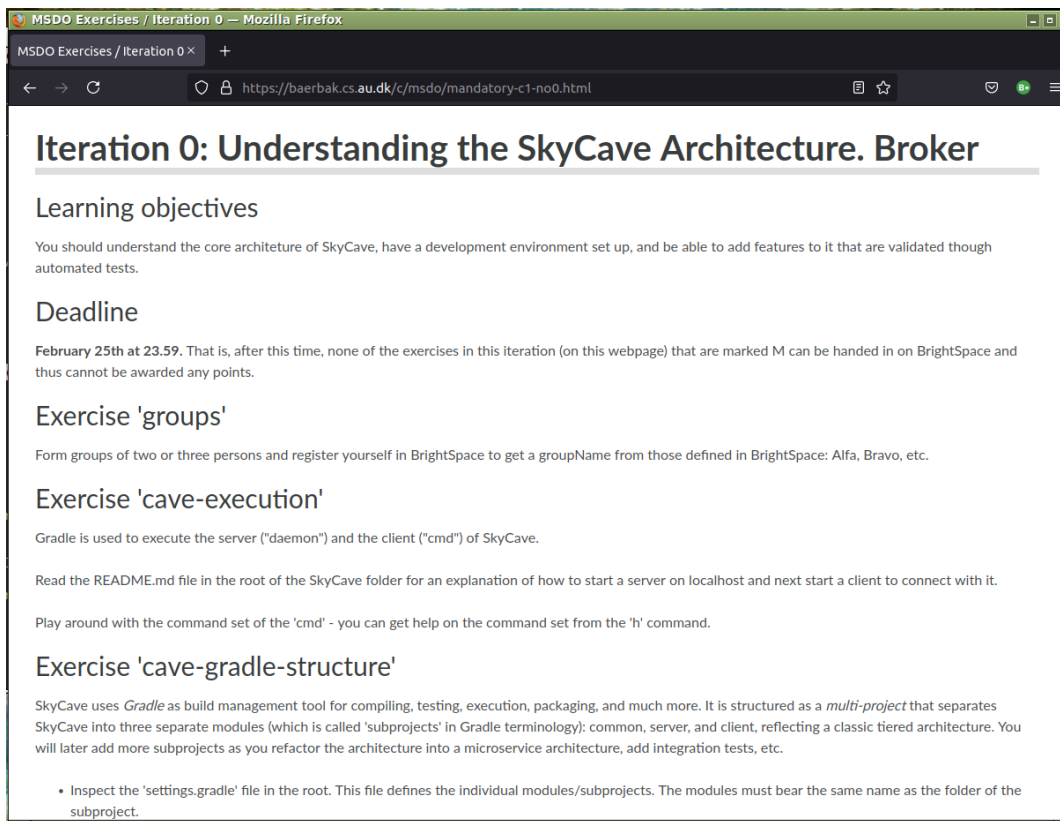
Henrik Bærbak Christensen

# Exercise 1

- Get the Mxx VM up and running
  - **User: csdev, pwd: csau**
- Alternative:
  - Install
    - Java SDK 11+
    - Gradle 6.8+
    - Docker
    - IntelliJ



- Start on the Mandatory Iteration 0 exercises...



**MSDO Exercises / Iteration 0 — Mozilla Firefox**

MSDO Exercises / Iteration 0 × +

← → ↻ 🔒 <https://baerbak.cs.au.dk/c/msdo/mandatory-c1-no0.html> 📄 ☆ 📧 + ☰

## Iteration 0: Understanding the SkyCave Architecture. Broker

### Learning objectives

You should understand the core architecture of SkyCave, have a development environment set up, and be able to add features to it that are validated through automated tests.

### Deadline

**February 25th at 23.59.** That is, after this time, none of the exercises in this iteration (on this webpage) that are marked M can be handed in on BrightSpace and thus cannot be awarded any points.

### Exercise 'groups'

Form groups of two or three persons and register yourself in BrightSpace to get a groupName from those defined in BrightSpace: Alfa, Bravo, etc.

### Exercise 'cave-execution'

Gradle is used to execute the server ("daemon") and the client ("cmd") of SkyCave.

Read the README.md file in the root of the SkyCave folder for an explanation of how to start a server on localhost and next start a client to connect with it.

Play around with the command set of the 'cmd' - you can get help on the command set from the 'h' command.

### Exercise 'cave-gradle-structure'

SkyCave uses *Gradle* as build management tool for compiling, testing, execution, packaging, and much more. It is structured as a *multi-project* that separates SkyCave into three separate modules (which is called 'subprojects' in Gradle terminology): common, server, and client, reflecting a classic tiered architecture. You will later add more subprojects as you refactor the architecture into a microservice architecture, add integration tests, etc.

- Inspect the 'settings.gradle' file in the root. This file defines the individual modules/subprojects. The modules must bear the same name as the folder of the subproject.