F21 SWTDISP
Oplæg til ”Delopgave Projekt”

Afleveringsopgave

Formål er at give en startplatform for eksamensprojektet.

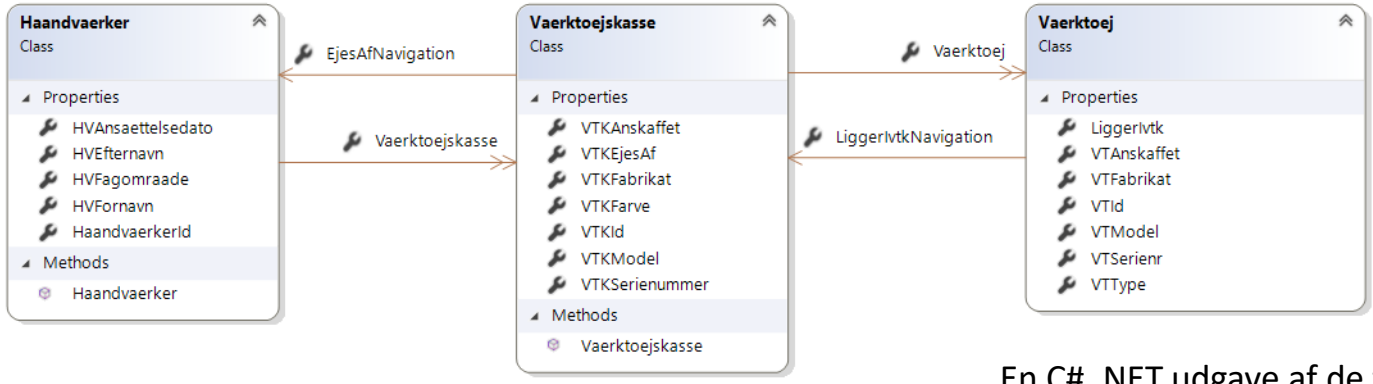
AU ECE SWTDISP

Jesper Rosholm Tørresø

Version: 22-02-2021 08:42

Hvad går denne opgave ud på?

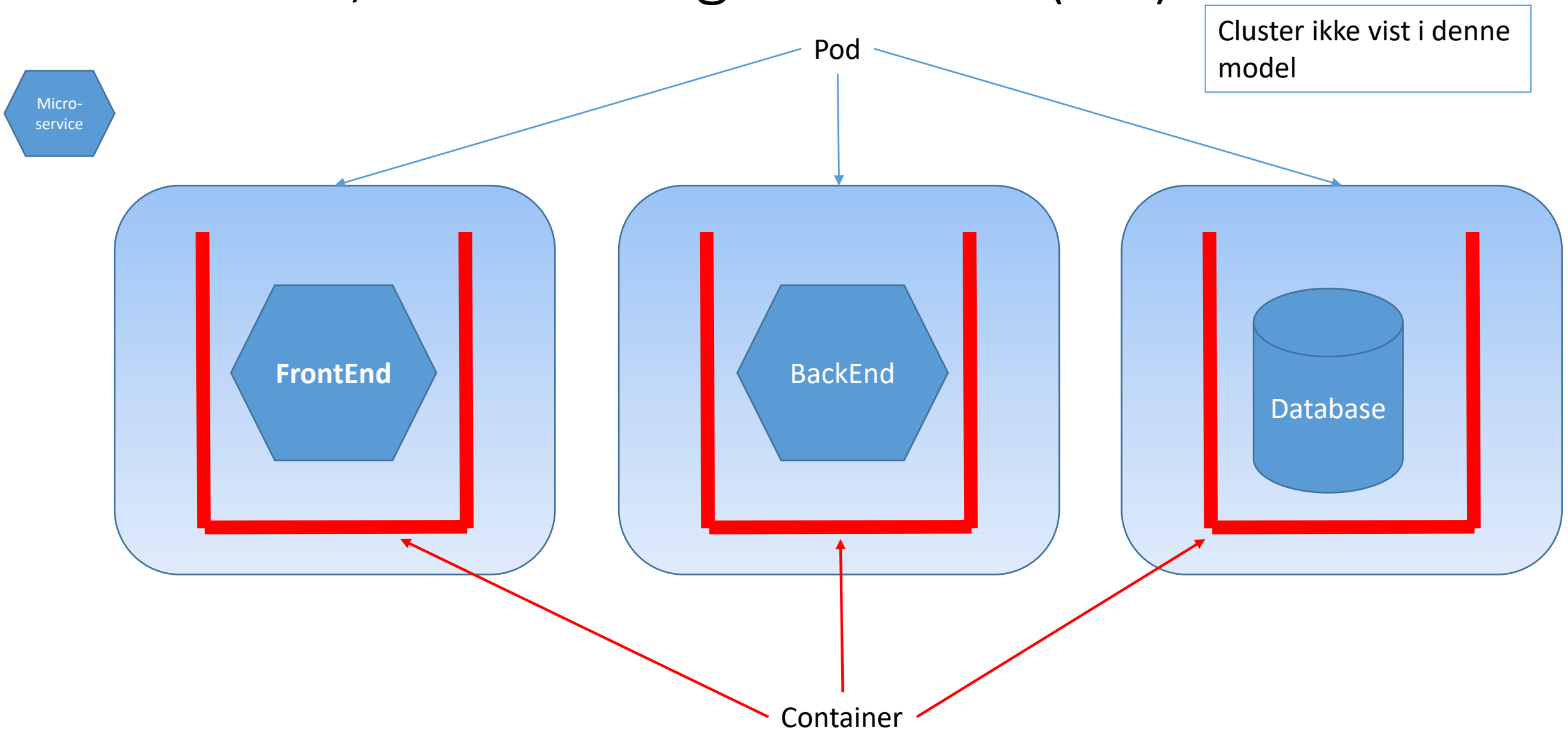
- At orkestrere tre services på et Kubernetes Cluster, til brug for en webapplikation, som kan manipulere (dvs. Database CRUD operationer) domæneobjekter fra denne domænemodel:



En C# .NET udgave af de tre klasser er tilgængelig på Blackboard

- De tre services der skal bruges er vist på næste side

1:Delprojektopgave. Samlet omfang: Frontend, Backend og Database (DB)



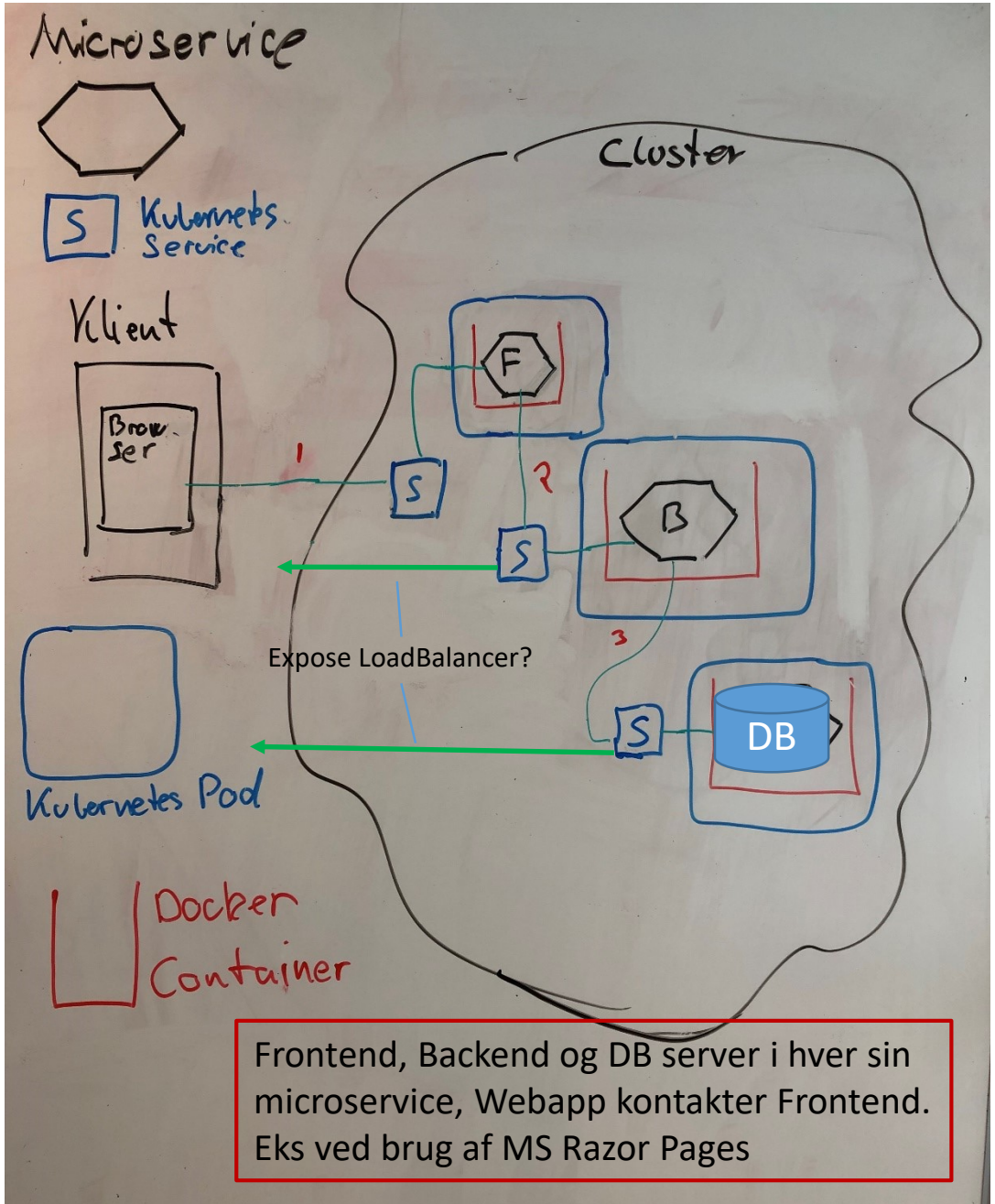
2: De tre services opgaver

- Frontend microservice: Skal kunne udføre en webapplikation for en browser klient
  - Kan bruge Microsoft Razor Pages som udgangspunkt! Men andre platforme kan bruges. Det er er dog ikke GUI delen der er så meget fokus på i opgaven, mere det, at der en webapplikation, som bruger services på et Kubernetes Cluster.
- Backend microservice: Skal kunne udføre forretningslogik i form opret, læse, opdater og slet (aka. CRUD) operationer mod en Database. Fuld REST semantik må gerne bruges altså POST GET PUT PATCH og DELETE.
- Database Server: Afvikler en standarddatabase (MS SQL Server/MySQL).

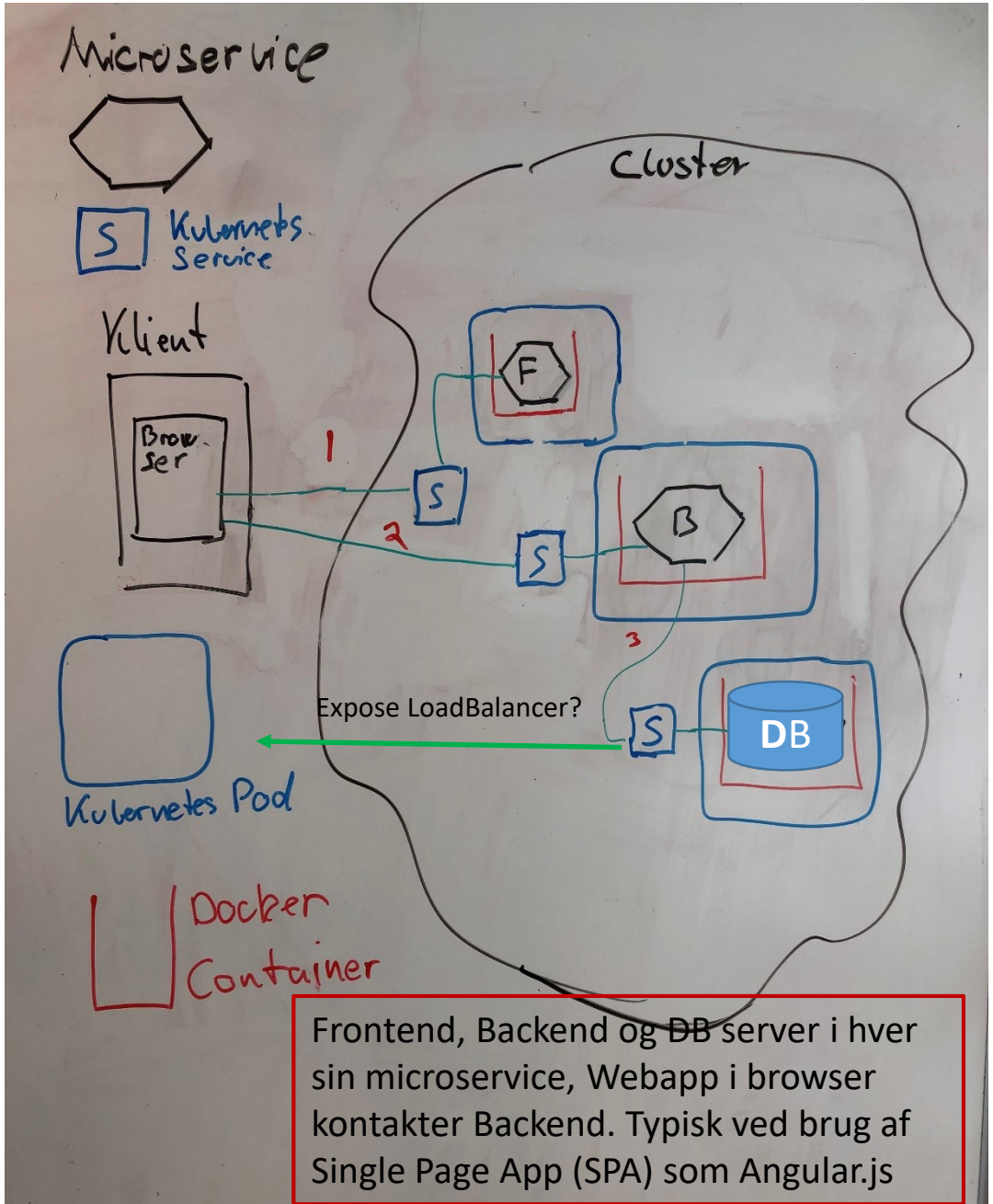
Hvorledes de tre services interagerer, kan der opstilles flere forskellige scenarier over afhængig af sammenbindingen/forbindelsen imellem de tre services. Se næste slide.

3: Forbindelser mellem microservices og db-server: Scenarie A og B

A



B



4: De tre udviklingstrin mht. microservice-platform

1. Trin: Microservices uden Docker Containers og dermed uden Kubernetes Pod.
2. Trin: Microservices med Docker Containers men uden Kubernetes Pod
   Her kan **docker-compose** indgå.
3. Trin: Microservices med Docker Container og med Kubernetes Pod.
   - Fra Docker til Kubectl
     - Link: https://kubernetes.io/docs/reference/kubectl/docker-cli-to-kubectl/

Trin 2 og 3 kan ”skaleres” under udvikling.

Krav: Alle tre trin bruges men 3. Trin skal bruges ved endelig aflevering af delprojektopgaven!

5: Hvilket service-forbindelses-scenarie skal bruges?

- Opgavens fokus er orkestrering af en webapplikation på et Kubernetes Cluster.
- **Delopgave Projekt afleveringen er derfor udførelsen af scenarie A eller B.**
- Gruppen kan arbejde iterativt ved først at udvikle simplere scenarier (Simplere orkestrering) for derefter udvikle A eller B.
  - Simplere scenarie. Kan være at FrontEnd og Backend er samme service. Kræver ikke yderligere programmering af Browser Klienten (WebApp), når en standard løsning for websider/webapplikation vælges (Razor Pages eller Multi Page App).
- Scenarie A og B. Kræver lidt ekstra programmering af Frontend og Backend. B kræver yderligere ekstra programmering i Browser Klient (Single Page Application: SPA).

6: Tekniske krav ASP.NET MVC Core

- I princippet kan platforme for Microservices og Database vælges frit.
- Men platform for Frontend og Backend services er som standard valgt til Microsoft .NET Core ASP.NET version 5.0 på Linux. Programmering i .NET 5.0 C#
  - Docker Images Link:
    - Link: https://hub.docker.com/_/microsoft-dotnet-aspnet
    - Link: https://hub.docker.com/_/microsoft-dotnet-core/
  - ASP.NET Core GitHub eksempler
    - https://github.com/dotnet/dotnet-docker/tree/master/samples/aspnetapp
  - .NET og GitHub Link: https://github.com/dotnet/dotnet-docker/issues/2375

7: Tekniske krav Database-microservice

- Database adgang er er som standard sat til et være udført med Microsoft ADO.NET Entity Frameword Core version
  - Link: https://docs.microsoft.com/da-dk/ef/#pivot=efcore
    - Providers Link: https://docs.microsoft.com/en-us/ef/core/providers/
      - Overvej som start at bruge ”In Memory Database”
  - Link: https://docs.microsoft.com/en-us/ef/core/
- Docker Image for MS SQL Server og MySQL
  - Link: https://hub.docker.com/_/microsoft-mssql-server
  - Link: https://hub.docker.com/_/mysql

// Start en MS SQL Server 2019 med Login SA og PW F21SWTDISP i en Docker Container med TCP/IP port 1433
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=F21SWTDISP" -p 1433:1433 -d mcr.microsoft.com/mssql/server:2019-latest

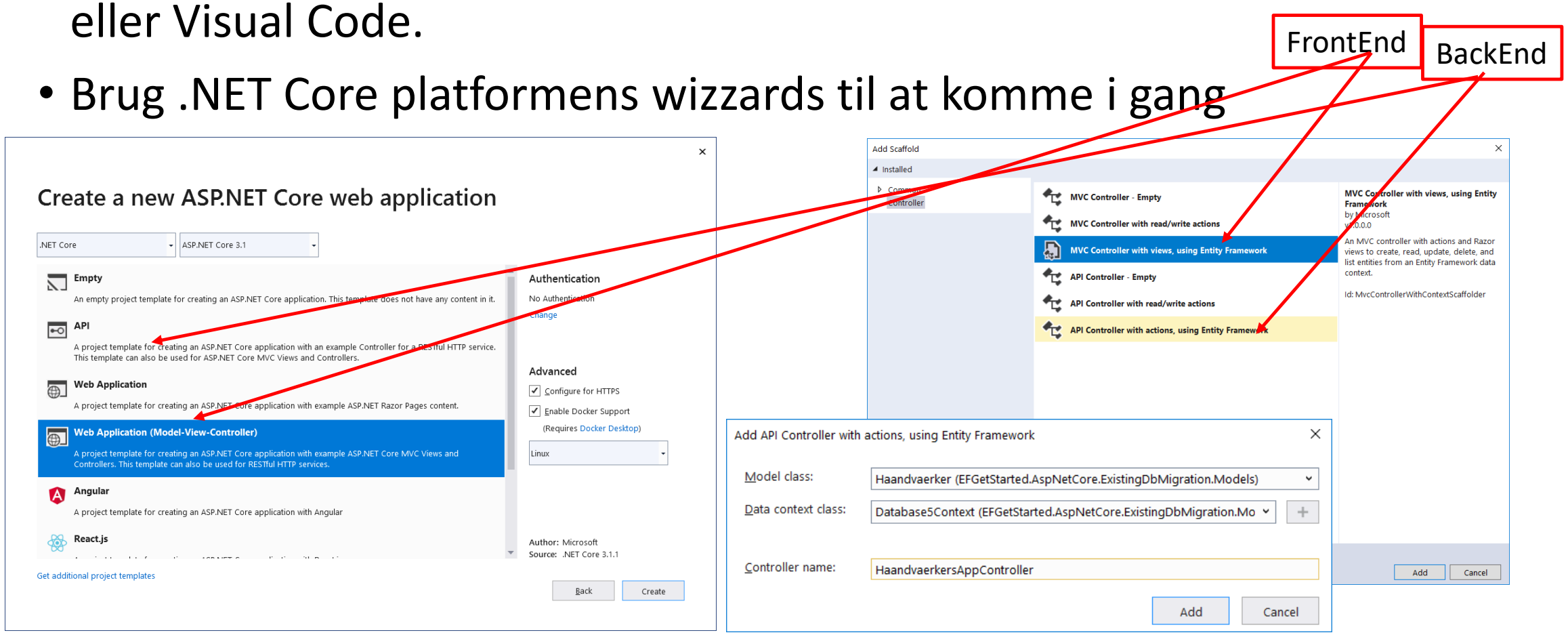
//Start en mysql rdb med root pw my-secret-pw standard TCP/IP port 3306
docker run --name Gr2swtdisp-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:latest

8:Deployment på Google Cloud Engine (GKE) Cluster

- Afht. til ressourcer på kursets GKE Cluster kan der, ved deployment hertil, blive tale om at benytte en fælledes Databaseserver (Eller en af to, hvis både MS SQL og MySQL servere benyttes).

9: ASP.NET MVC Core Service Udviklingsmiljø

- Backend og Frontend kan udføres med hjælp af Visual Studio 2019 eller Visual Code.
- Brug .NET Core platformens wizzards til at komme i gang



10: Docker

- Hvert enkelt image af de tre microservices der bruges: Backend, Frontend og Database, håndteres til start hver for sig.
- Docker Compose kan efterfølgende bruges til at starte de to til tre services op som applikation før deployment til Kubernetes
  - Følge Instruktioner i video sammen med opgaven på Blackboard.
- Frontend og Backend images lægges på DockerHub.
- Database-server image hentes fra Dockerhub.

11: Kubernetes

- Hver enkelt microservices lægges til start enkeltvis på Kubernetes Cluster som en Kubernetes Pod med en tilførende Kubernetes Service.
- Applikationen skal ligge i et Kubernetes Namespace kaldet ”**swtdisp-ap1-gr[x]**”eks. ”swtdisp-ap1-gr3”.
- Senere i forløbet uføres en YAML konfigurationsfil som en samlet deployment, som så vil kunne udføres på et andet Kubernets Cluster
- Alle images hentes fra DockerHub

12: Hint

- Opstil tre deployment-modeller på forhånd for implementation
  1. .NET Core Deployment uden Docker og Kubernetes.
  2. .NET Core Deployment med Docker evt. med Docker Compose
  3. .NET Core Deployment med Kubernetes.
- Navngiv komponenter entydigt i alle modeller
  - Herunder: Brug og vis Labels og Tags for Docker og Kubernetes i modellerne
- Husk også at medtage alle Kubernetes objekter, som Pod, Service, ReplicaSet og Deployment.

13: Aflevering og anden praktisk information

- Der arbejdes i F21SWTDISP projektgrupper.
- Der afleveres en zipped folder med ”udviklingsmiljøet” for Delprojektopgave applikationen. Navngives F21SWTDISPDelProjektGr[x].zip ***eks. F20SWTDISPDelProjektGr4.zip***
- De tre deployment modeller, med navne labels tag etc., afleveres i en PDF fil. Navngives F21SWTDISPDeployementGr[x].pdf ***eks. F21SWTDISPDeploymentGr4.pdf.*** Noter etc. må gerne tilføjes PDF filen.
- Der oprettes en opgaveaflevering på Blackboard.