

# Project #1 – Multi-Container App

## Assignment #1 – Containers Basics

3<sup>rd</sup> October 2024

### Advanced Infrastructures for Data Science

Pedro Neves, [pedroneves@dei.uc.pt](mailto:pedroneves@dei.uc.pt), 2024/2025

Master in Data Science and Engineering (MDSE) Course

# Project #1 – Multi-container App

Project Name: Tracking Goals Web Application

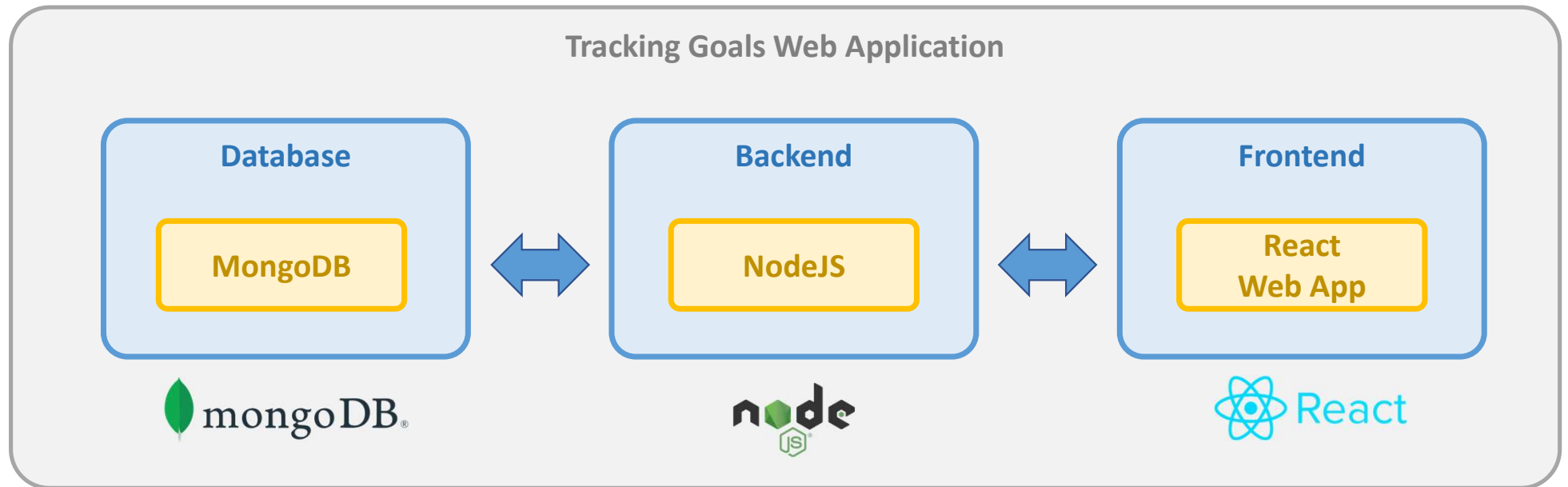
Project Objective: Deploy a multi-container app with Docker  
and Kubernetes

# Web Application Description

# Tracking Goal Web Application

App Objective, Architecture & Technologies

**App Objective:** Allows users to manage (create, view and delete) their goals



# Backend Web Application

## Source Code

Backend

NodeJS



Backend listening on  
port 80

```
EXPLORER
└─ MULTI-CONTAINER-GOALS-APP
   └─ backend
      ├── logs
      ├── models
      └─ JS app.js
         {} package-lock.json
         {} package.json
      └─ frontend

JS app.js
backend > JS app.js > ...
25 res.setHeader('Access-Control-Allow-Headers', 'Content-Type');
26 next();
27 });
28
29 app.get('/goals', async (req, res) => {
30   console.log('TRYING TO FETCH GOALS');
31   try {
32     const goals = await Goal.find();
33     res.status(200).json({
34       goals: goals.map((goal) => ({
35         id: goal.id,
36         text: goal.text,
37       })),
38     });
39     console.log('FETCHED GOALS');
40   } catch (err) {
41     console.error('ERROR FETCHING GOALS');
42     console.error(err.message);
43     res.status(500).json({ message: 'Failed to load goals.' });
44   }
45 });
46
47 app.post('/goals', async (req, res) => {
48   console.log('TRYING TO STORE GOAL');
49   const goalText = req.body.text;
50
51   if (!goalText || goalText.trim().length === 0) {
52     console.log('INVALID INPUT - NO TEXT');
53     return res.status(422).json({ message: 'Invalid goal text.' });
54   }
55 }
```

# Frontend Web Application

## Source Code

Frontend

React  
Web App



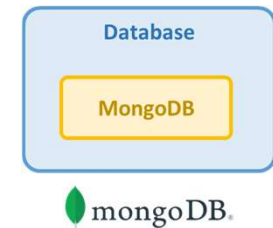
```
9   const [isLoading, setIsLoading] = useState(false);
10  const [error, setError] = useState(null);
11
12  useEffect(function () {
13    async function fetchData() {
14      setIsLoading(true);
15
16      try {
17        const response = await fetch('http://localhost/goals');
18
19        const resData = await response.json();
20
21        if (!response.ok) {
22          throw new Error(resData.message || 'Fetching the goals failed.');
```

Frontend listening  
on port 3000

Deploy Web Application  
@Host Machine

# Running @Host

## MongoDB

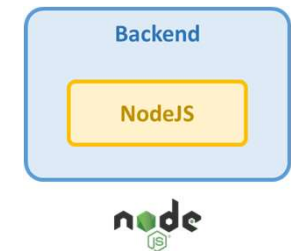


```
[user@user-virtualbox multi-container-goals-app]$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("cc34e550-84a3-4b4b-b8db-0a32421710ea") }
MongoDB server version: 3.6.8
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user
Server has startup warnings:
2022-09-20T15:53:58.978+0100 I STORAGE [initandlisten]
2022-09-20T15:53:58.978+0100 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-09-20T15:53:58.978+0100 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-09-20T15:54:09.567+0100 I CONTROL [initandlisten]
2022-09-20T15:54:09.567+0100 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-09-20T15:54:09.567+0100 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-09-20T15:54:09.567+0100 I CONTROL [initandlisten]
> db
test
> █
```



# Running @Host

## Backend



```
[user@user-virtualbox backend]$ npm install
```

```
up to date, audited 86 packages in 925ms
```

```
2 packages are looking for funding
  run `npm fund` for details
```

```
found 0 vulnerabilities
```

```
[user@user-virtualbox backend]$ sudo node app.js
```

```
[sudo] password for user:
```

```
CONNECTED TO MONGODB
```

```
TRYING TO FETCH GOALS
```

```
FETCHED GOALS
```

```
TRYING TO STORE GOAL
```

```
STORED NEW GOAL
```

```
TRYING TO DELETE GOAL
```

```
DELETED GOAL
```

```
TRYING TO STORE GOAL
```

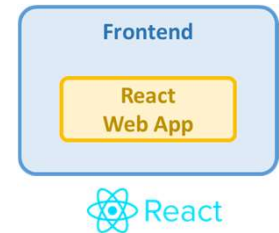
```
STORED NEW GOAL
```

```
TRYING TO STORE GOAL
```

```
STORED NEW GOAL
```

# Running @Host

## Frontend



```
[user@user-virtualbox frontend]$ npm start
```

```
> docker-frontend@0.1.0 start
> react-scripts start
```

```
Compiled successfully!
```

```
You can now view docker-frontend in the browser.
```

```
Local: http://localhost:3000
```

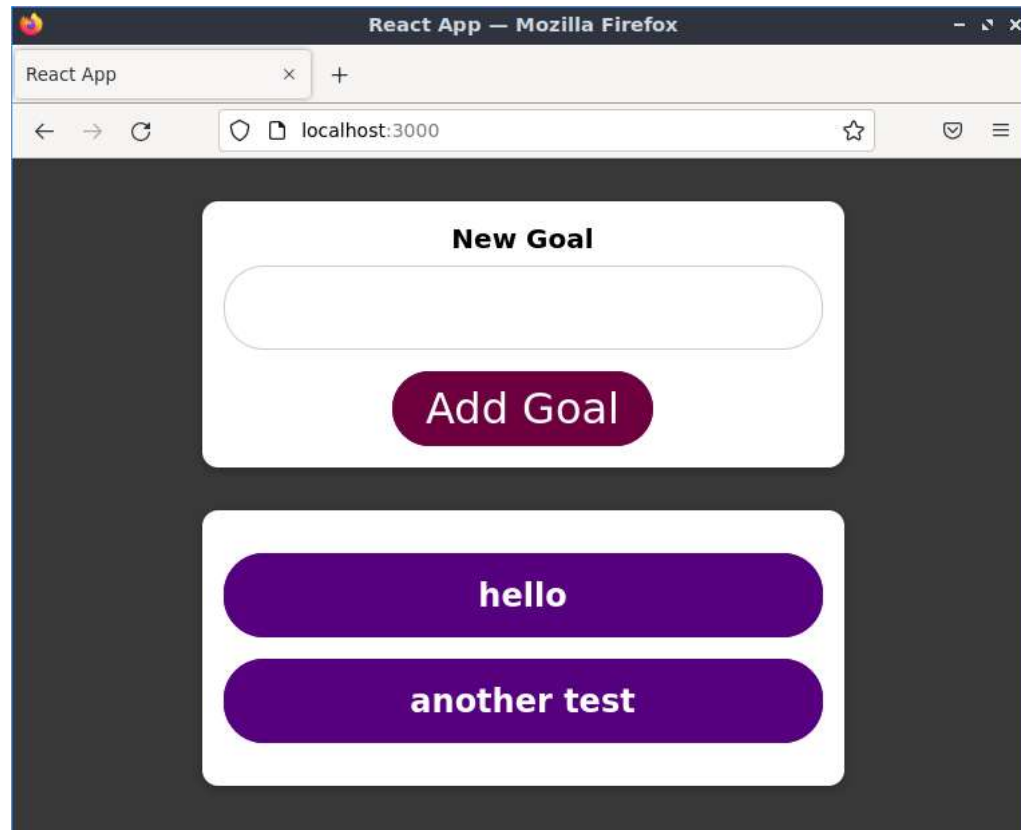
```
On Your Network: http://10.0.2.15:3000
```

```
Note that the development build is not optimized.
To create a production build, use npm run build.
```

```
webpack compiled successfully
```

# Running @Host

Using the App – Graphical User Interface



# Assignment #1

# Project #1 Assignments

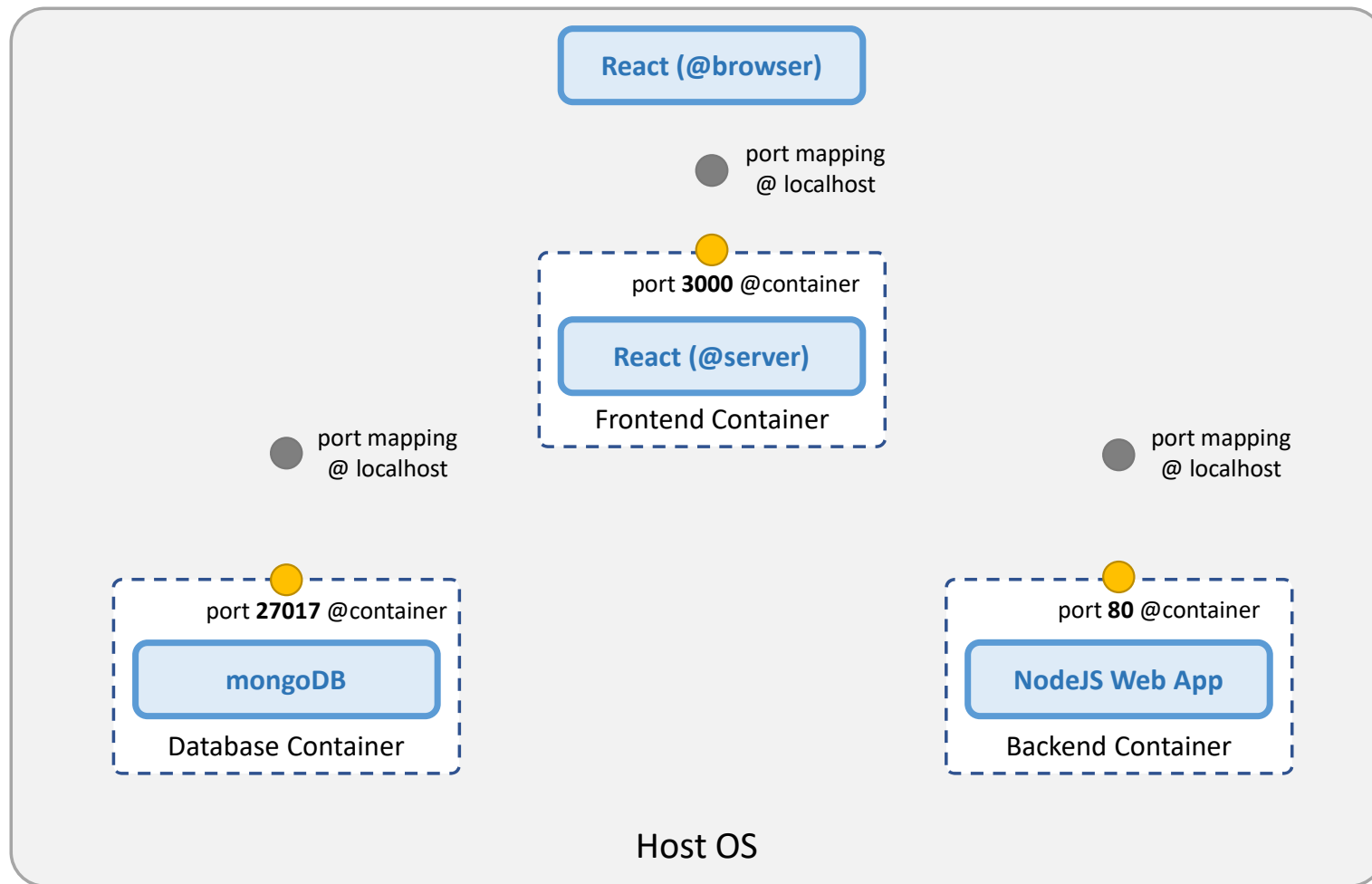
- Project #1
  - Assignment #1 (3<sup>rd</sup> October): Build images and launch containers

# Assignment #1

## Tasks

1. Pre-requisite: Install Docker
2. **Create** and/or **fetch** the required **images** for the multi-container web app
3. **Run** the web app **containers in standalone mode (no interworking yet)**

# Assignment #1 Output





**Good  
Work**