Module Distributed Systems and Blockchain 1
FS 2021

CHALLENGE-TASK PRESENTATION:

SPLITFAIR

Thomas Zahner, Marc Eberhard

Rapperswil, 25. Mai 2021

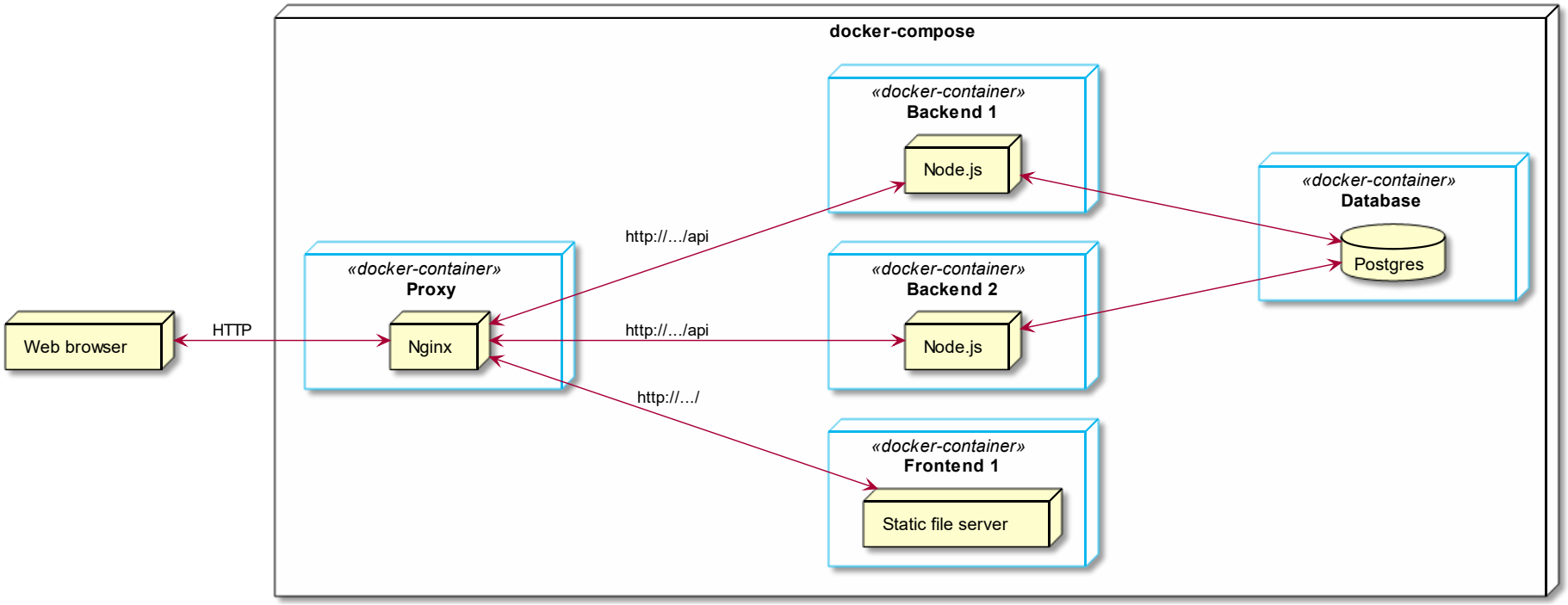
Agenda

- **About us**
- **Distributed System**
  - Components used
  - Architecture
- **Demo**
- **Jwt Authentication**
- **Challenge Requirements**
- **Lessons learned**
- **Questions**

Components used

- **Docker(-compose)**
- **NGINX**
- **PostgreSQL**
- **NodeJS:** express, pg, cookie-parser, jsonwebtoken
- **Vue.Js -> HTML, CSS, JavaScript**

Architecture



Architecture for development

docker-compose -f docker-compose.dev.yml up

- **Directories mounted into containers**
- **Backend: nodemon**
- **Frontend: webpack**

Architecture for production

docker-compose up --build --scale backend=3 --scale frontend=2

- **Directories copied when building the image**
- **Backend: node**
- **Frontend: build static files, serve with static file server (serve)**

DEMO

JWT Authencitaction

index.js

```
// Middleware functions
app.use(bodyParser.json());
app.use(cookieParser());
app.use(express.urlencoded( options: { extended: true }));
app.use(authenticateJwtCookie(uncheckedJwtRoutes));
app.use(logRequests);

// Database
const db = new Database();
await db.connect();

// Routes
app.use('/api', setupRoutes(db));
```

middlewareAuthentication.js

```
import jwt from 'jsonwebtoken';

const secret = process.env.JWT_SECRET;

if (!secret) {
  throw new Error('Environment variable JWT_SECRET needs to be defined');
}
```

```
export function setJwtCookie(data, res) {
  res.cookie('jwt', jwt.sign(data, secret), {
    httpOnly: true, // security: don't expose the cookie to the JavaScript
    sameSite: 'strict', // security: mitigate CSRF
    secure: false,
  });
}

export function authenticateJwtCookie(excludedRoutes) {
  return function (req, res, next) {
    const token = req.cookies?.jwt;
    const skipCheck = excludedRoutes.includes(req.path);

    if (!token && !skipCheck) {
      console.error('No cookie provided');
      return res.status(401).end();
    }

    jwt.verify(token, secret, options: (err, decodedData) => {
      if (err && !skipCheck) {
        console.error(err);
        return res.status(401).end();
      } else {
        req.jwtData = decodedData;
        return next();
      }
    });
  }
}
```

NGNIX

```
events {}


http {
    server{
        listen 80;


        location / {
            proxy_pass http://frontend:8080;
        }


        location /api {
            proxy_pass http://backend;
        }
    }
}
```

Challenge Requirements

- ✓ **Simple frontend: vue.js**
- ✓ **LoadBalancing: NGINX**
- ✓ **Multiple instances of a service: backend**
- ✓ **Storage backend: PostgreSQL**
- ✓ **JWT: jsonwebtoken to generate and verify, send and store as cookie**

Lessons learned

- **Thomas**
  - JWT
  - Express validator
  - Docker-Compose, Dockerfiles
- **Marc**
  - Vue.JS, JWT
  - JavaScript in backend, Containerized database
  - Read and respect the errors/warnings

Questions?

- www.splitfair.ch