

Distributed Systems & Blockchain

URL Shortener

Marc Emch & Diluxion Marku & Samuel Lemmenmeier

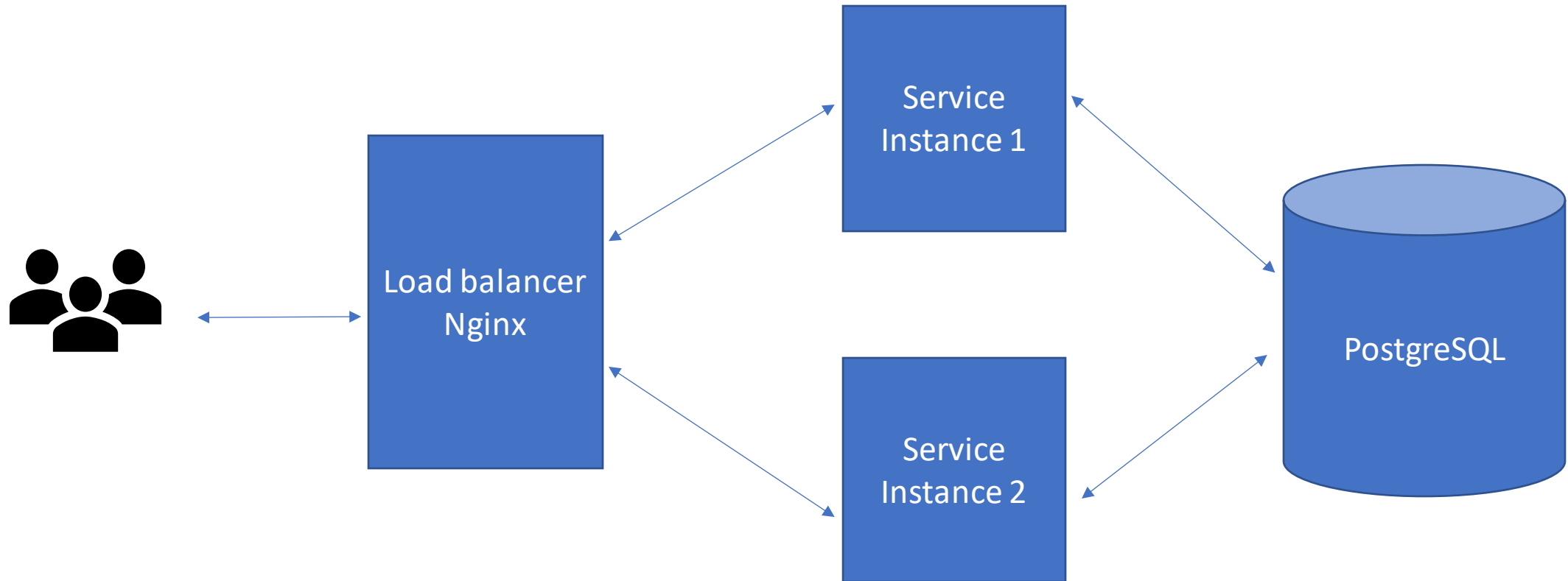
Motivation & Goals

- Testat
- Something simple and achievable
- Using technologies familiar to us
- Something extensible

Demonstration



Architecture



Technologies, Libraries, Frameworks

- Docker
- Load balancer: Nginx
- Service: Node instance
- Database: PostgreSQL

Load balancer: Nginx

- "Off-the-shelf" web server
- Wir mussten nur konfigurieren
 - Essentials

```
pid run/nginx.pid;  
error_log run/error.log warn;  
worker_rlimit_nofile 1024;  
  
http{  
    access_log run/access.log combined;  
  
    upstream node {  
        least_conn;  
        server 127.0.0.1:3000;  
        server 127.0.0.1:3001;  
    }  
  
    server {  
        listen 8080;  
  
        location / {  
            proxy_pass http://node;  
            proxy_set_header X-Forwarded-Proto $scheme;  
            proxy_set_header X-Forwarded-Host $host;  
            proxy_set_header X-Forwarded-Port $server_port;  
        }  
    }  
}  
  
events{  
}
```

Load balancer: Nginx

- "Off-the-shelf" web server
- Wir mussten nur konfigurieren
 - Essentials
 - **Listening end**

```
pid run/nginx.pid;
error_log run/error.log warn;
worker_rlimit_nofile 1024;

http{
    access_log run/access.log combined;

    upstream node {
        least_conn;
        server 127.0.0.1:3000;
        server 127.0.0.1:3001;
    }

    server{
        listen 8080;

        location /{
            proxy_pass http://node;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_set_header X-Forwarded-Host $host;
            proxy_set_header X-Forwarded-Port $server_port;
        }
    }
}

events{
}
```

Load balancer: Nginx

- "Off-the-shelf" web server
- Wir mussten nur konfigurieren
 - Essentials
 - Listening end
 - Passthrough

```
pid run/nginx.pid;
error_log run/error.log warn;
worker_rlimit_nofile 1024;

http{
    access_log run/access.log combined;

    upstream node {
        least_conn;
        server 127.0.0.1:3000;
        server 127.0.0.1:3001;
    }

    server{
        listen 8080;

        location /{
            proxy_pass http://node;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_set_header X-Forwarded-Host $host;
            proxy_set_header X-Forwarded-Port $server_port;
        }
    }
}

events{
}
```

Load balancer: Nginx

- "Off-the-shelf" web server
- Wir mussten nur konfigurieren
 - Essentials
 - Listening end
 - Passthrough
 - Target

```
pid run/nginx.pid;
error_log run/error.log warn;
worker_rlimit_nofile 1024;

http{
    access_log run/access.log combined;

    upstream node {
        least_conn;
        server 127.0.0.1:3000;
        server 127.0.0.1:3001;
    }

    server {
        listen 8080;

        location / {
            proxy_pass http://node;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_set_header X-Forwarded-Host $host;
            proxy_set_header X-Forwarded-Port $server_port;
        }
    }
}

events {
}
```

Load balancer: Nginx

- "Off-the-shelf" web server
- Wir mussten nur konfigurieren
 - Essentials
 - Listening end
 - Passthrough
 - Target
- Load balancing: **least_conn**
 - hash/ip_hash
 - random

```
pid run/nginx.pid;
error_log run/error.log warn;
worker_rlimit_nofile 1024;

http{
    access_log run/access.log combined;

    upstream node {
        least_conn;
        server 127.0.0.1:3000;
        server 127.0.0.1:3001;
    }

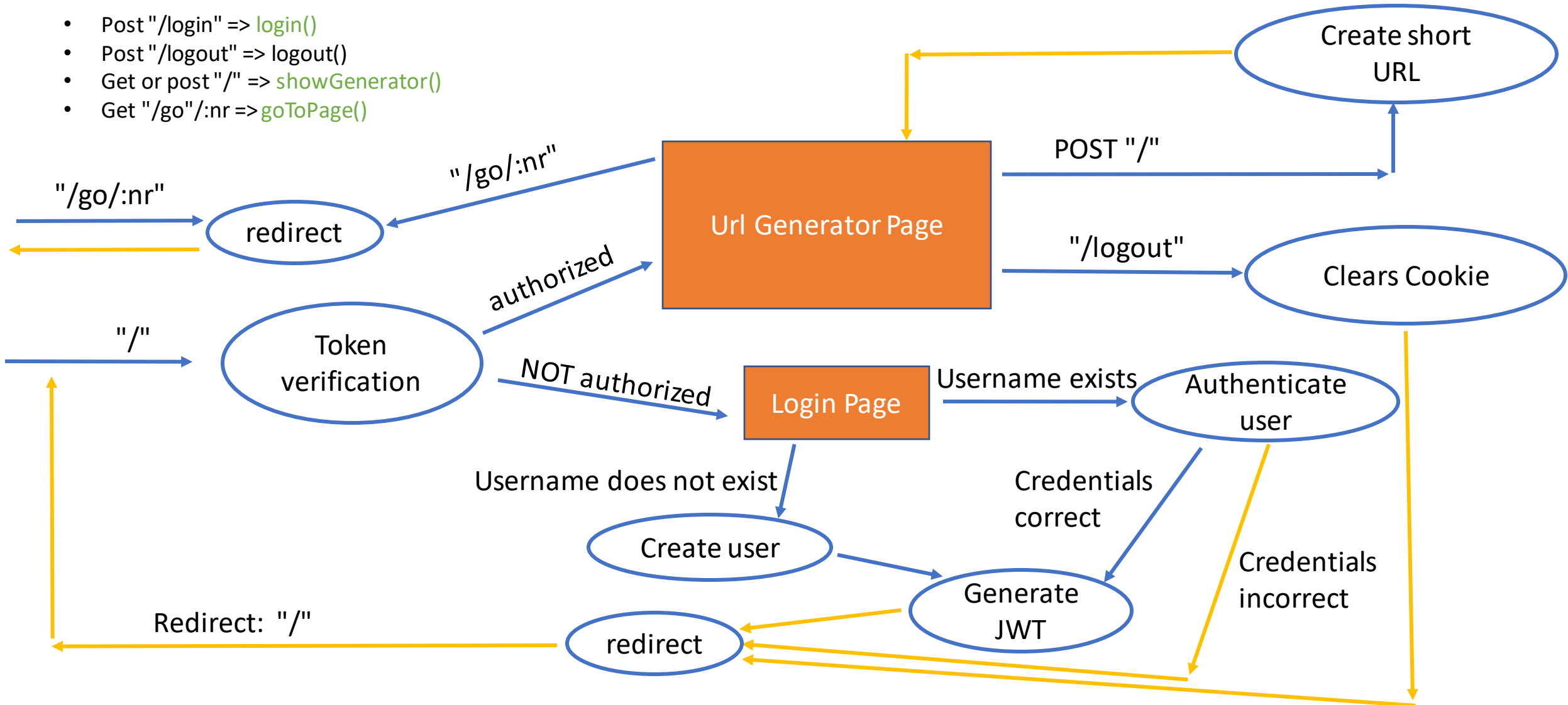
    server{
        listen 8080;

        location /{
            proxy_pass http://node;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_set_header X-Forwarded-Host $host;
            proxy_set_header X-Forwarded-Port $server_port;
        }
    }
}

events{
}
```

Service: Node Instance

- Post "/login" => login()
- Post "/logout" => logout()
- Get or post "/" => showGenerator()
- Get "/go"/:nr => goToPage()

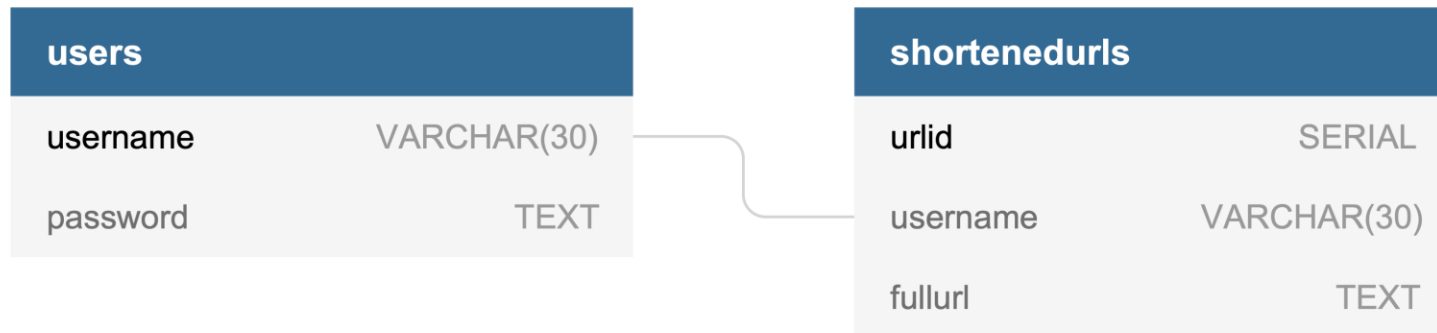
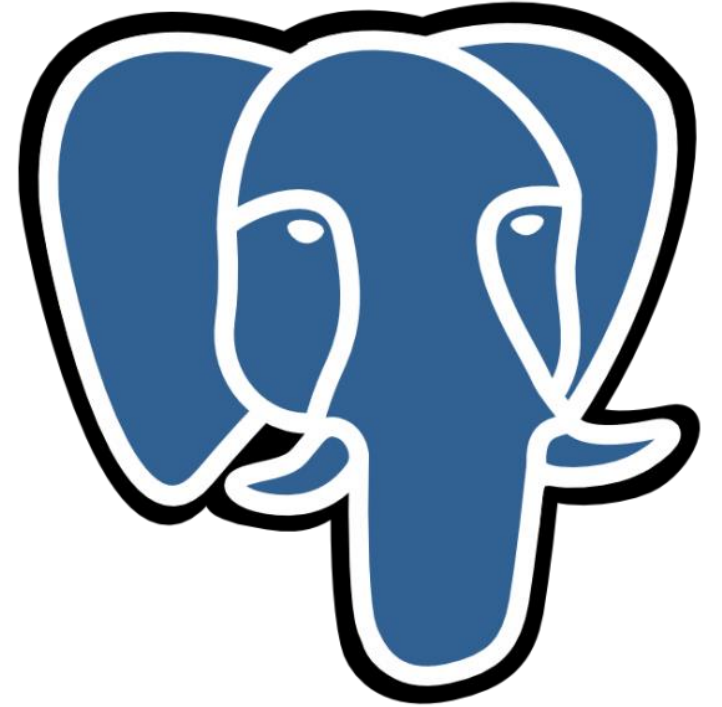


Service: Node Instance

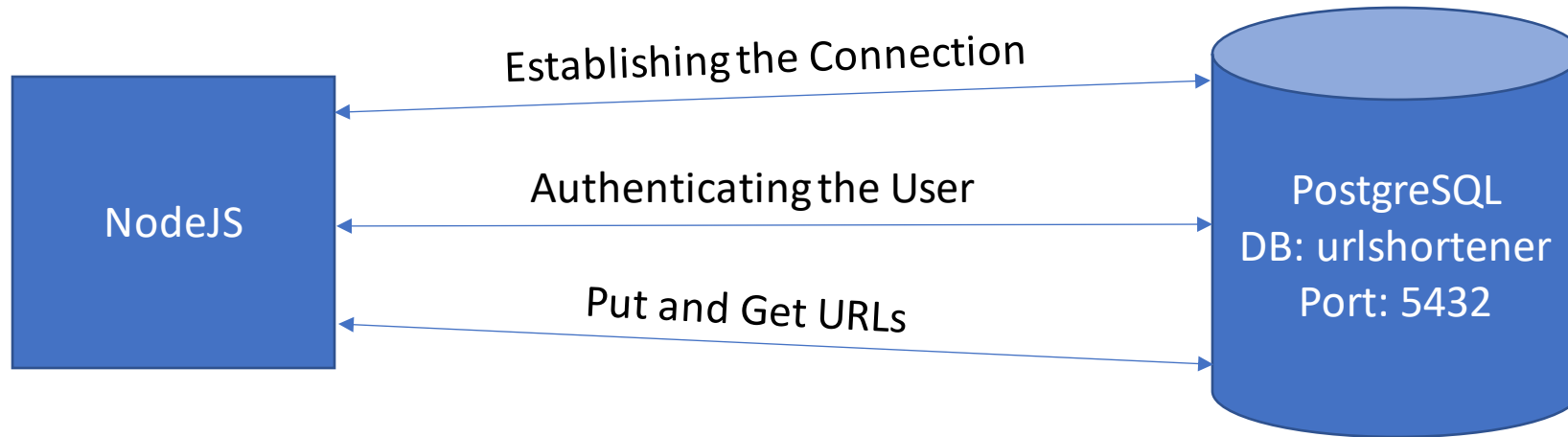
- Node.js
- Express Framework
- Core-Modules: http, path, crypto
- Modules: cookie-parser, jsonwebtoken
- JWT-Token: generated/validated in node instance => stateless server
=> scaling
- Cookie to store JWT-Token => stateless server => scaling

Database: PostgreSQL

- One database and two tables
- Table for the users
- Table for the URLs



Database: Connection and Interaction



Generating Short URLs

Group22: URL-Shortener

[Logout](#)

Enter a url to generate a short one

<https://example.com>

Generate

You have entered

<https://example2.com>

Your short url

<http://localhost:8080/go/24>

You are logged in as exampleUser

long Url

<https://example.com>

<https://example1.com>

<https://example2.com>

short Url

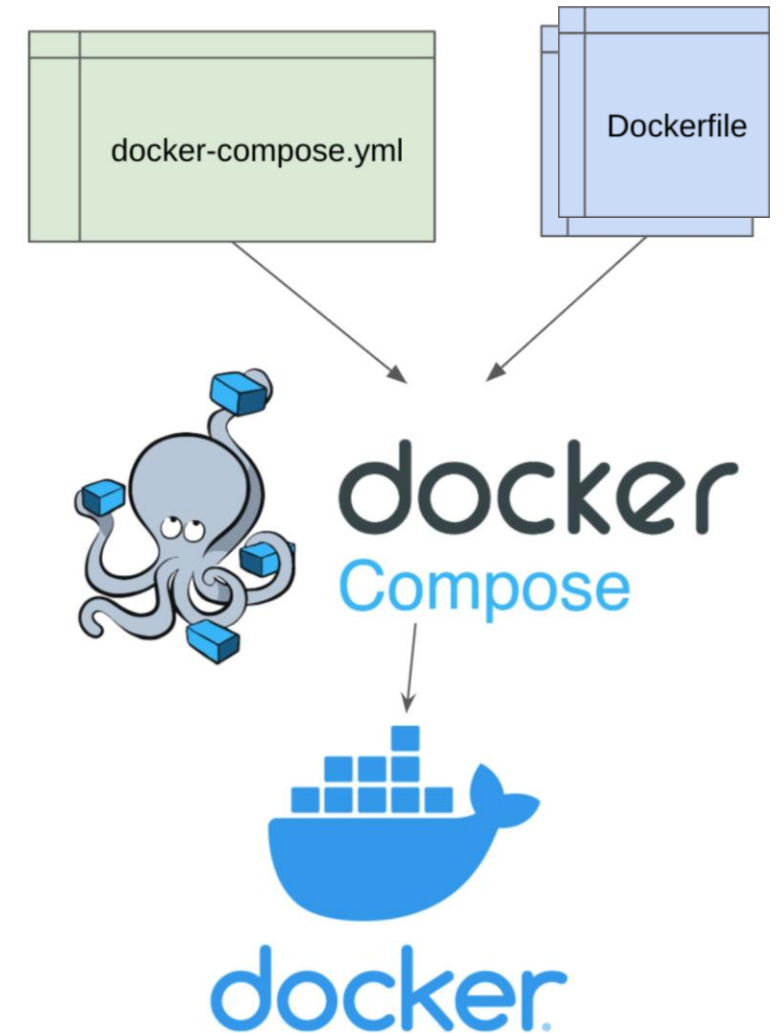
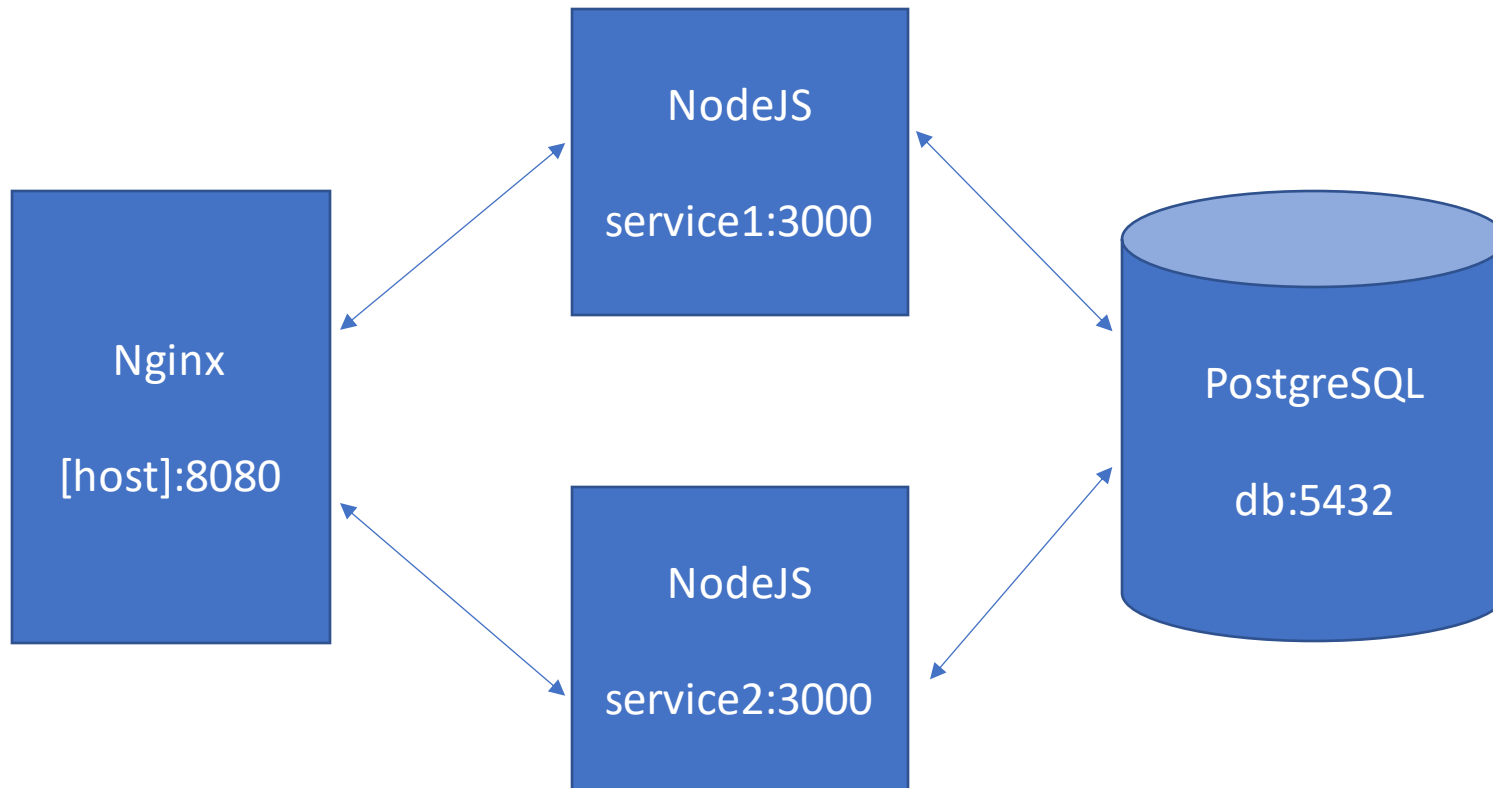
<http://localhost:8080/go/22>

<http://localhost:8080/go/23>

<http://localhost:8080/go/24>

Docker

- Debian als Basis



Requirements

- Load balancing: Nginx
- Scalable service: Docker
- JWT authentication: Yes
- Dockerized: Yes
- Use latest stable releases of chosen libraries and frameworks: Yes