
Basic Health Tracker

*Andreas Bachmann, Israel
Leuenberg, André Schmid*

Inhalt

Idee

Architektur

Demo

Q&A

Idee

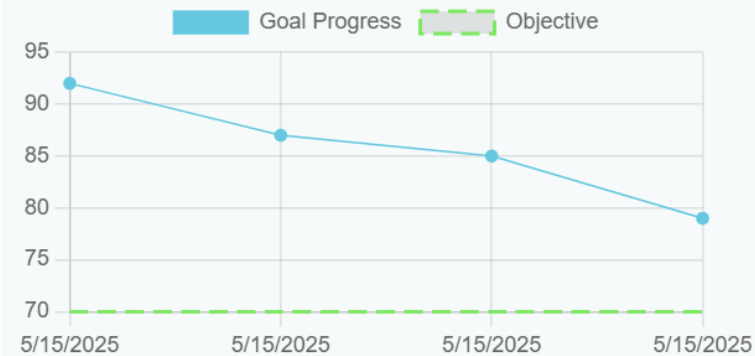
Basic Health Tracker

Weight

Goal: 70 kg

Last Progress: 79 kg

Progress



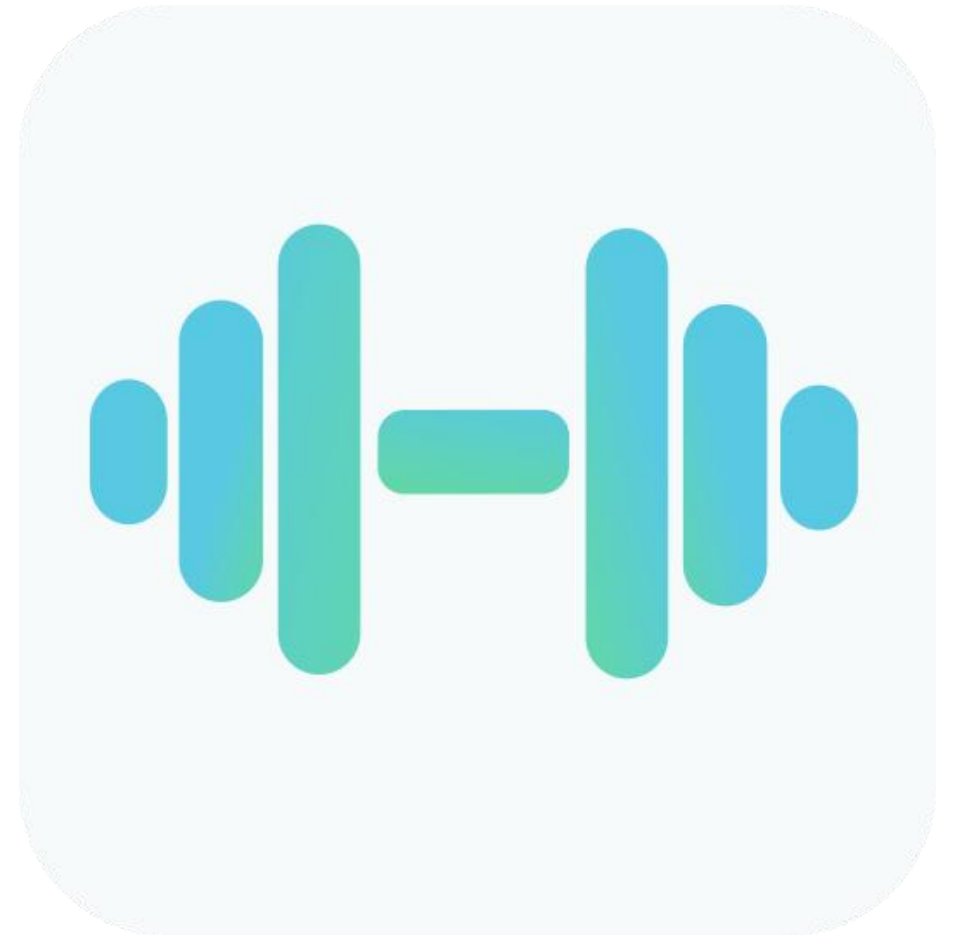
Delete

View Progress

Back

Idee

- Gesundheits- oder Leistungsparameter wie Gewicht, Ausdauer oder BMI tracken
- Persönliche Ziele festlegen
- Fortschritt mitverfolgen



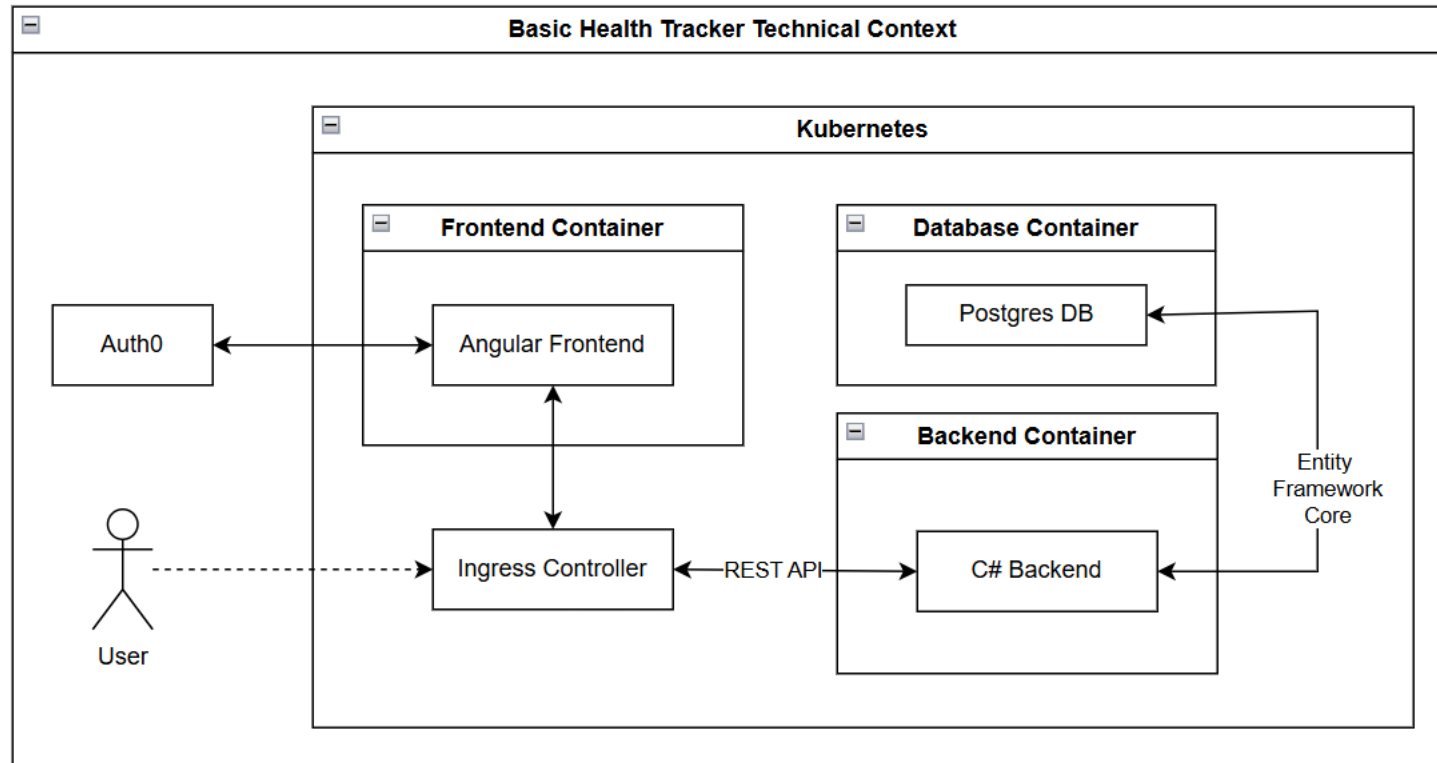
Anforderungen

- Load balancing mit mehreren Instanzen
- Dockerized
- SSE ausgelöst, der das Speichern eines Eintrages bestätigt
- Persistente Speicherung
- Skalierbar auf mehrere Pods



Architektur

Deployment



```

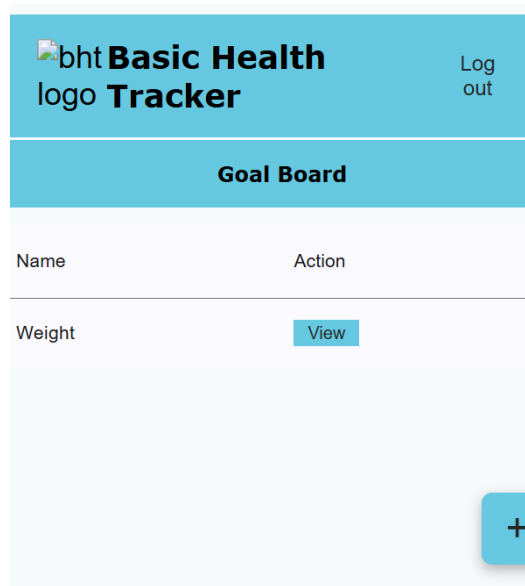
references | Israel Leuenberg, 34 days ago | 1 author, 1 change
public GoalService(AppDbContext context)

    _context = context;

references | Israel Leuenberg, 34 days ago | 1 author, 2 changes
public IEnumerable<GoalDto> GetGoalsByUserId(st

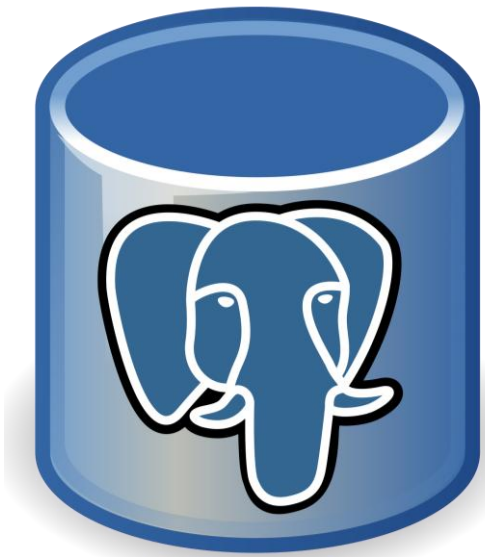
var goalsByUser = _context.Goals.Where(g =>
if (goalsByUser == null || !goalsByUser.Any
{
    return new List<GoalDto>();
}
return goalsByUser.Select(g => new GoalDto
{
    GoalId = g.GoalId,
    Name = g.Name,
    Unit = g.Unit,
    TargetValue = g.TargetValue,
    OwnerId = g.OwnerId
}).ToList();

```

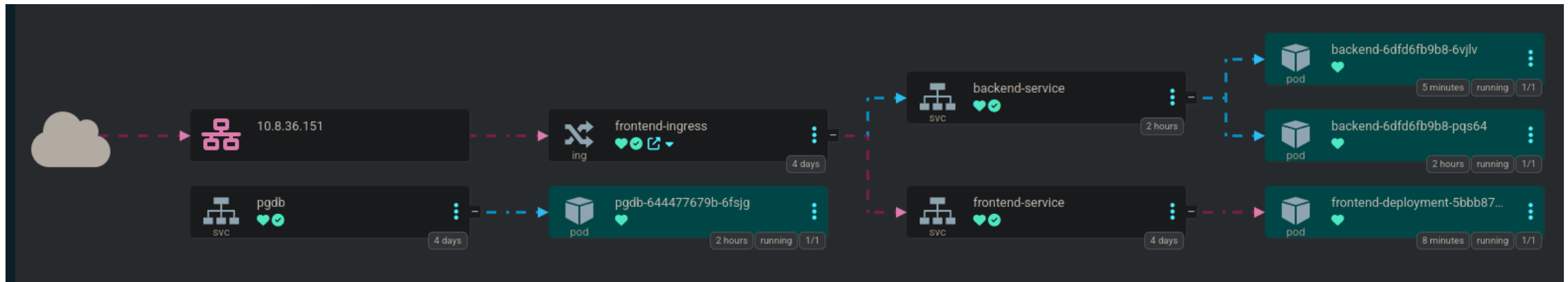


Komponenten

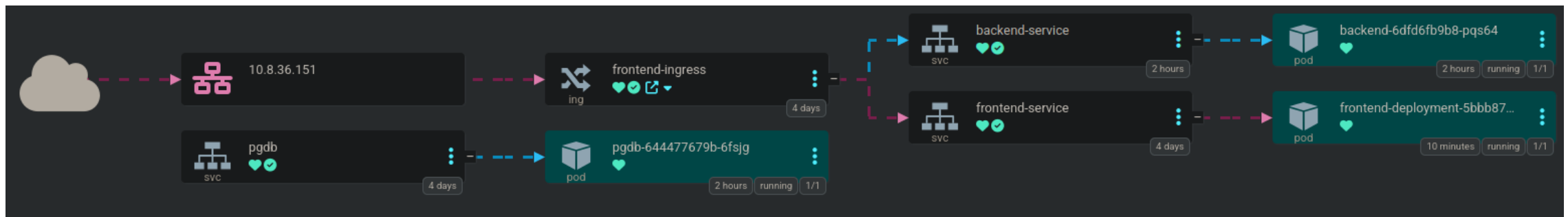
- Orchestration mit Kubernetes
- Dockerized
- Loadbalancing mit MetalLB
- C# Backend mit Entity Framework (Stateless)
- Angular Frontend
- Postgres DB



Scaling & Networkflow



Pod ausgeschaltet:



Server Side Event

Backend

Frontend

```
1 using backend.Services.Event;
2 using Microsoft.AspNetCore.Mvc;
3
4 namespace backend.Controllers
5 {
6     [ApiController]
7     [Route("api/events")]
8     public class EventsController : ControllerBase
9     {
10         private readonly ISseService _sseService;
11
12         public EventsController(ISseService sseService)
13         {
14             _sseService = sseService;
15         }
16
17         [HttpGet("subscribe")]
18         public async Task Subscribe()
19         {
20             Response.Headers.Add("Content-Type", "text/event-stream");
21             _sseService.AddClient(Response);
22
23             while (!HttpContext.RequestAborted.IsCancellationRequested)
24             {
25                 await Task.Delay(1000);
26             }
27         }
28     }
29 }
```

Controller

```
2 references | Andre, 6 days ago | 1 author, 1 change
public async Task SendEventAsync(string data)
{
    var bytes = Encoding.UTF8.GetBytes($"data: {data}\n\n");
    foreach (var client in _clients)
    {
        if (!client.HttpContext.RequestAborted.IsCancellationRequested)
        {
            try
            {
                await client.Body.WriteAsync(bytes, 0, bytes.Length);
                await client.Body.FlushAsync();
            }
            catch
            {
                Console.WriteLine("there was an error during the server side event");
            }
        }
    }
}
```

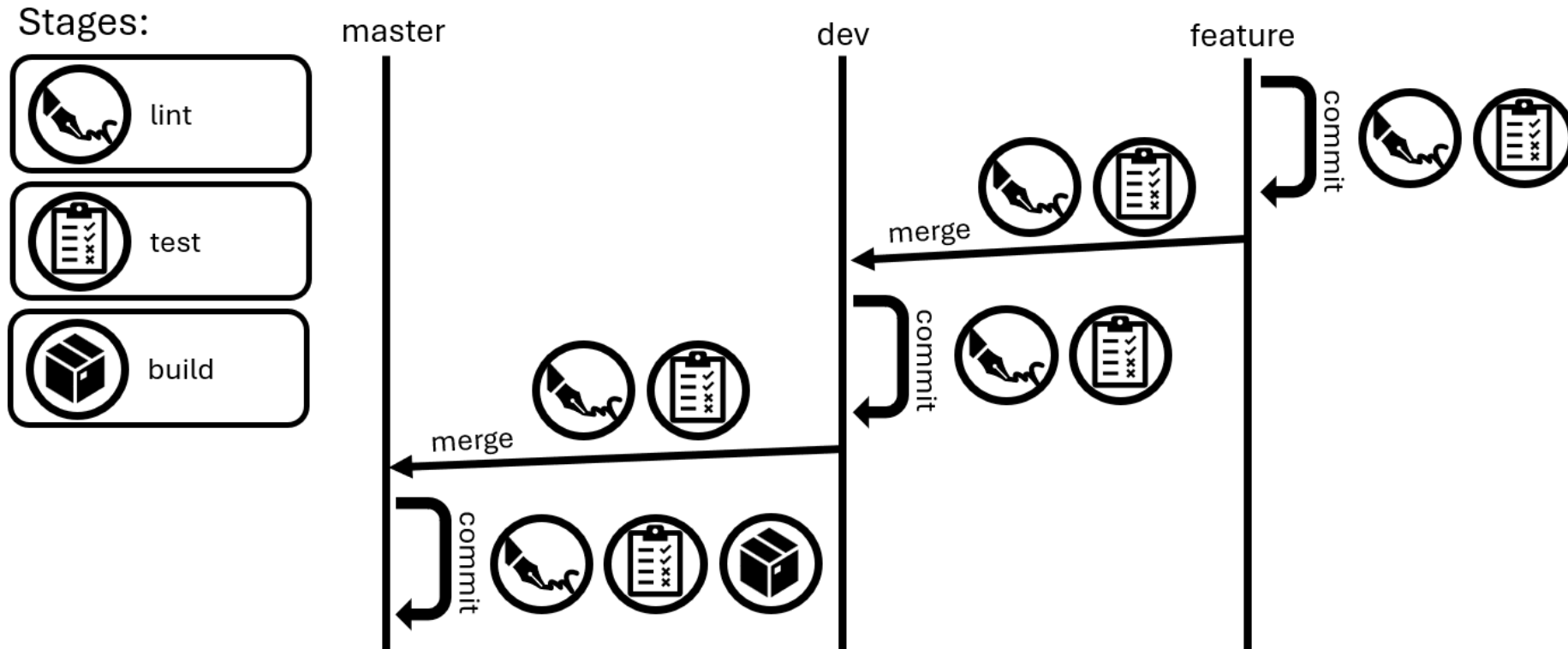
Service

```
this.sseService.getServerSentEvent('http://localhost:8081/api/events/subscribe')
    .subscribe({
        next: msg => {
            this.snackBar.open(`SSE received: ${msg}`, 'Close', { duration: 3000 });
        },
        error: err => console.error('SSE error:', err)
    });
```

Subscription

SSE received: Goal progress created: 77 Close

Pipeline



Demo

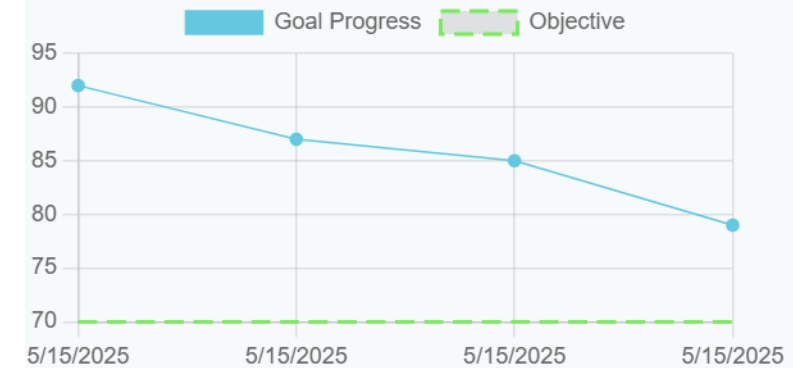
Basic Health Tracker

Weight

Goal: 70 kg

Last Progress: 79 kg

Progress



Delete

View Progress

Back



Q&A