

Full-Stack Revolution: Streamlining Cloud-Native App Development with **NEXT**.js

Ryan Elia

Software Engineer



Open Tech Summit Indonesia 2024

Jakarta, Indonesia

25 November 2024



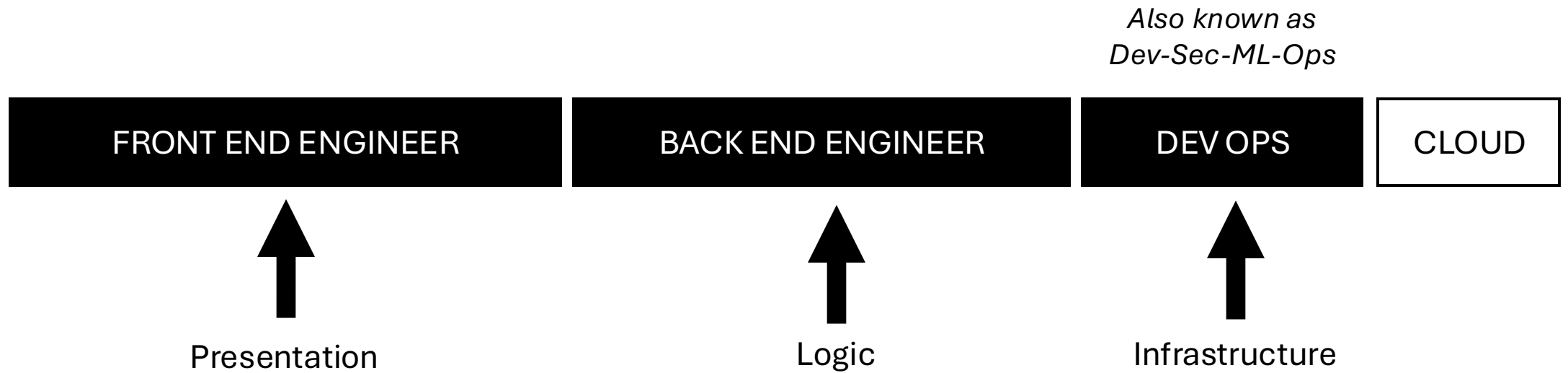
A long time ago in a galaxy far, far away...



One does not simply build a website



Today: This is Fine 😊 🔥



Challenges faced by a team of teams

- **Silos**

- Each team operates in their own domain.
- Requiring constant meetings for alignment. Slower decision making, miscommunication.

- **Dependency Bottleneck**

- One team's delay affect other teams.
- For example, Front-End team waits for API updates from the Back-End team before they can deploy the UI updates.

- **Lack of end-to-end ownership**

- Front-End team struggles to fix issues originating from the Back-End or Infrastructure
- Back-End team is detached from Front-End constraints and UX
- DevOps team does not understand the application logic

Tool and Languages Fragmentation

- **Front-End:**

- TypeScript / JS
- React, Vue, Svelte, Angular, Solid, etc.

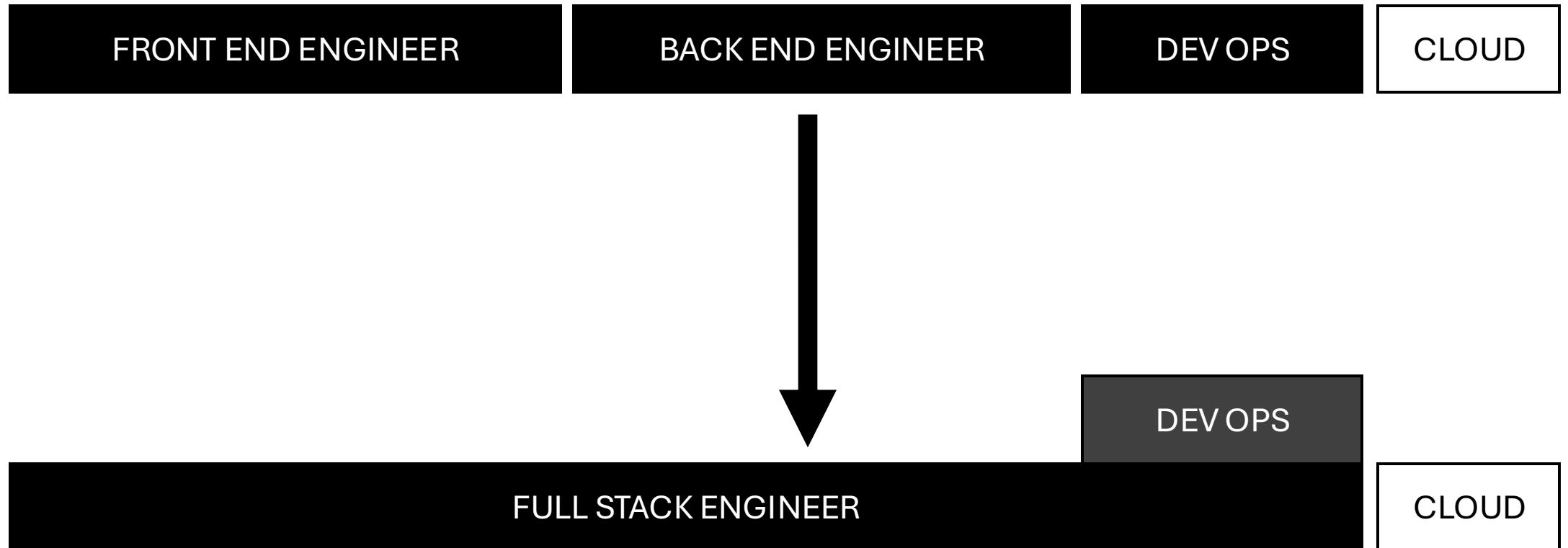
- **Back-End:**

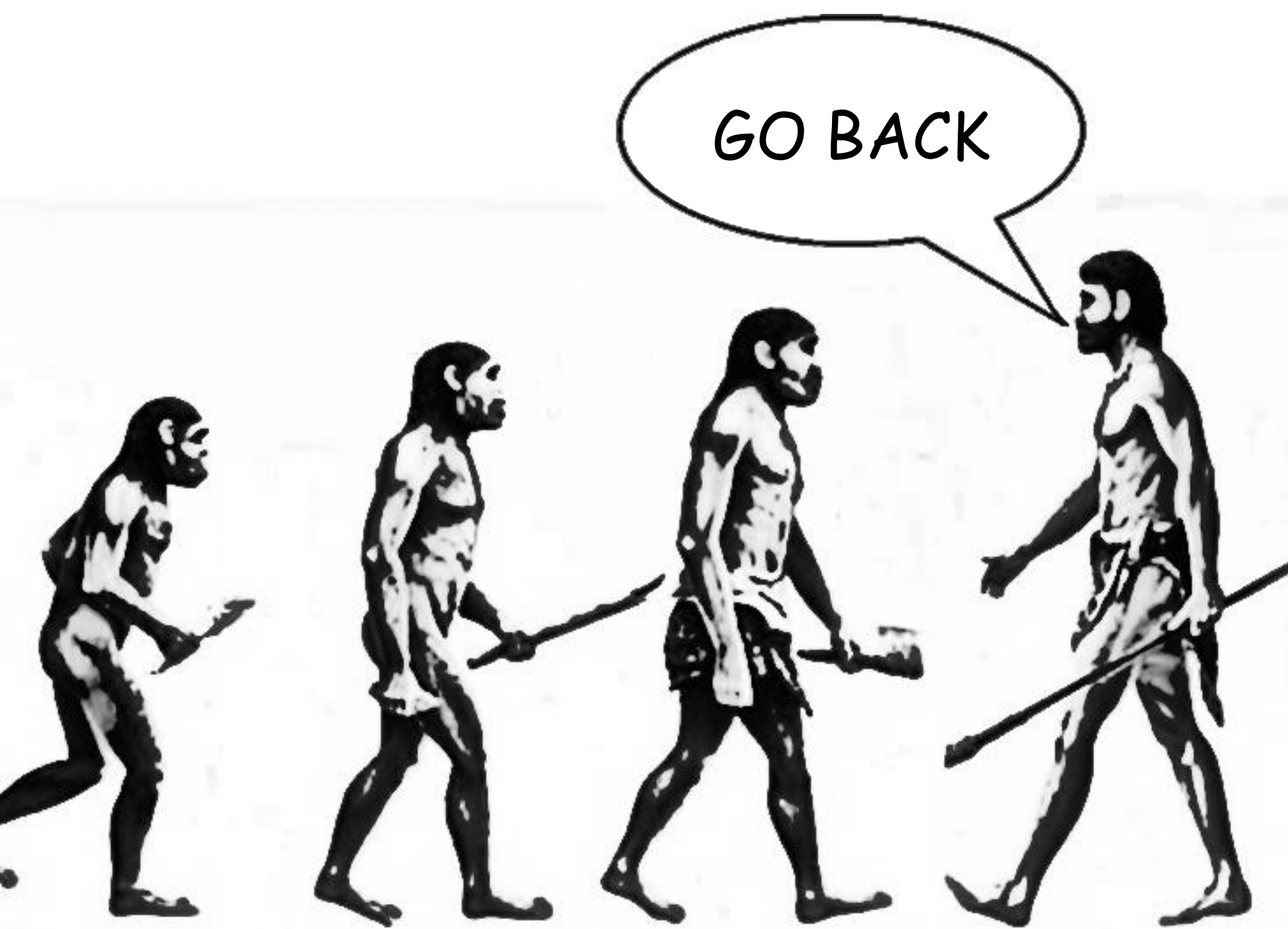
- Go / Python / .NET / Java / Node.js

- **DevOps:**

- Kubernetes / Docker YAML
- Terraform, AWS CLI

Today's talk:







The React Framework for the Web

Used by some of the world's largest companies, Next.js enables you to create **high-quality web applications** with the power of React components.

[Get Started](#)[Learn Next.js](#)

▲ ~ `npx create-next-app@latest`

What is Next.js

- Contrary to popular belief, it is not a front-end application framework.
- It is a back-end application framework that serves a React application
- It has opinionated, good defaults for routing the pages in the application. (File-Based Routing)
- It has API routes that allows you to write JSON Web API
- It manages the compiler and build system so you do not have to 😊

[Docs](#)[FAQ](#)

★ 25K



The best way to start a full-stack, typesafe Next.js app

[Documentation >](#)[GitHub](#)

```
npm create t3-app@latest
```



Open Source: <https://github.com/t3-oss/create-t3-app>



The best of the full stack TypeScript ecosystem...

...but **ONLY** the parts you need

Take what you want and nothing more!



Next.js

Next.js offers a lightly opinionated, heavily optimized approach to creating applications using React. It's the industry standard and we're proud to build on top of it :)



TypeScript

We firmly believe TypeScript will help you be a better web developer. Whether you're new to JS or a seasoned pro, the "strictness" of TypeScript leads to smoother building.



tRPC

If your frontend and backend are TypeScript, it's really hard to beat the DX of tRPC. Kinda like GraphQL but without the work - seriously this lib is magic.



Prisma

Prisma is the best way to work with databases in TypeScript. It provides a simple, type-safe API to query your database, and it can be used with most SQL dialects (and Mongo too!).



Tailwind CSS

Tailwind CSS is a utility-first CSS framework that helps you build beautiful, responsive designs without any extra configuration. It's built with utility-first principles, and is completely customizable and extendable.



NextAuth.js

When you need flexible, secure, and scalable auth, NextAuth.js is top notch. It ties into your existing database and provides a simple API to manage users and sessions.

```
◇ What will your project be called?  
open-tech-summit-indonesia-2024  
◇ Will you be using TypeScript or JavaScript?  
TypeScript  
◇ Will you be using Tailwind CSS for styling?  
Yes  
◇ Would you like to use tRPC?  
Yes  
◇ What authentication provider would you like to use?  
None  
◇ What database ORM would you like to use?  
Prisma  
◇ Would you like to use Next.js App Router?  
Yes  
◇ What database provider would you like to use?  
PostgreSQL  
◇ Should we initialize a Git repository and stage the changes?  
Yes  
◇ Should we run 'pnpm install' for you?  
Yes  
◇ What import alias would you like to use?  
~/
```

pnpm create t3-app@latest

./start-database.sh

Database container 'open-tech-summit-indonesia-2024-postgres' was successfully created

pnpm db:push

[✓] Pulling schema from database...

[✓] Changes applied

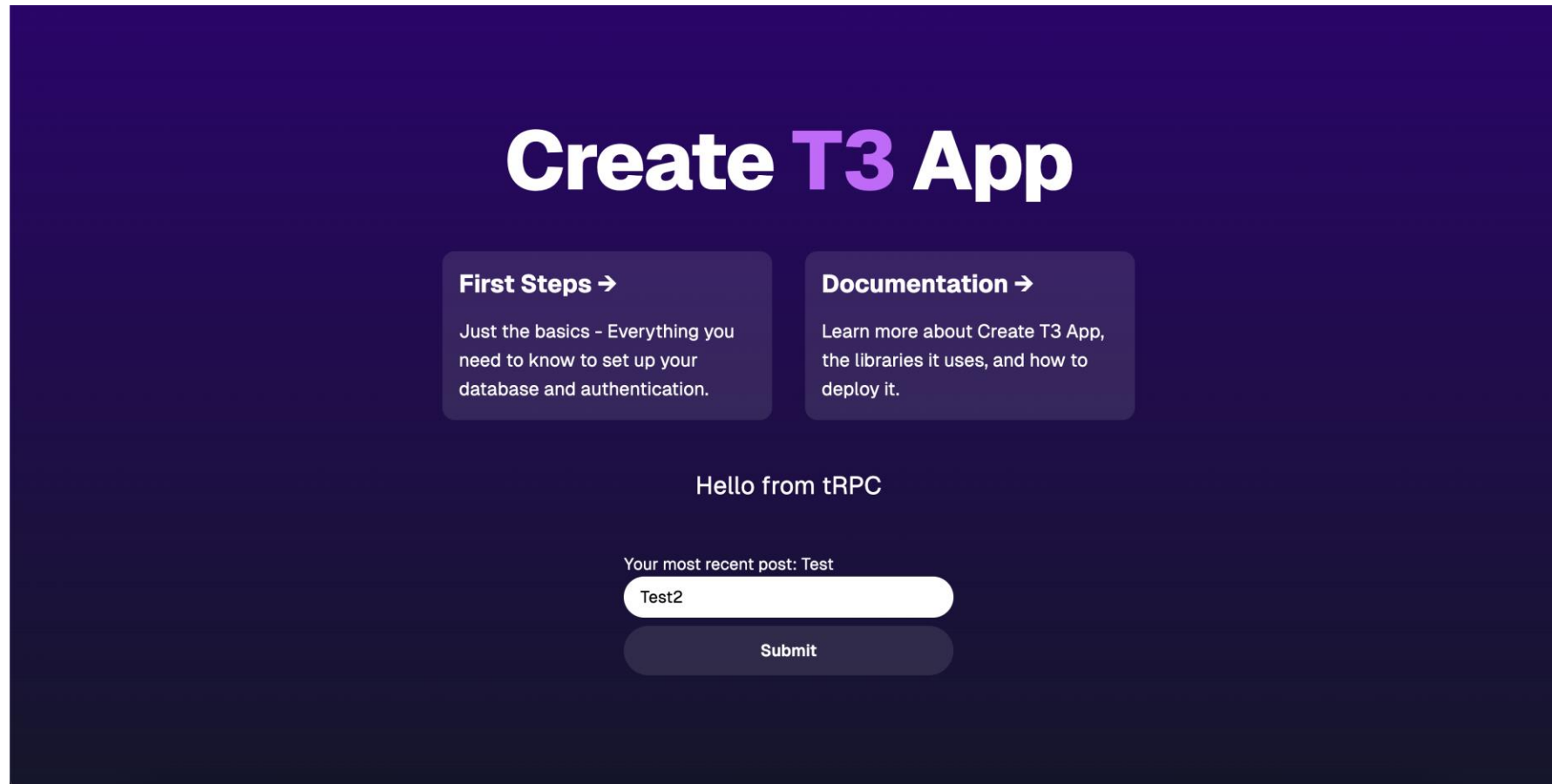
pnpm dev

▲ Next.js 15.0.3 (Turbopack)

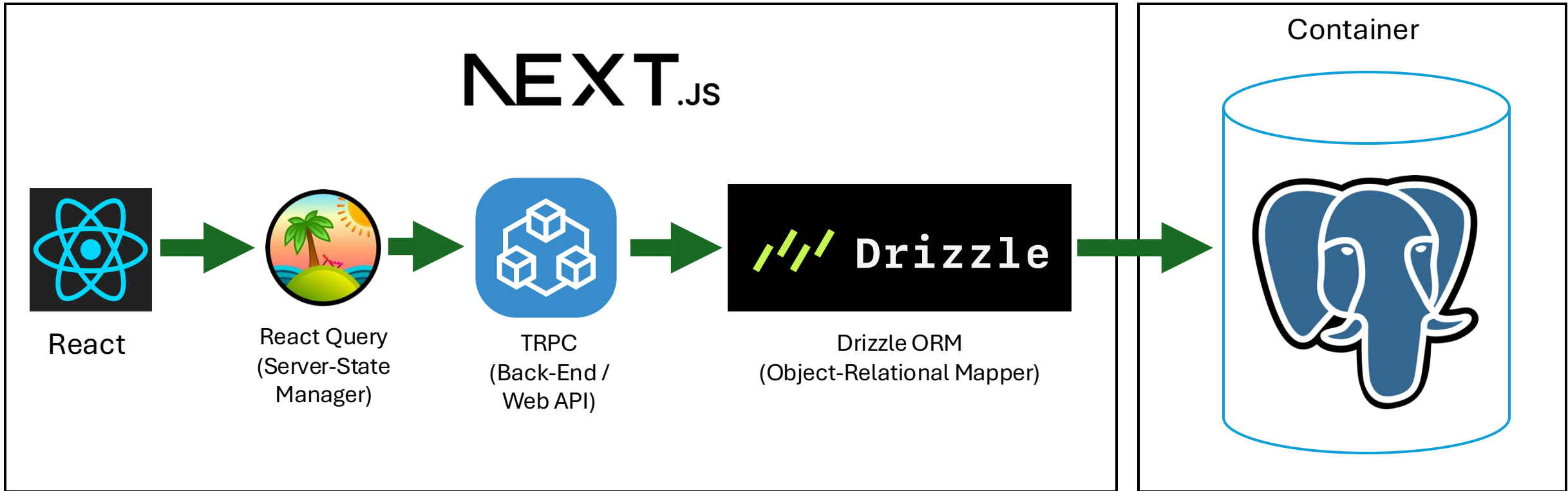
- Local: http://localhost:3000

- Environments: .env

http://localhost:3000



Architecture (Local Dev)

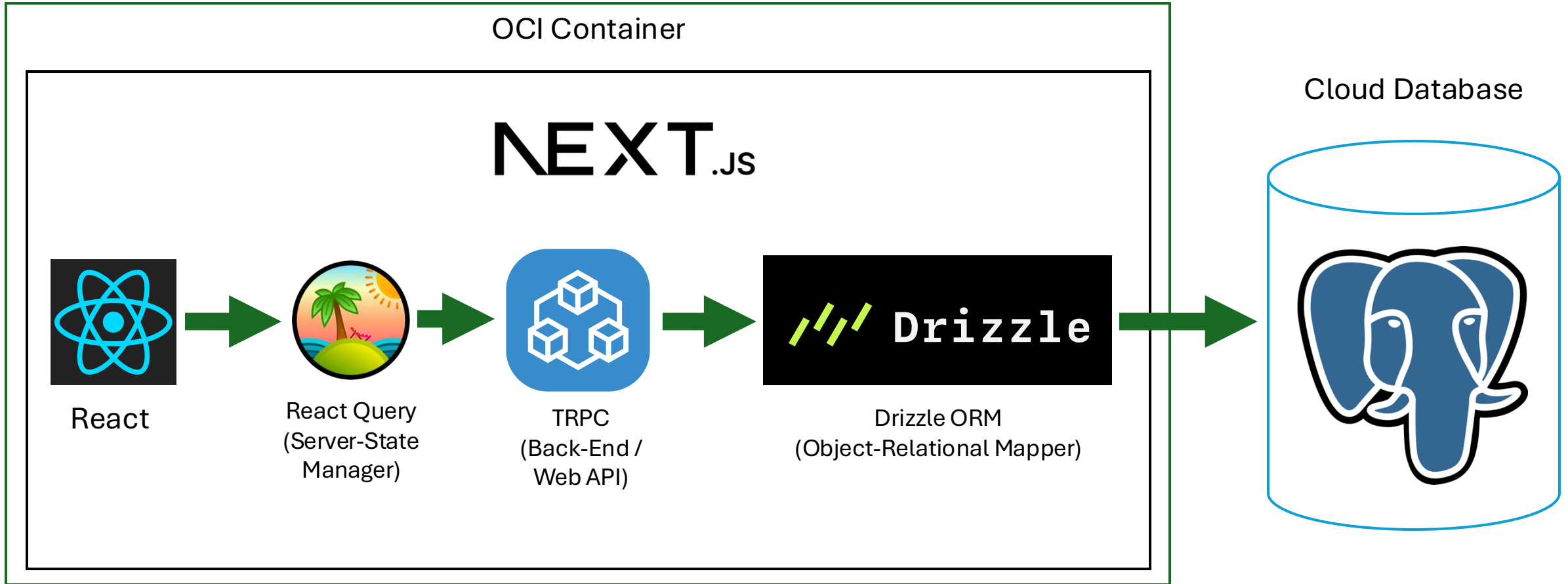


<https://github.com/trpc/trpc>

<https://github.com/TanStack/query>

<https://github.com/drizzle-team/drizzle-orm>

Architecture (Cloud Native)



End-to-End Demonstration

Make a new PostgreSQL table. To-Do List

Develop TRPC API of To-Do List

Develop React Components of To-Do List

Make Next.js OCI Container

Deploy Next.js to Cloud Platforms

src/server/db/schema.ts

```
export const todos = createTable(
  "todo",
  {
    id: text("id").primaryKey(),
    title: text("title").notNull(),
    isCompleted: boolean("is_completed").default(false).notNull(),
    createdAt: timestamp("created_at", { withTimezone: true })
      .default(sql`CURRENT_TIMESTAMP`)
      .notNull(),
    updatedAt: timestamp("updated_at", { withTimezone: true }).$onUpdate(
      () => new Date(),
    ),
  },
);
```

Don't forget to run **pnpm db:push**

src/server/api/routers/todos.ts

src > server > api > routers > TS todos.ts > ...

```
1  import { eq } from "drizzle-orm";
2  import { z } from "zod";
3
4  import { createTRPCRouter, publicProcedure } from "~/server/api/trpc";
5  import { todos } from "~/server/db/schema";
6
7  export const todoRouter = createTRPCRouter({
8    getAll: publicProcedure.query(async ({ ctx }) => {
9      return await ctx.db.query.todos.findMany({
10        orderBy: (todos, { desc }) => [desc(todos.createdAt)],
11      });
12    }),
13
14    create: publicProcedure
15      .input(z.object({ title: z.string().min(1) }))
16      .mutation(async ({ ctx, input }) => {
17        await ctx.db.insert(todos).values({
18          id: crypto.randomUUID(),
19          title: input.title,
20        });
21      }),
22  });
```

React Query

React Query Mutation

```
"use client";  
import { api } from "~/trpc/react";
```

```
function ToDoList() {  
  const { data, isLoading, isLoadingError } = api.todo.getAll.useQuery();  
  
  if (!data || isLoading) {  
    return <div>Loading...</div>;  
  }  
  
  if (isLoadingError) {  
    return <div>Error loading todos</div>;  
  }  
  
  if (data.length === 0) {  
    return <div>No todos</div>;  
  }  
  
  return (  
    <>  
      {data.map((api) => (  
        <div key={api.id}>{api.title}</div>  
      ))}  
    </>  
  );  
}
```

This is React Query!

<https://tanstack.com/query/latest/docs/framework/react/reference/useQuery>

src/app/page.tsx

```
export default function Home() {  
  return (  
    <div className="flex flex-col gap-8 p-16 items-center">  
      <h1 className="text-4xl">To-Do List</h1>  
      <CreateNewForm />  
      <ToDoList />  
    </div>  
  );  
}
```

SQL runner

Drizzle runner

schema: public

Search tables

open-tech-summit-in... 3

open-tech-summit-in... 0

⚙️

📌

🔔

⚠️

📄

<

>

🔄

⚙️ Filters

🔧 Columns

Add record

2 rows • 15ms

<

50

0

>

🔄

📄

<input type="checkbox"/>	id	↕	title	↕	is_completed	↕	created_at	↕	updated_at	↕
<input type="checkbox"/>	192db88d-2ee...		Hello World!		FALSE		2024-11-22 20:01:53.061384+00		NULL	
<input type="checkbox"/>	e1a7489b-720...		Open Tech Su...		FALSE		2024-11-22 20:02:20.872155+00		NULL	

Loading...

<div><div><div></div><div></div></div><div>Elements</div><div>Console</div><div>⚠ Network</div><div>Sources</div><div>Performance</div><div>Memory</div><div>Application</div><div>Security</div><div>Lighthouse</div><div>Recorder</div><div>Sensors</div><div>Components *</div><div>Profiler *</div><div>⚙ ⋮ ✕</div></div>						
<div><div><div>⏹</div><div>🔍</div></div><div><input checked="" type="checkbox"/> Preserve log</div><div><input checked="" type="checkbox"/> Disable cache</div><div>Fast 4G</div><div>⌵</div><div>📶</div><div>⬆</div><div>⬇</div></div> <div><div>⌵ Filter</div><div><input type="checkbox"/> Invert</div><div>More filters ⌵</div><div>All</div><div>Fetch/XHR</div><div>Doc</div><div>CSS</div><div>JS</div><div>Font</div><div>Img</div><div>Media</div><div>Manifest</div><div>WS</div><div>Wasm</div><div>Other</div></div>						
Name	Method	Status	Type	Initiator	Size	Time

Network Throttling: Fast 4G

```
function CreateNewForm() {  
  const [title, setTitle] = useState("");  
  
  const { mutateAsync, isError, error, isPending, reset } =  
    api.todo.create.useMutation();  
  
  const utils = api.useUtils();  
  
  const handleSubmit = async (e: React.FormEvent<HTMLFormElement>) => {  
    e.preventDefault();  
    reset();  
    await mutateAsync({ title });  
  
    setTitle("");  
    reset();  
    utils.todo.getAll.invalidate();  
  };  
};
```

Re-fetch the list

```

return (
  <form onSubmit={handleSubmit} className="flex flex-col max-w-96 gap-1">
    <input
      disabled={isPending}
      className="border border-neutral-600 p-2 py-1 disabled:opacity-60"
      name="title"
      placeholder="Title"
      value={title}
      onChange={(e) => setTitle(e.target.value)}
    />
    {isError && (
      <div className="text-red-600 text-xs">
        {error.data?.zodError?.fieldErrors?.title}
      </div>
    )}
    <button
      disabled={isPending}
      type="submit"
      className="bg-blue-500 rounded text-white py-2 disabled:opacity-60"
    >
      {isPending ? "Saving..." : "Save"}
    </button>
  </form>
);

```

Server-Side Validation!

To-Do List

Save

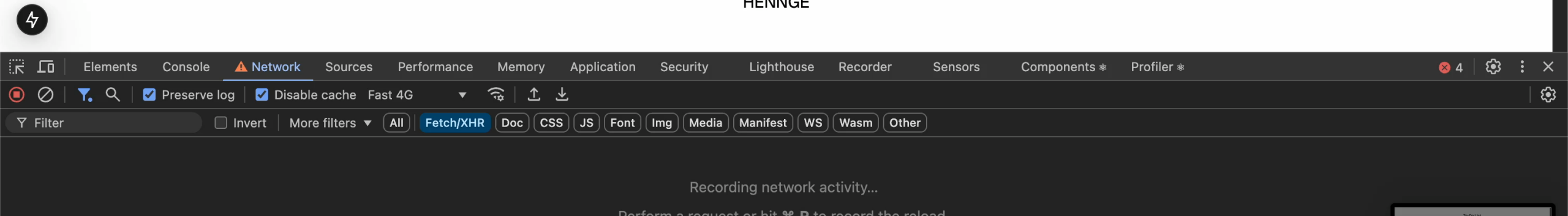
Hello

Hola

Shibuya

Tokyo

HENNGE



Network Throttling: Fast 4G

Recap: What we developed so far

- Defined the PostgreSQL DB schema using TypeScript with Drizzle ORM
- Defined the Back-End / Web API using TypeScript with TRPC
- Managed the to-do list data from the Web API using TypeScript with React Query
- Managed the create to-do form submission to the Web using TypeScript with React Query Mutation
- Developed the presentation layer using TypeScript with React

NEXT.JS ENGINEER

DEV OPS

CLOUD

A yellow shipping container with vertical ridges is positioned on the right side of the frame. The background is a solid blue sky with a small, wispy white cloud on the left. A white rectangular box is overlaid on the left side of the image, containing text.

Let's build some container images

This is the cloud-native part of the talk

BUILDER

```
FROM node:22-slim AS builder
WORKDIR /app
COPY . .
ENV NEXT_TELEMETRY_DISABLED=1
RUN npm install -g pnpm
RUN pnpm install --frozen-lockfile
RUN SKIP_ENV_VALIDATION=1 pnpm run build
```

RUNNER

```
FROM node:22-slim AS runner
WORKDIR /app
ENV NODE_ENV=production
ENV NEXT_TELEMETRY_DISABLED=1

COPY --from=builder /app/next.config.js ./
COPY --from=builder /app/public ./public
COPY --from=builder /app/package.json ./package.json

COPY --from=builder /app/.next/standalone ./
COPY --from=builder /app/.next/static ./next/static

EXPOSE 3000
ENV PORT=3000
```

```
CMD ["server.js"]
```

Dockerfile

- `docker build -t opentechsummit2024 .`
- Test the container against the PostgreSQL database instance we used for development:
- `docker run -p 3000:3000 -e DATABASE_URL="postgresql://postgres:password@host.docker.internal:5432/open-tech-summit-indonesia-2024" opentechsummit2024`

pnpm db:generate

open-tech-summit-indonesia-2024_todo 5 columns 0 indexes 0 fks

[✓] Your SQL migration file → drizzle/0000_reflective_cargill.sql 

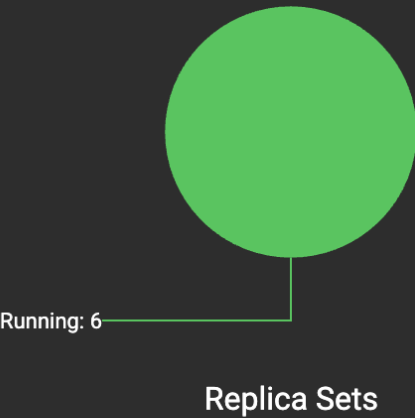
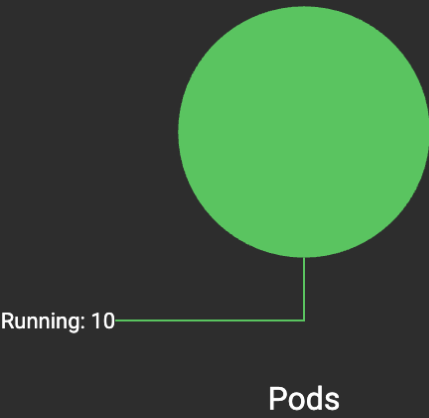
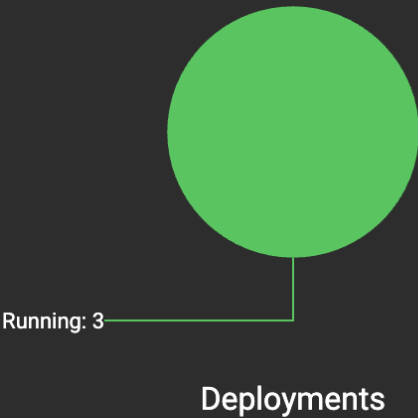
```
CREATE TABLE IF NOT EXISTS "open-tech-summit-indonesia-2024_todo" (  
  "id" text PRIMARY KEY NOT NULL,  
  "title" text NOT NULL,  
  "is_completed" boolean DEFAULT false NOT NULL,  
  "created_at" timestamp with time zone DEFAULT CURRENT_TIMESTAMP NOT NULL,  
  "updated_at" timestamp with time zone  
);
```

Kubernetes deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: opentechsummit2024
spec:
  replicas: 8
  selector:
    matchLabels:
      app: opentechsummit2024
  template:
    metadata:
      labels:
        app: opentechsummit2024
    spec:
      containers:
        - name: opentechsummit2024
          image: opentechsummit2024:latest
          imagePullPolicy: Never
          ports:
            - containerPort: 8080
          env:
            - name: DATABASE_URL
              value: "postgresql://postgres:password@postgresql:5432/open-tech-summit-indonesia-2024"
```

```
apiVersion: v1
kind: Service
metadata:
  name: opentechsummit2024
spec:
  selector:
    app: opentechsummit2024
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
```

Workload Status

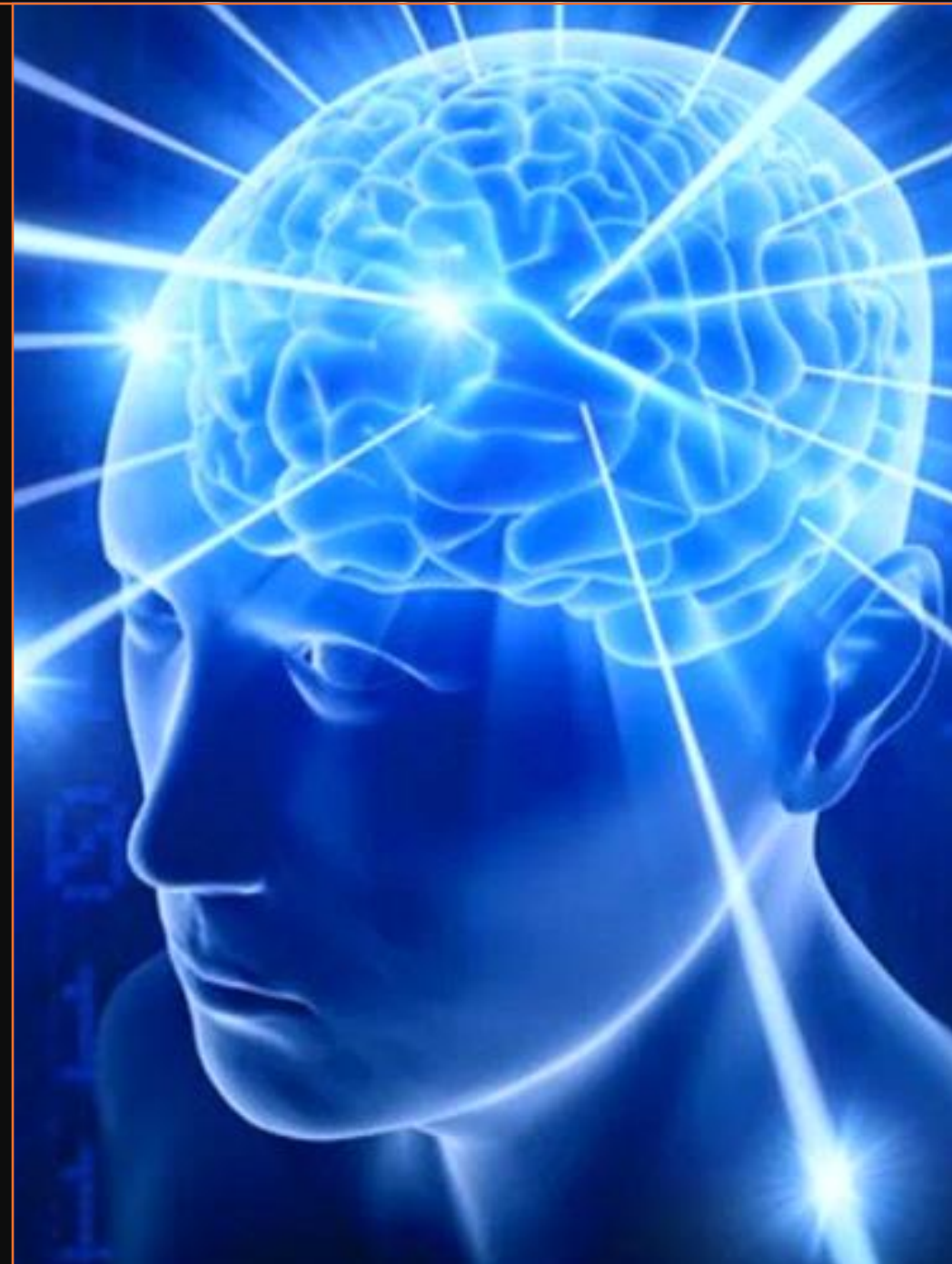


Deployments

Name	Images	Labels	Pods	Created ↑	
● opentechsummit2024	opentechsummit2024:latest	-	8 / 8	21 minutes ago	⋮
● pgadmin	dpage/pgadmin4	-	1 / 1	21 minutes ago	⋮
● postgresql	postgres	-	1 / 1	21 minutes ago	⋮

What if...

Instead of **YAML**, you can
also use **TypeScript**? 😊




```

import * as k8s from "@pulumi/kubernetes";

const openTechSummitDeployment = new k8s.apps.v1.Deployment(
  "opentechsummit2024",
  {
    metadata: { name: "opentechsummit2024" },
    spec: {
      replicas: 8,
      selector: { matchLabels: { app: "opentechsummit2024" } },
      template: {
        metadata: { labels: { app: "opentechsummit2024" } },
        spec: {
          containers: [
            {
              name: "opentechsummit2024",
              image: "opentechsummit2024:latest",
              imagePullPolicy: "Never",
              ports: [{ containerPort: 8080 }],
              env: [
                {
                  name: "DATABASE_URL",
                  value:
                    "postgresql://postgres:password@postgresql:5432/open-tech-summit-indonesia-2024",
                },
              ],
            },
          ],
        },
      },
    },
  },
);

```

Run: **pulumi up**



Pulumi

<https://github.com/pulumi/pulumi-kubernetes>

<https://github.com/pulumi/pulumi>

```

const openTechSummitService = new k8s.core.v1.Service("opentechsummit2024", {
  metadata: { name: "opentechsummit2024" },
  spec: {
    selector: { app: "opentechsummit2024" },
    ports: [{ protocol: "TCP", port: 3000, targetPort: 3000 }],
  },
});

```

```

import * as pulumi from "@pulumi/pulumi";
import * as aws from "@pulumi/aws";
import * as awsx from "@pulumi/awsx";

const cluster = new aws.ecs.Cluster("cluster", {});
const loadbalancer = new awsx.lb.ApplicationLoadBalancer("loadbalancer", {});
const repo = new awsx.ecr.Repository("repo");
const image = new awsx.ecr.Image("image", {
    repositoryUrl: repo.url,
    imageName: "opentechsummit2024",
    context: "../..",
    platform: "linux/amd64",
});
const service = new awsx.ecs.FargateService("service", {
    cluster: cluster.arn,
    assignPublicIp: true,
    taskDefinitionArgs: {
        container: {
            name: "opentechsummit2024",
            image: image.imageUri,
            cpu: 128,
            memory: 512,
            essential: true,
            portMappings: [
                {
                    containerPort: 3000,
                    targetGroup: loadbalancer.defaultTargetGroup,
                }
            ],
        }
    },
});

export const frontendURL = pulumi.interpolate`http://${loadbalancer.loadBalancer.dnsName}`;

```



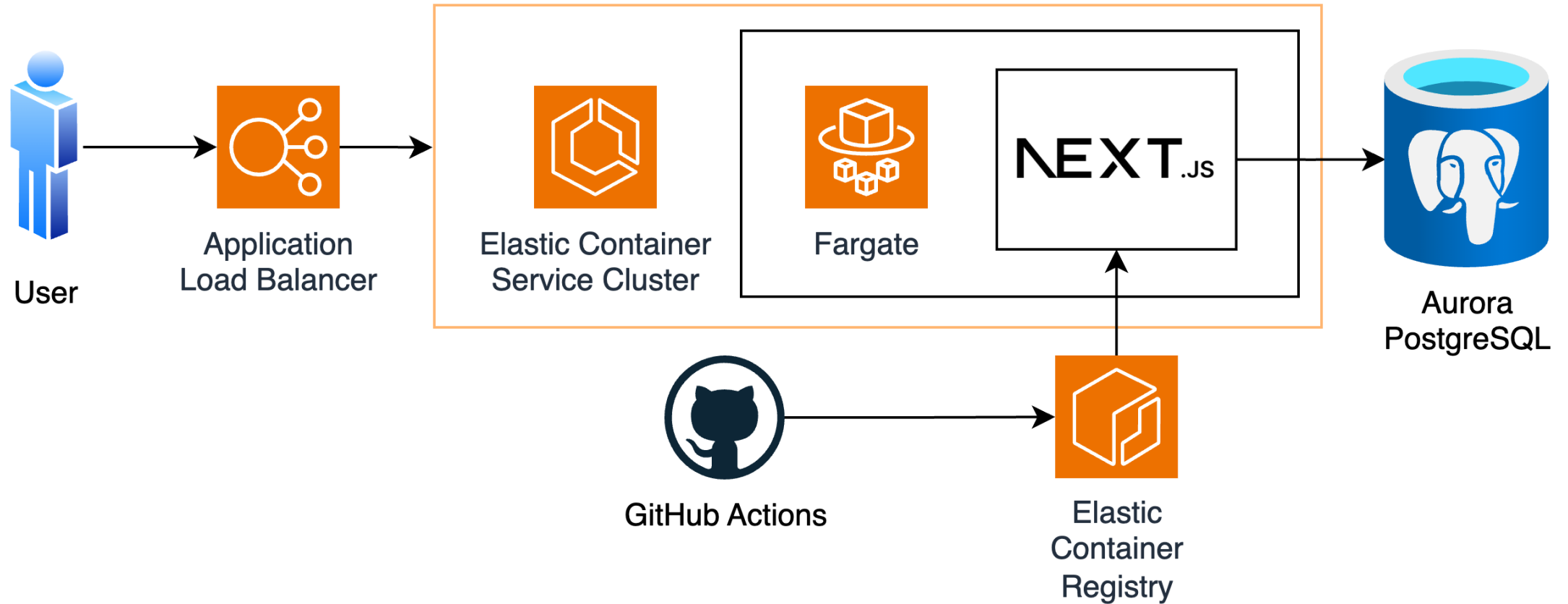
Pulumi Crosswalk for AWS

<https://github.com/pulumi/pulumi-awsx>

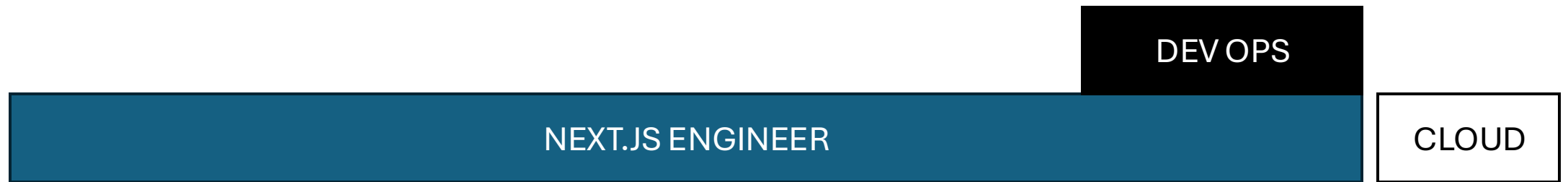
<https://github.com/pulumi/pulumi-aws>

<https://www.pulumi.com/registry/packages/awsx/installation-configuration/>

Architecture on AWS



Full-Stack Revolution



Presentation

Business Logic

Infrastructure



From Front-End to Back-End to Dev Ops
One Language, One Team, One Stack

Takeaways

- Start with one team and one vision to achieve product delivery
- Iterate quickly with one programming language and one technology stack for less development overhead
- Next.js and TypeScript can cover most common use cases
- Over time, add specialized microservices as needed around Next.js to accommodate business needs.
 - Go / .NET worker service for processing RabbitMQ or Kafka streams
 - Python FastAPI for running AI models Web API
 - Real-Time service using Node.js Socket.io



Thank You for Listening!

Terima Kasih!

ありがとうございます!

[https://github.com/ryanelian/
open-tech-summit-indonesia-2024](https://github.com/ryanelian/open-tech-summit-indonesia-2024)

Tech Internship in Shibuya, Tokyo:

<https://recruit.hennge.com/en/gip/>

Join Us!

[https://recruit.hennge.com/en/
mid-career-ngh/](https://recruit.hennge.com/en/mid-career-ngh/)