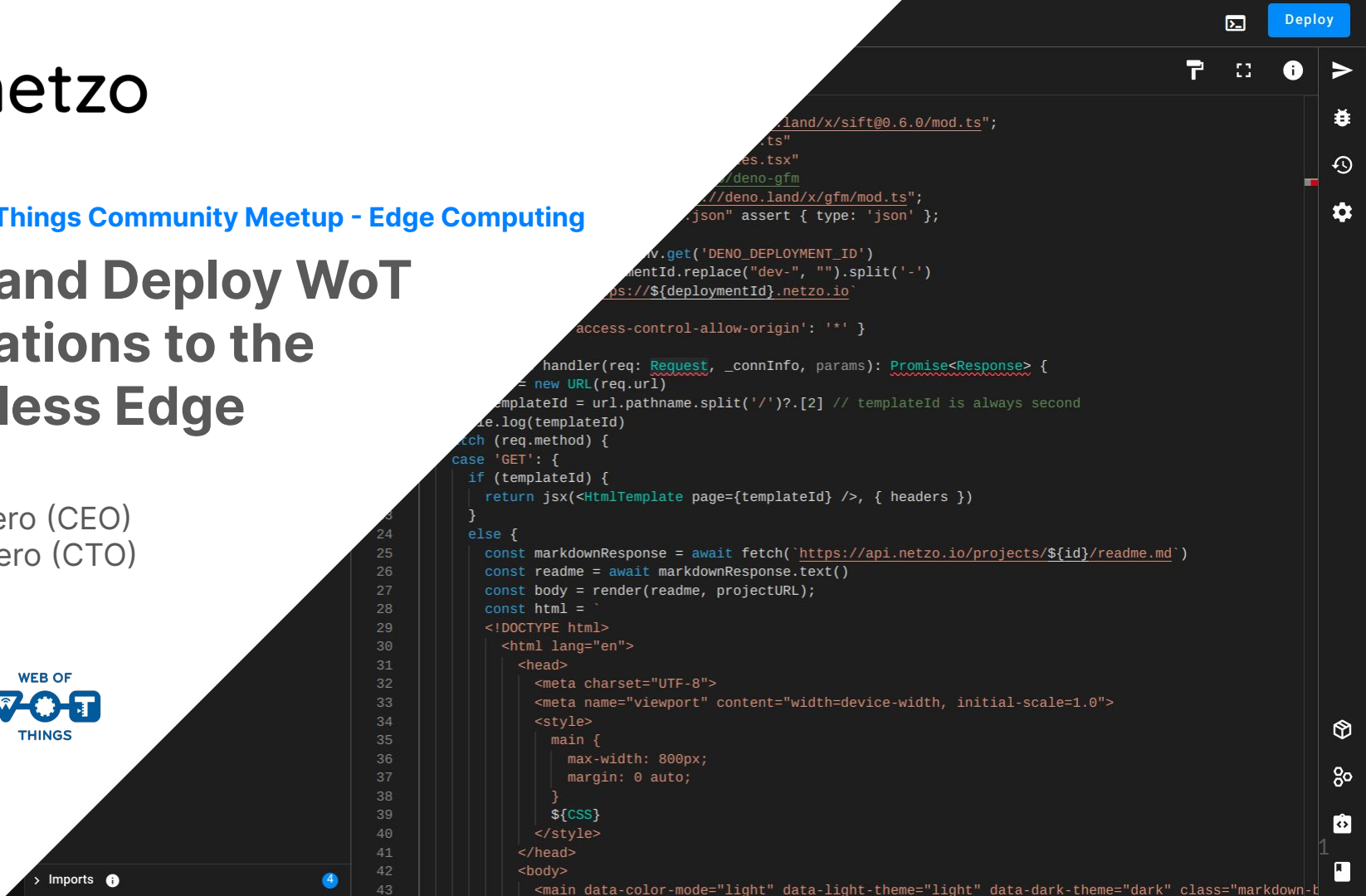
W3C Web of Things Community Meetup - Edge Computing

Script and Deploy WoT Applications to the Serverless Edge

Arturo Romero (CEO)
Miguel Romero (CTO)

15.12.2022



Netzo: The scripting platform for developers

Deploy JavaScript and TypeScript to URL endpoints instantly. Connect essential APIs, automate processes and build tools faster, without managing infrastructure.

Production deployments on a click

Instantly turn scripts into custom workflows and UIs at URL Endpoints.

Integrate anything with an API

Integrate business apps directly or from the open-source Marketplace.

Run anything on a schedule

Configure jobs and workflows to run at specific times and intervals.

- Webhook Handling
- Request Proxying
- Data Processing
- Admin Panels
- Form Handling
- Server-side Rendering (with JSX)
- KPI Dashboards
- Request Scheduling (Cron Jobs)
- Workflow Automation

and more...

Instantly turn scripts into automations and UIs

Integrate anything with an API

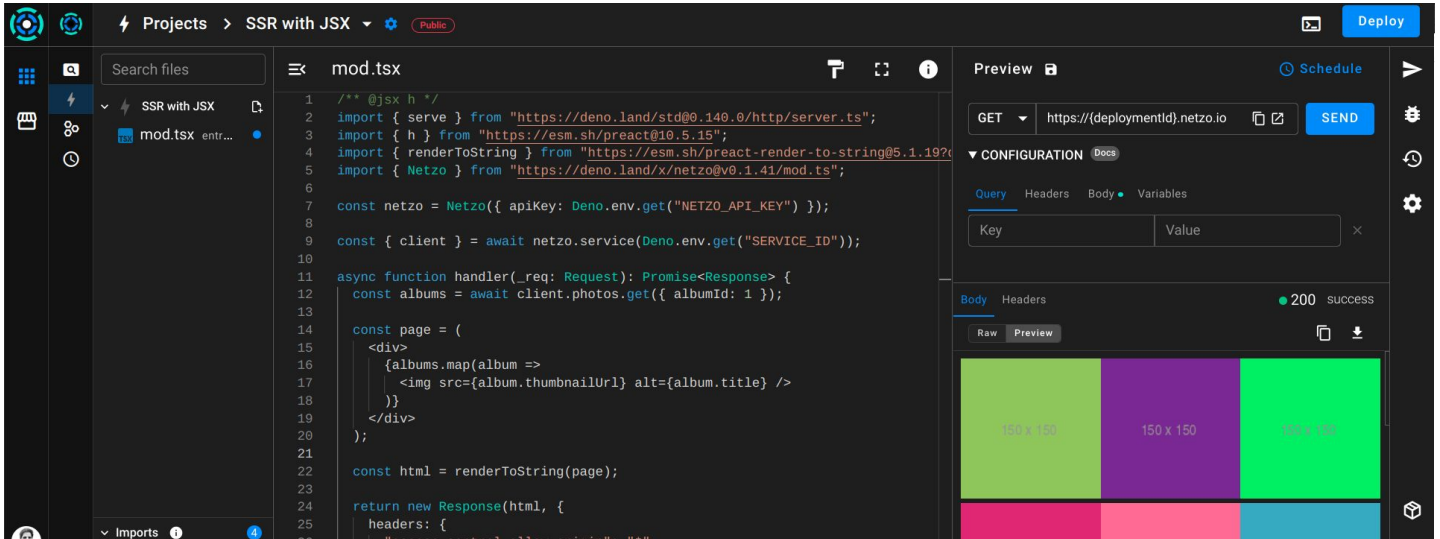
Code fast in the browser

Automate work and get more done

Deploy instantly, skip the Devops

Run anything on a schedule

Empower developers to innovate



Boost productivity across organizations

Sales and Marketing

Close more sales, faster by optimizing lead generation, prospect management and revenue generation

Production

Capture production process and inventory data to improve quality and supply chain controls

Operations

Keep data in sync across all teams and build portals and interfaces that enhance internal business operations

Internet of Things

Enhance process data, trigger workflows and augment operations through IoT device interactions

Accounting and Finance

Load and sync electronic invoice data with multiple systems to enhance reporting, file naming and storage.

IT

Increase cybersecurity and minimize disruptions through isolated functions and identify incidents immediately.

and more...

Workers: Custom HTTP endpoints for any use case

Click Here

URL imports

Env Variables

Native TS

TSX/JSX

Web Standards

```tsx
/** @jsx h */
import { serve } from "https://deno.land/std@0.140.0/http/server.ts";
import { h } from "https://esm.sh/preact@10.5.15";
import { renderToString } from "https://esm.sh/preact-render-to-string@5.1.19?deps=preact@10.5.15";
import { Netzo } from "https://deno.land/x/netzo@v0.1.41/mod.ts";

const netzo = Netzo({ apiKey: Deno.env.get("NETZO_API_KEY") });

const { client } = await netzo.service(Deno.env.get("SERVICE_ID"));

async function handler(_req: Request): Promise<Response> {
  const photos = await client.photos.get({ albumId: 1 });

  const page = (
    <div>
      {photos.map(photo =>
        <img src={photo.thumbnailUrl} alt={photo.title} />
      )}
    </div>
  );

  const html = renderToString(page);

  return new Response(html, {
    headers: {
      "access-control-allow-origin": "*",
      "content-type": "text/html"
    },
  });
};

serve(handler);
```

Netzo SDK

API Integrations

SSR (server- side rendering)

HTTP Server to handle Requests

netzo

Runs locally as well: `deno run -A https://api.netzo.io/projects/:id/mod.tsx`

Scripting WoT applications at worker runtimes (serverless edge)

Worker Runtimes are the new standard for writing HTTP servers in JavaScript.

Netzo streamlines scripting and deployment of Web of Things applications, enabling users to bring ideas to production faster. For the Web of Things, scripts can be used to...

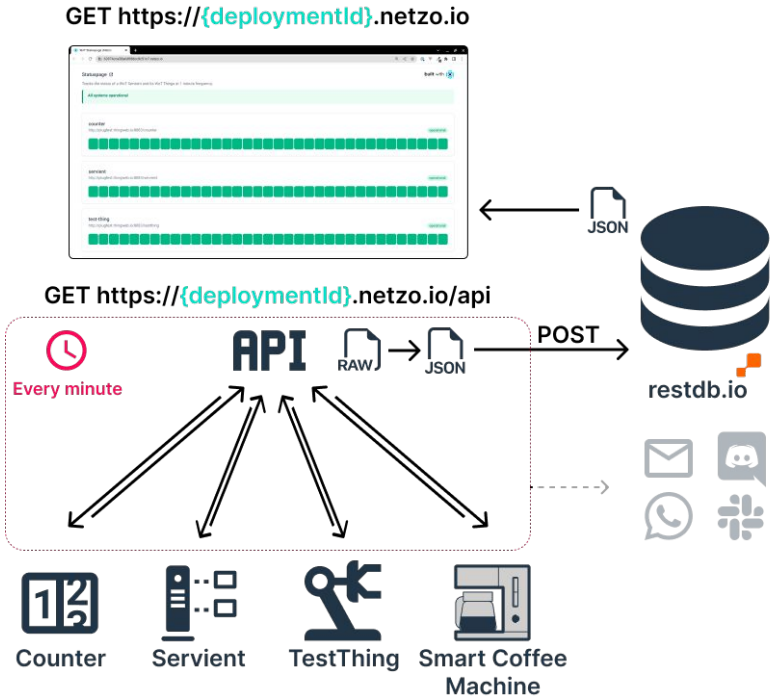
- **augment** existing WoT deployments
- **integrate** WoT entities with cloud services
- **proxy** WoT entities (act as HTTP Gateway)
- **process** data (aggregated, transform, etc.)
- **respond** to events (e.g. webhooks)
- **automate** interactions of WoT Things
- **schedule** recurring jobs

Demo 01: WoT statuspage for multiple WoT systems

Click Here

Goal: Provide monitoring capabilities to existing WoT Things and automate status reporting (e.g. notifications, email)

- `GET /` → serves UI
- `GET /api` → serves JSON data after fetching, aggregating and storing real-time state to DB
- **Process:** aggregate real-time state
- **Store:** state of WoT systems to DB
- **Alert:** when systems are down

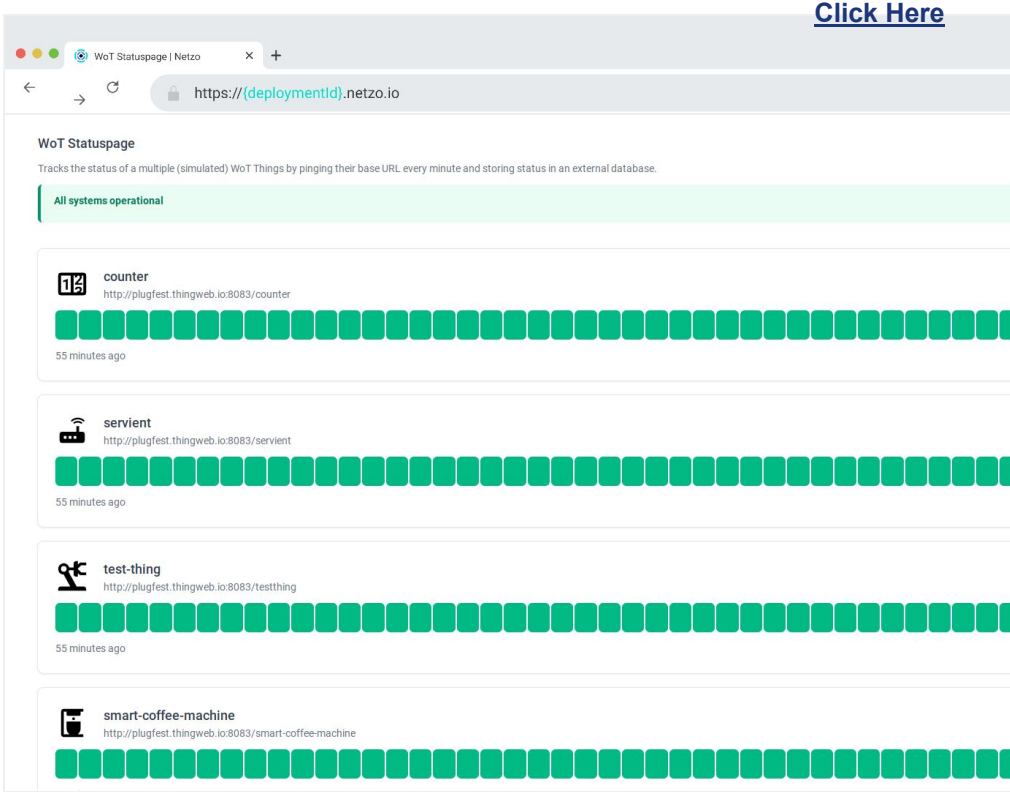


Demo 01: WoT statuspage for multiple WoT systems

Click Here

This demo showcases:

- **SSR** (server-side rendered) UI built with TSX (no build step)
- **Schedules**: polls the real-time state and persists it to an external database
- **API Integrations**: GETs real-time state and POSTs it to external database
- **Database:** persist data to external DB
- **Routes**: separate handlers for routes

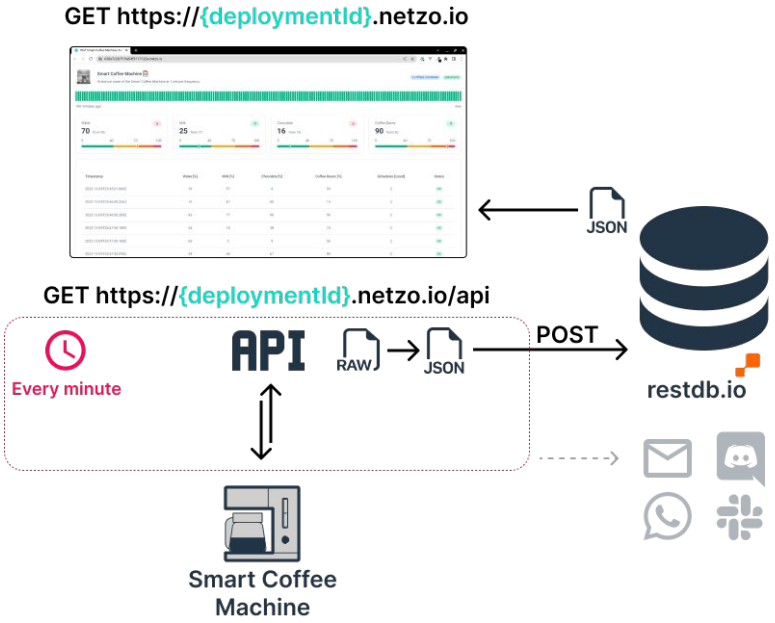


Demo 02: WoT dashboard for monitoring and control

Click Here

Goal: Persists historical state (time-series data) of a WoT Thing, augment it, provide a dashboard for monitoring and control and add reporting (e.g. via notifications, email)

- `GET /` → serves UI
- **Collect:** data from multiple sources
- **Process:** aggregate real-time stats
- **Visualize:** render information and insights in a actionable dashboard
- **Report:** notify to external services

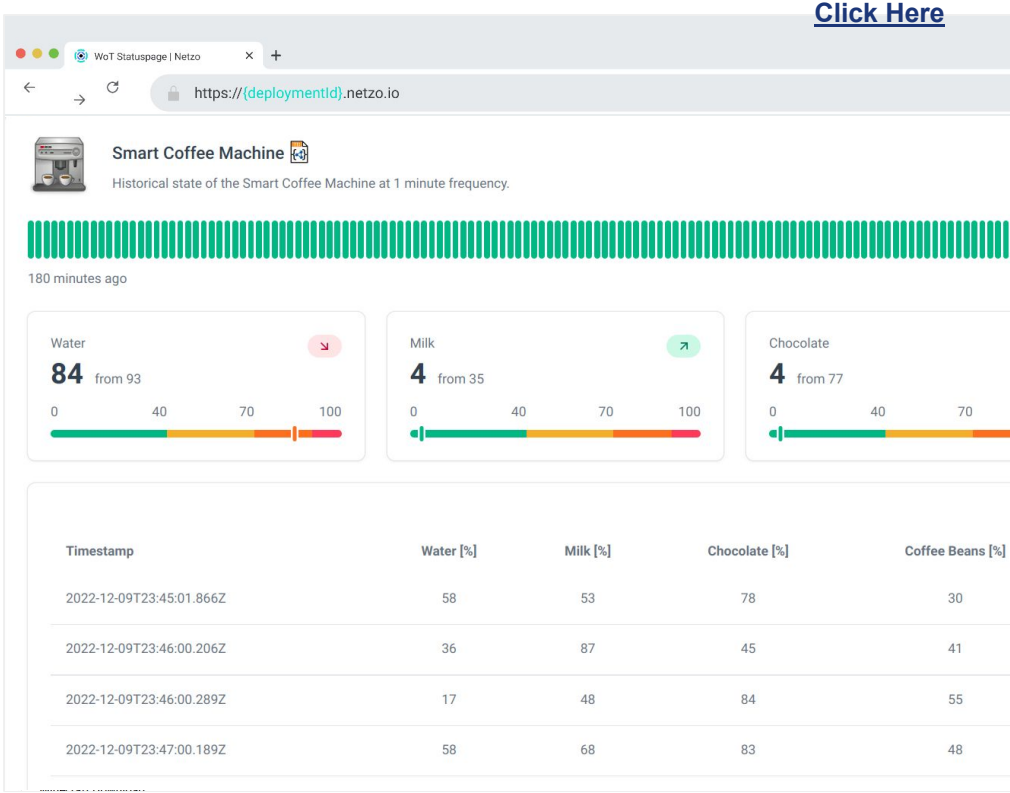


Demo 02: WoT dashboard for monitoring and control

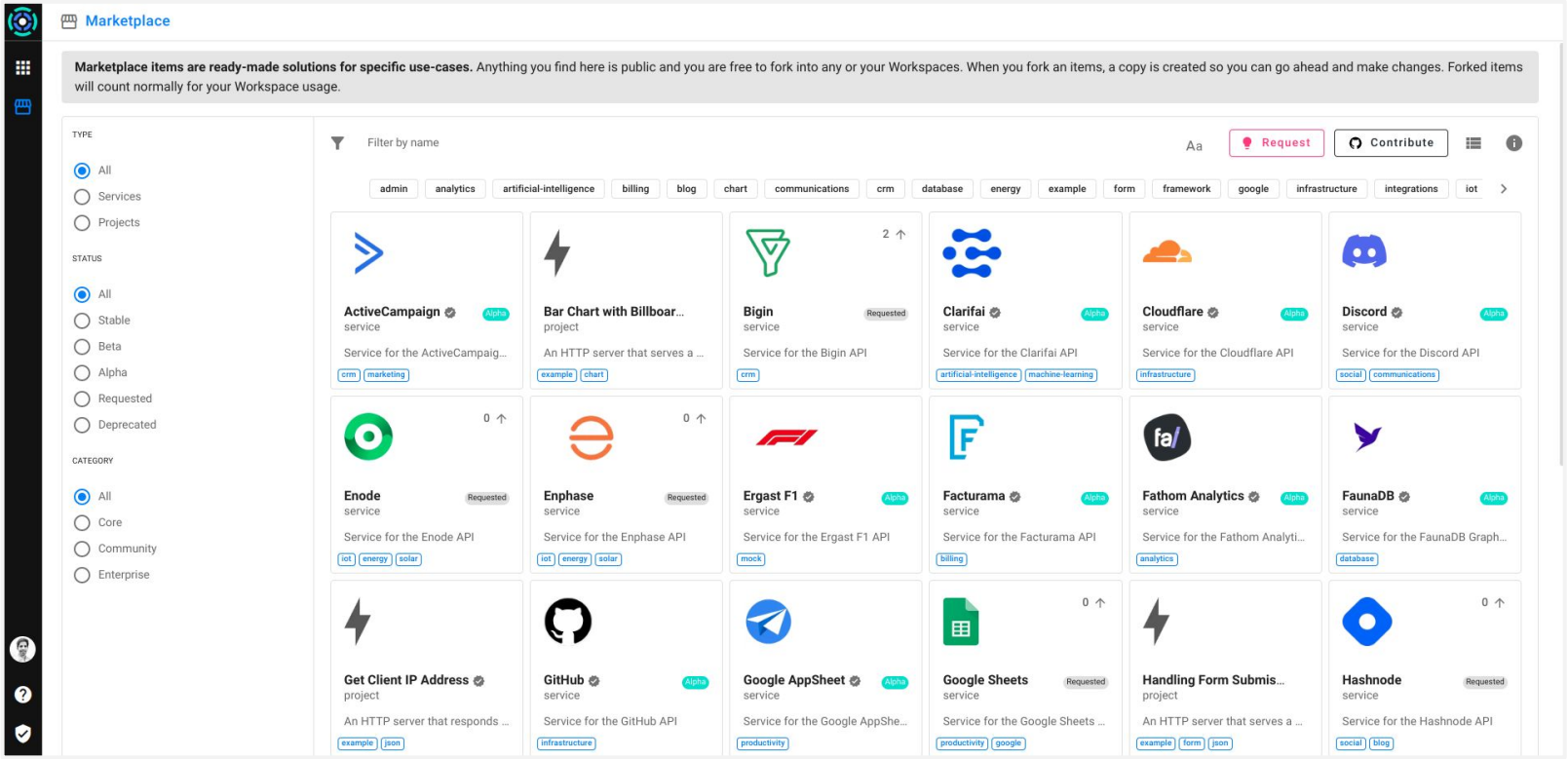
Click Here

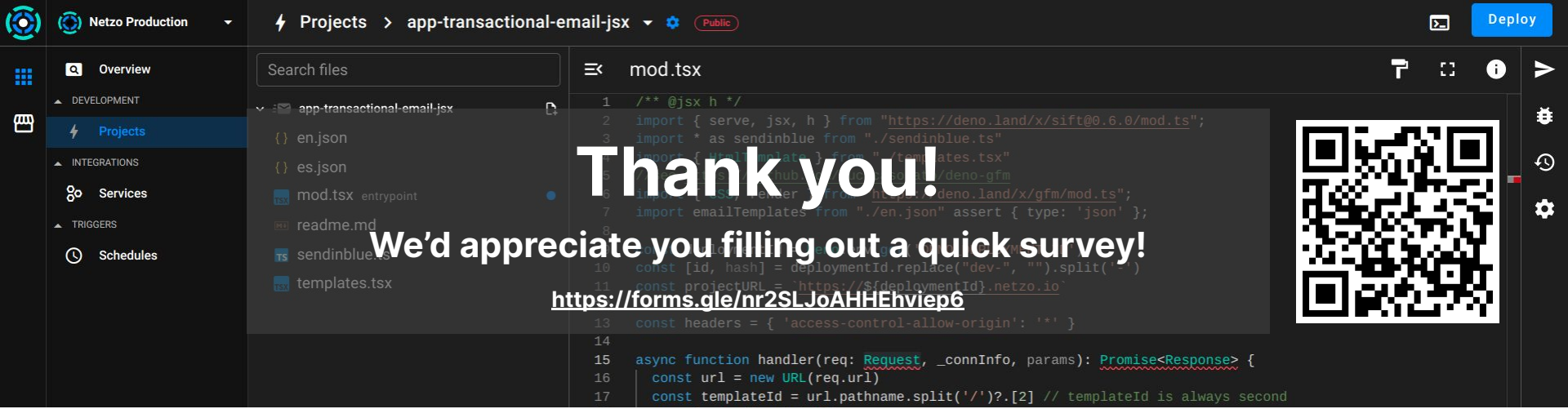
This demo showcases:

- **Environment Variables:** provides separate dev/prod environments
- **Static file hosting**: serves WoT Thing Description (JSON) at a Web URL
- **SSR** (server-side rendered) UI built with TSX (no build step)
- **Schedules:** polls the real-time state and persists it to an external database
- **API Integration:** GETs real-time state and POSTs it to external database
- **Database:** persist data to external DB



Marketplace of ready-made solutions





ROKAWARE SL

P. de la Castellana 89, Planta 8ª
28046 Madrid, España
netzo.io | hello@netzo.io | +34 910 601 536
Netzo © ROKAWARE SL

The information provided in this document has been obtained from sources we believe to be reliable. However, we cannot guarantee the accuracy or completeness of this information and assume no liability for it. Copies of the contents of this presentation, in particular videos, screenshots, printouts, copies or publications on electronic media, shall only be authorized with the written consent of ROKAWARE SL.