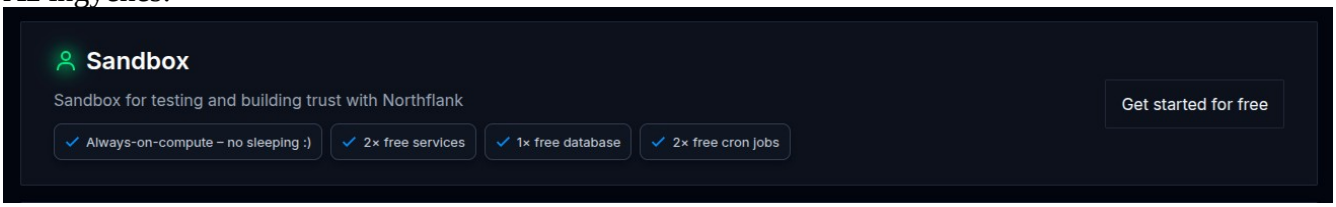
A projektet saját Githubon hoztam létre publikus repoban, külön brancheken. Funkcionalitásban azonosak. A Northflank és Cloudflare fiókok szintén saját adatokkal jöttek létre.

Northflank

Link: https://p01--frontend--pxyfv22r4grv.code.run/

Az ingyenes:



Regisztráció

Egyszerű folyamat. Bankkártya megadása szükséges, de ez csak akkor derült ki mikor már bekonfiguráltam az adatbázist és csak rá kellett nyomni a “Létrehozás”ra

Projekt létrehozása & Első deploy

Nem volt nehéz:
a backend és frontend 1-1 külön service, az adatbázis egy 'addon', ezzel ki is van maxolva az ingyenes rész a 2 cron job-on kívül. Tartós fájlrendszer nincs ingyen, ezért a “kiskutya” a backend konténer része, backend/data/message.txt-ből jön. Github repo és branch megadás, mindenhez saját beállítás, pl network, build/runtime variables stb. A legtöbb helyre jó volt az alapértelmezett.

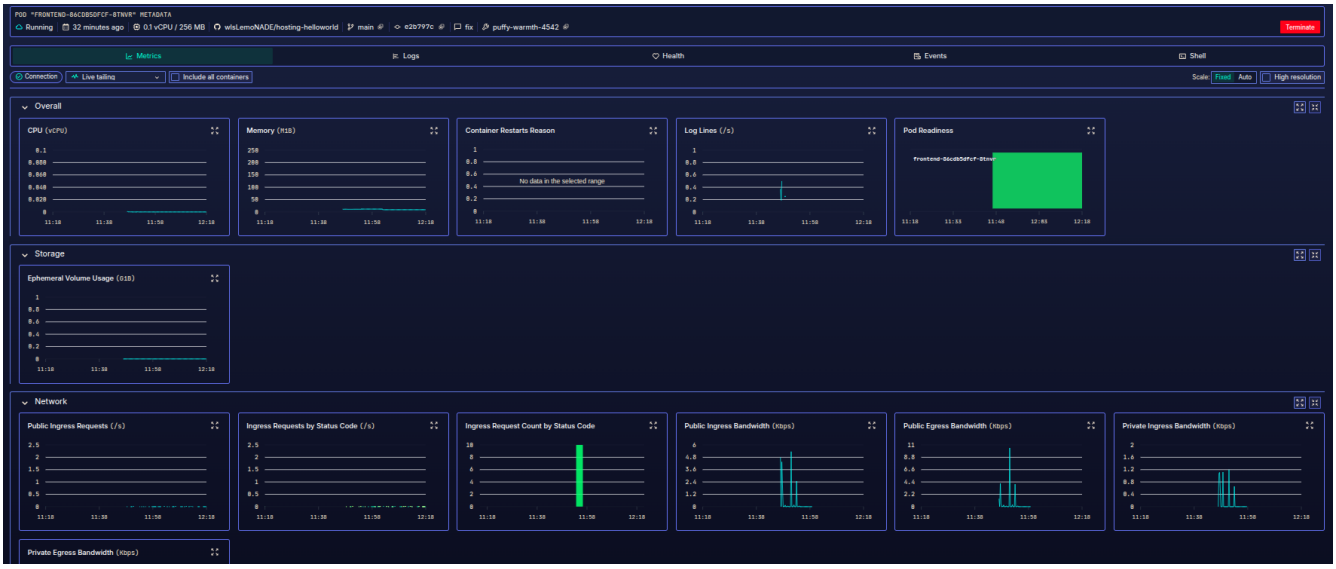
További deploy

Auto CI/CD-hez össze kell kötni a Northflanket és a git szolgáltatót, Gitlab is támogatva van, de azzal nem próbáltam. A Githubos megoldás simán ment.

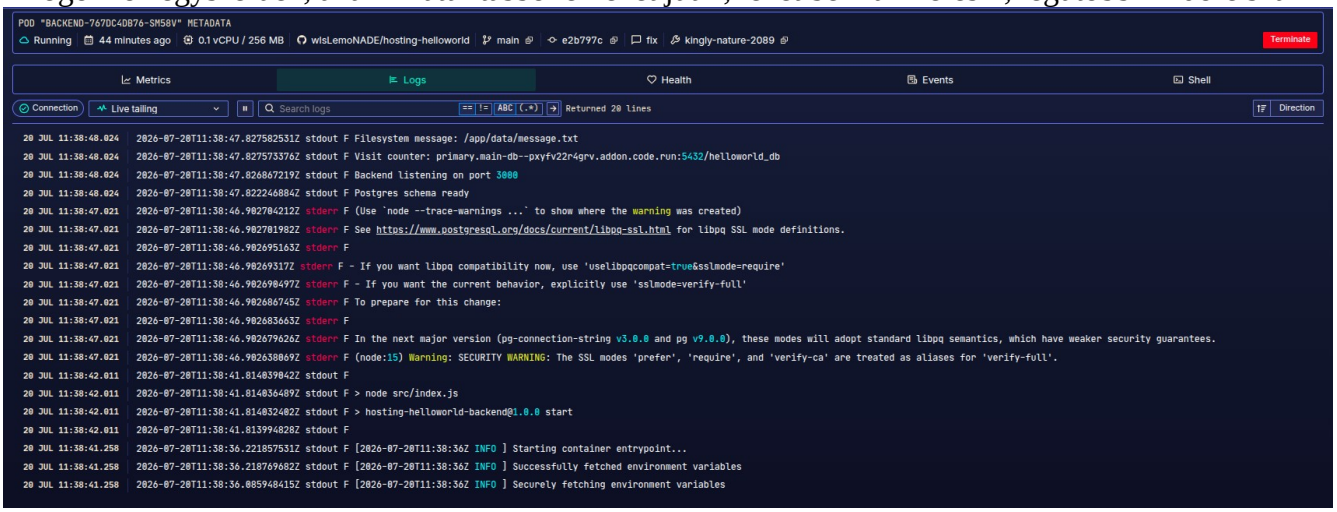
Az auto CI/CD könnyen kikapcsolható service-nként

Nyomon követés

Egyszerű felületen viszonylag sok metrika látható működés közben



A logokhoz egyszerűen, akár 2 kattintásból el lehet jutni, lehet bennük keresni, legutóbbi x időre szűrni



Van shell hozzáférés a fájlokhoz, kis negatívum hogyha átkattintunk pl a Logs-ra majd vissza akkor az ssh ablak reset-el

Vélemény

A felület egyszerűnek bizonyult, mindent megtaláltam benne amire (szerintem) szükség lehet. Nem okozott gondot a navigáció annak ellenére, hogy minden oldalon sok információ és lehetőség van egyszerre. Az alap domain nem túl bizalomgerjesztő.

Cloudflare

Link: https://hosting-helloworld.pages.dev/

Regisztráció

Egyszerű email+jelszó pár, bankkártya nem kellett.

Projekt létrehozása & Első deploy

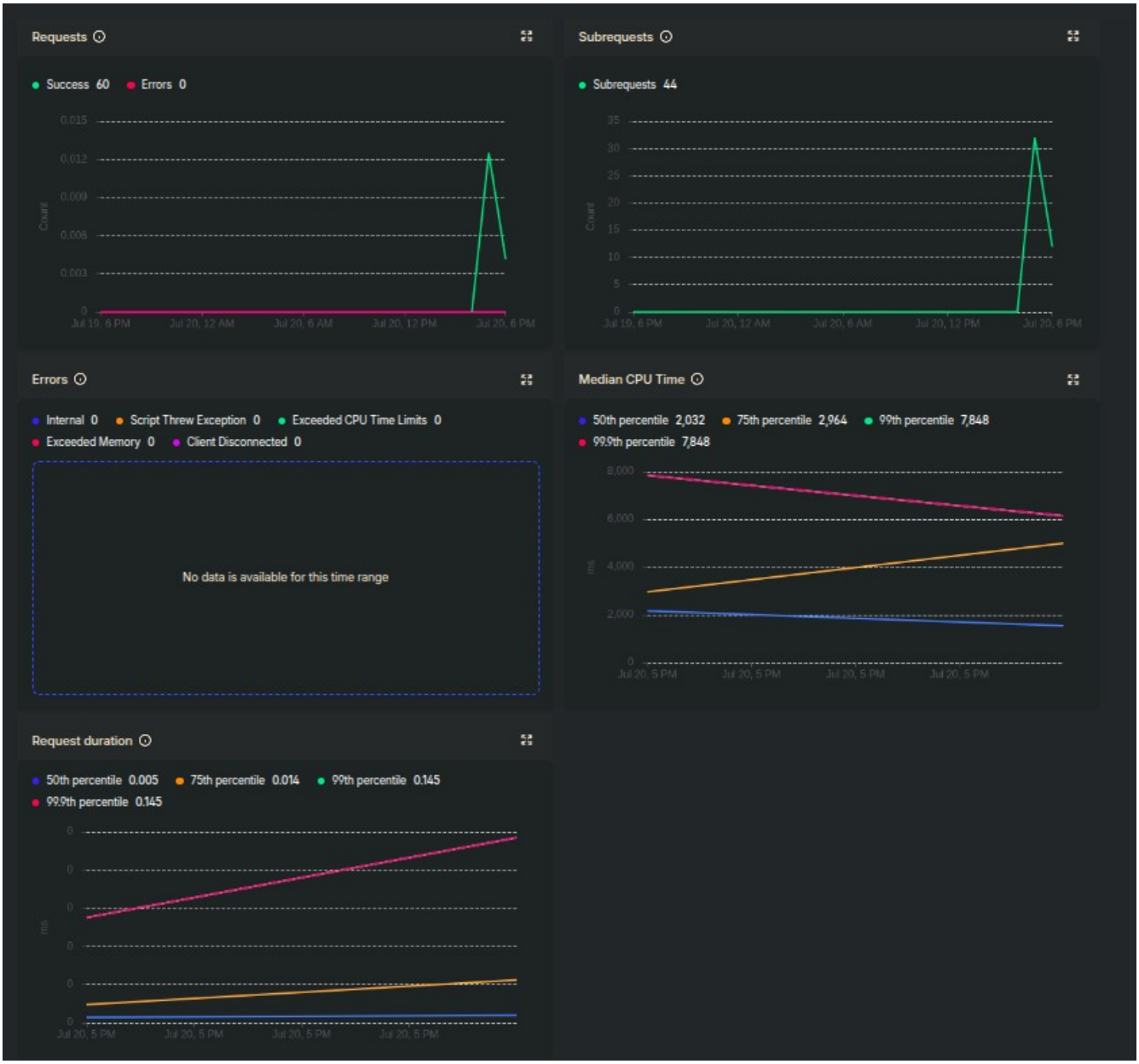
Kicsit nehezebb volt eligazodni azon, hogy mit is keresek, de összességében jóval gyorsabb volt mint a Northflank, bár ez inkább annak köszönhető, hogy a 80%-át megtudta csinálni a cursor. A projekt azonos repoban külön branchen módosításra került, mivel bizonyos részei máshogy működnek cloudflaren, vagy nem lett volna ingyenes. Nincs külön frontend/backend hanem egy közös Pages project van A kiskutya itt is projekten belüli fájlból jön, de ha minden igaz R2 Object Storage vagy worker Key-Value val is meg lehetne oldani ingyen.

További deploy

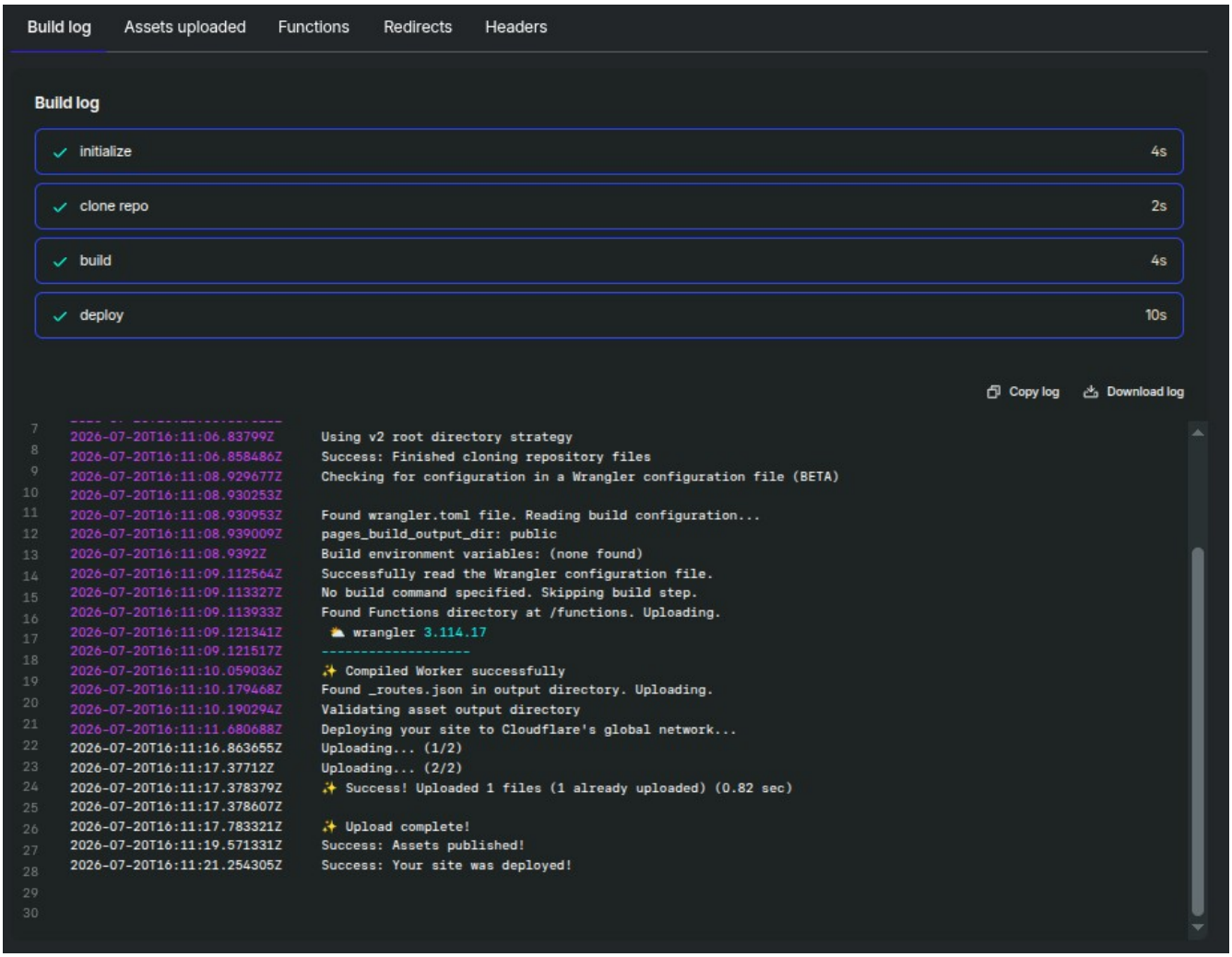
Auto CI/CD-hez össze kell kötni a Cloudflaret és a git szolgáltatót, Gitlab is támogatva van, de azzal nem próbáltam. A Githubos megoldás simán ment. Szintén kikapcsolható az auto CI/CD, de sokkal több kattintásba kerül.

Nyomon követés

Itt is vannak metrikák, bár sokkal kevesebb, projekt és account szinten is.



A logokhoz elég nehéz eljutni hacsak nincs bent a legutóbbi elemek között, illetve nem találtam hozzá fukciókat mint pl keresés/szűrés. Vannak extra log managelő részek, de azok különálló modulok és pénzért.



Nincs ssh access, nincs meg minden fájl egy helyen áttekintés szinten sem.

Vélemény

Bár jóval hamarabb “beindult” sokkal kevésbé értettem meg, hogy hogyan is tette és hogyan működik, nagyon szokatlan és eltérő az eddigiekhez képest, főleg a docker hiánya és a helyette lévő más felépítés miatt (Functions, public/, etc.) A felület nekem nehezebben kezelhető, kevésbé intuitív és jobban keresgélős, amin beépített ai chattel próbálnak segíteni. Széleskörű szolgáltatások, de csak külön modulokként, amiknek vagy van ingyen része vagy nincs.

Működtetésbeli különbségek

	Northflank	Cloudflare Pages + Functions
Process model	Always-on containers	On-demand isolates (wake per request, de <1s ahogy néztem)
How many “services”	Separate frontend + backend (+ DB)	One project; routes decide static vs Function
Networking	Private DNS (`backend:3000`), proxy in nginx	Same origin; `/api/*` is just Functions
Runtime	Full Node + Linux filesystem	Workers runtime (limited Node APIs)
Scaling	Scale containers	Cloudflare scales Functions automatically
Cost shape	Pay for running services/DB even idle	Free tier: pay mainly for requests/DB ops

---

Teljesítménybeli különbségeket nem tudtam tesztelni.

Előnyök, hátányok

Northflank	
Pros	**Cons**
Megszokott model/működtetés	A több különálló rész baj is lehet
Egyértelmű elválasztás komponensek között	Ha fizetésre kerül a sor az idle időért is fizetni kell (Nem tudom, hogy többe kerülne-e, mint a cf)
A backend nem kell hogy publikusan elérhető legyen	Kötelező bankkártya
Hamar megszokható kezelhető felület	Az alap domain úgy néz ki mintha a májamat szeretné
Valós Always-On	

Cloudflare	
Pros	**Cons**
Egyben deployolható minden része	Eltérő felépítés
Idle idő nem költség	Nehezebben kezelhető felület
Sokkal szélesebb userbase / „Nagyobb nevű”	Bizonyos funkciók külön modulokban vannak (bár semmi ellehetetlenítő)
Nem kell semmilyen fizetési mód	Nagymértékű újratervezést igényel a projekt
Szebb alap domain	Lehet mégis van minimális cold start, már nem tudom követni, de nem tűnt fel