



Michał Mazur

Zrób se cebulowego SaaSa,
czyli cebularstwo stosowane w projektach IT



Javeloper
2026
spring edition

14.05.2026



 **techevents**



Zrób se cebulowego SaaSa

Cebularstwo stosowane w projektach IT

Javeloper 2026 - Michał Mazur



Mindset cebulowego developera



- Cebulowy developer = osoba, która świadomie tnie koszty bez ukrywania ryzyka
- Po co przepłacać? Szukamy szybszych i tańszych rozwiązań
- **Darmowe tiery, tanie VPS-y, open source, dzielone licencje**
- **Cebula działa:** MVP, hobby, edukacja
- **Cebula nie działa:** enterprise, wysokie SLA (gwarantowana dostępność), skala, wymagania prawne



WE NEED TO GO

CHEAPER

O czym to będzie?

- Jak postawić projekt **za darmo albo prawie za darmo**
- Compute, bazy, storage, CI/CD, security, monitoring, licencje – **wszystko cebulowo**
- Wzorce  i antywzorce 
- Case study: **NeighborhoodWatch**



Kto ja?

- 16 lat w IT – freelancer, małe firmy i korporacje
 - Po godzinach: Junto, cybernetyka społeczna, mentoring, coaching kognitywny
- 👉 **Dziś pokażę, jak zrobić SaaS za uczciwą cenę i nie dać się złapać na pułapki.**

NeighborhoodWatch

Społecznościowe zgłaszanie problemów w okolicy (dziura w drodze, światło nie działa).

Jedno case study dla wszystkich decyzji: koszt, czas, ryzyko i wyjście awaryjne.



NeighborhoodWatch: wymagania startowe

Aplikacja

Publiczny formularz i lista zgłoszeń dla mieszkańców.

API

Backend do zgłoszeń, statusów i kont użytkowników.

Postgres

Relacyjna baza bez ręcznego stawiania klastra.

Zdjęcia

Kilka GB plików na start, bez drogich pobrań.

Admin

Panel tylko dla właściciela projektu, poza publicznym internetem.

Budżet MVP: 0-25 zł/mies. Setup ma trwać godziny, nie tygodnie.

Najtańszy stack to ten, który dowozi cel projektu przy akceptowalnym koszcie czasu i ryzyka.

Jak obliczyć, co się opłaca, a co nie?

- Ilość pracy (setup + maintenance + development)
- Co będzie potrzebne i czego chcesz się nauczyć?
- Koszty (miesięczne i jednorazowe)
- Jak szybko chcesz mieć działający produkt?



Cebulo-metryka opłacalności

Liczmy koszt całkowity przez horyzont 6 mies.:

$$C = M \cdot 6 + T \cdot S + R - L$$

- M – miesięczny koszt infra
- T – godziny setup/utrzymania
- S – Twoja stawka godzinowa (albo „wartość wolnego czasu”)
- R – ryzyko (nieprzewidzane opłaty i wyzwania)
- L – wartość nauki (odejmujemy, jeśli chcesz się tego świadomie nauczyć)

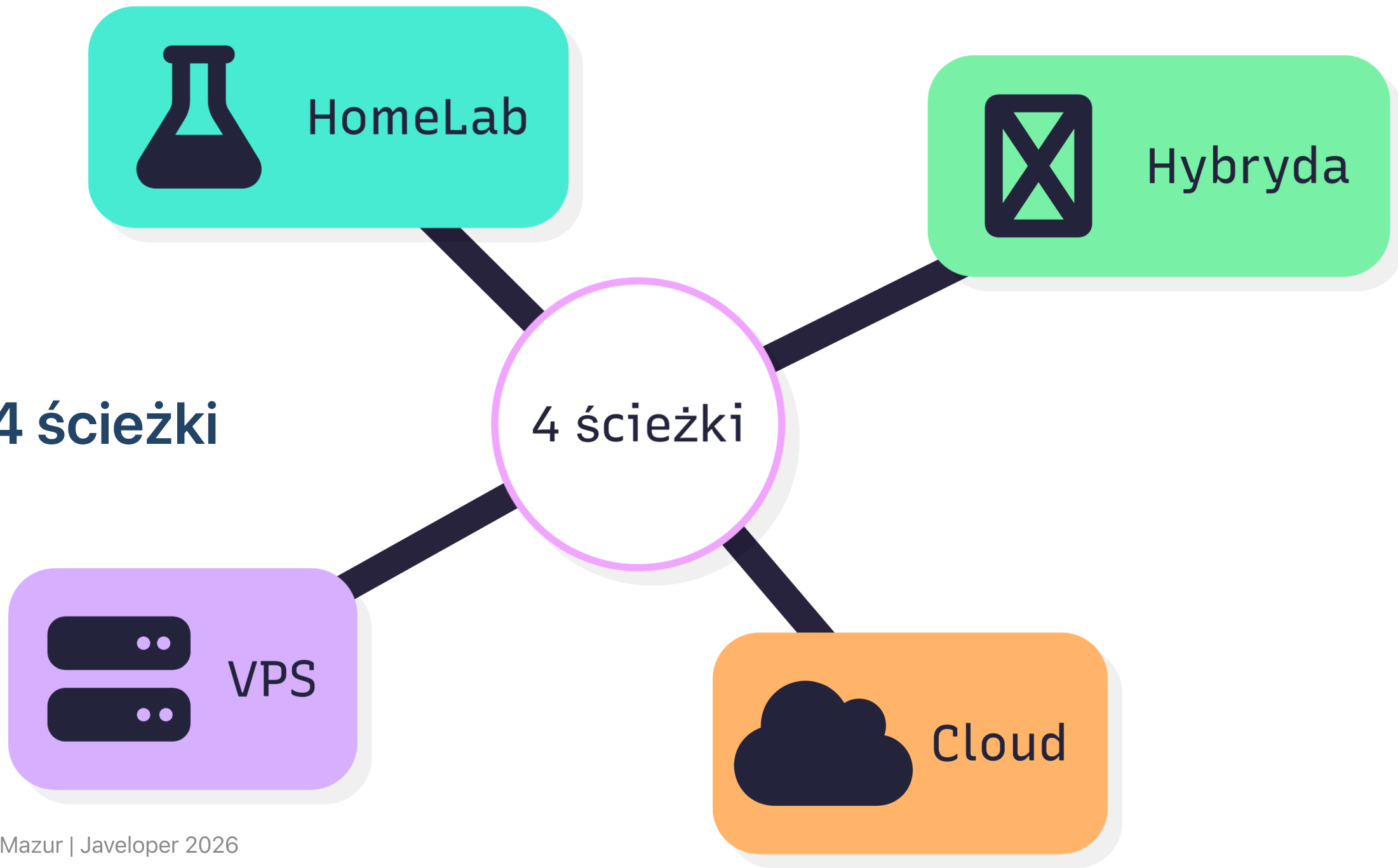
Przykład: Supabase vs VPS

Założenie: $S = 150 \text{ zł/h}$, horyzont 6 mies.

Opcja	M	T	Szacunek
Supabase Free	0 zł	1 h	150 zł
VPS + Postgres	25 zł	8 h	1350 zł

Jeśli walidujesz produkt, czas zwykle wygrywa z abonamentem. Jeśli uczysz się infry, odejmujesz L.

4 ścieżki



Ścieżka 1: 🏠 Homelab

- ✓ Można wykorzystać stary sprzęt, np. laptopa
- ✓ Pełne możliwości konfiguracji
- ✓ Można się dużo nauczyć
- ✗ Bardzo trudna konfiguracja
- ✗ Mało skalowalne
- ✗ Prąd
- ✗ Dostęp do/z internetu (stałe IP, tunelowanie, izolacje sieci)

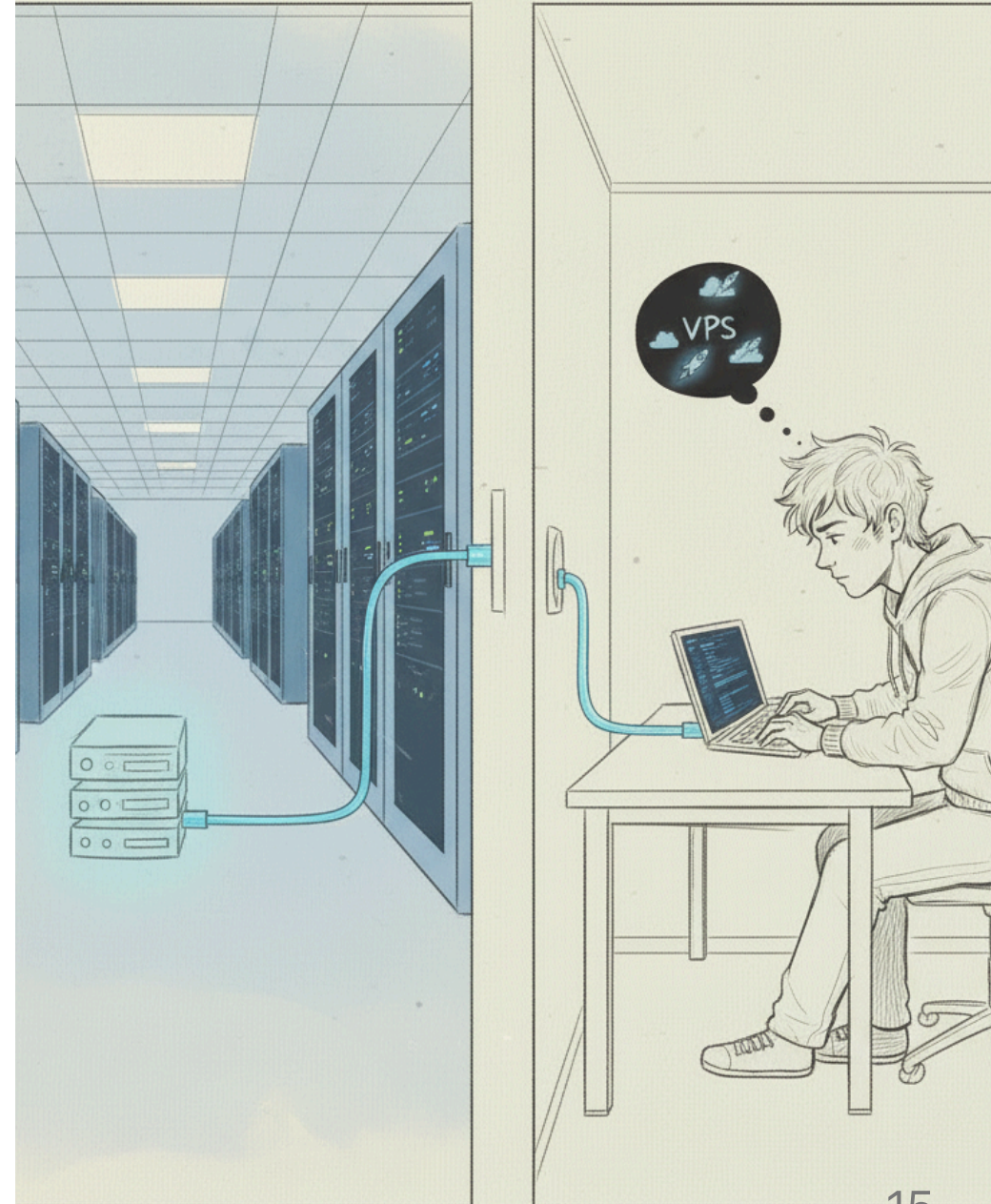


PROJECT LAUNCH!

Ścieżka 2: 🌐 VPS

Wykupiony serwer z aplikacją(-ami) siedzi w data center

- ✓ Datacenter odpowiada za utrzymanie serwera
- ✓ Pełne możliwości konfiguracji
- ✗ Stały koszt
- ✗ Ograniczona wydajność
- ✗ Ukryte koszty w postaci limitów, np. egress



Ścieżka 3: ☁ Chmura

- ✓ Sprawdzone i pewne rozwiązania w każdej skali
- ✓ Bardzo szybko można postawić aplikacje
- ✓ Darmowe tokeny na start / free tier
- ✗ Można popłynąć finansowo
- ✗ Vendor lock-in



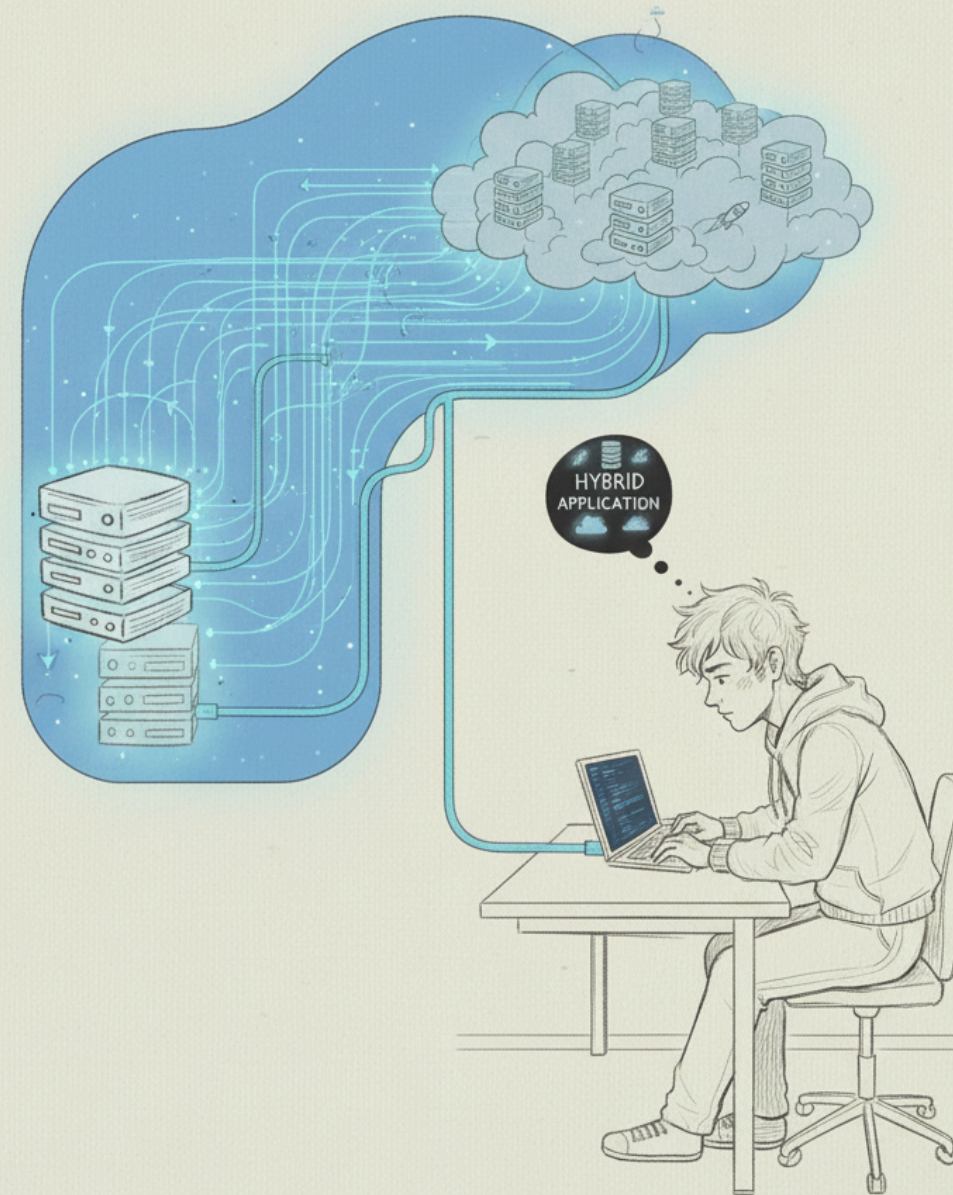
 **Kto miał sytuację, że darmowy limit nagle się skończył i rachunek zaskoczył?**

Napiszcie w czacie „tak”, jeśli limit Was kiedyś zaskoczył.

Ścieżka 4: Hybryda

Łączymy style, mieszamy gatunki...

- ✓ Wybieramy to, czego potrzebujemy
- ✓ Wybieramy to, co najbardziej się opłaca



Cena z faktury to nie cały koszt

Homelab

KOSZT

niski prąd + sprzęt

SETUP

wysoki

RYZYKO

sieć, prąd, utrzymanie

NAJLEPSZE DLA

nauka, hobby, lab

VPS

KOSZT

5-25 zł/mies.

SETUP

średni

RYZYKO

limity, aktualizacje,
backup

NAJLEPSZE DLA

małe SaaS, API, staging

Cloud free tier

KOSZT

0 zł na start

SETUP

niski

RYZYKO

billing, limity, lock-in

NAJLEPSZE DLA

walidacja MVP

Hybrid

KOSZT

0-25 zł/mies.

SETUP

średni

RYZYKO

więcej integracji

NAJLEPSZE DLA

tani start + plan B

Homelab jest tani w pieniądzu, drogi w utrzymaniu.

Zanim wybierzesz stack — 5 pytań

1. **Ile miesięcznie chcesz wydać (i ile Twój czas jest wart)?**
2. **Co musi działać w weekend — a co może poczekać na Ciebie w poniedziałek?**
3. **Jaki jest Twój plan B, gdy provider zmieni warunki lub projekt odniesie sukces?**
4. **Gdzie chcesz mieć kontrolę i odpowiedzialność?**
5. **Jakie masz cele?**

3. Cebulowe zestawy

Compute, bazy, storage, monitoring, security

Ceny i limity: stan na maj 2026.

NeighborhoodWatch: stack startowy

FRONTEND	Vercel / Netlify	statyczny UI, deploy z <code>git push</code>
API	Cloud Run	kontener z Javą, GraalVM native skraca cold start
DANE	Supabase Free	Postgres + Auth bez stawiania bazy
ZDJĘCIA	Cloudflare R2	10 GB free, brak opłat za pobrania
MONITORING	Better Stack + Sentry	uptime, logi i błędy aplikacji
ADMIN	Tailscale	panel poza publicznym internetem

Compute: darmowy start

Najmniej setupu

- Vercel/Netlify: hosting frontendu; deploy z `git push`
- Cloud Run: kontener na żądanie; 2M req/mies. free
- Railway: PaaS dla backendu; free: \$1 kredytu/mies., Hobby: \$5/mies.

Na co uważać

- Cold start = kilka sekund opóźnienia po uśpieniu aplikacji
- GraalVM native image kompiluje Javę do binarki, więc skraca start JVM
- Limity free tier trzeba sprawdzić przed prezentacją i wdrożeniem

Compute: VPS klasyk

- Mikr.us / Hetzner + Docker Compose + Caddy
- Caddy = web server z automatycznym SSL, alternatywa dla nginx
- Jeden plik `compose.yml` wystarczy do startu i odtworzenia usługi
- Migracja: `scp` na inny VPS i aplikacja działa dalej

```
git pull && docker compose up -d
```

Koszt: około 5-25 zł/mies. Stały rachunek, pełna kontrola nad runtime.

Compute: Homelab

- Raspberry Pi 5, mini PC albo stary laptop z Linuxem
- Docker + Tailscale albo Cloudflare Tunnel
- Tunnel = dostęp do prywatnego serwera bez publicznego IP
- Coolify / CasaOS = self-hosted PaaS z GUI

Koszt: około 0 zł/mies. poza prądem. Ryzyko: sieć domowa, zasilanie i utrzymanie sprzętu.

Bazy danych: prosty start

SQLite + Litestream

- Baza = pojedynczy plik
- Litestream = replikacja SQLite do R2/B2
- Zero serwera DB, świetne do MVP (~kilku tys. userów)
- ⚠ jeden writer, słabe dla rozproszonych zapisów

Supabase Free

- 500 MB DB, 50k MAU (aktywnych użytkowników mies.)
- RLS (Row Level Security) = uprawnienia do wierszy w Postgresie
- Pod spodem czysty Postgres

Bazy danych: self-hosted

 Postgres + pg_dump → R2/B2 — *self-hosted klasyk*

- Postgres w Dockerze
- ~5 zł/mies za VPS, R2/B2 free 10 GB na backupy
- Pełna kontrola, agnostyczne, łatwa migracja

Kiedy to ma sens?

- Masz VPS-a i chcesz pełną kontrolę
- Umiesz zadbać o backup i monitoring
- Nie chcesz vendor locka na warstwie DB

Storage / CDN: cloud

Cloudflare R2 + Cloudflare CDN — *pliki bez opłat za pobrania*

- 10 GB storage free
- Cloudflare CDN = cache plików przy użytkowniku
- S3 API: działa z bibliotekami dla Amazon S3
- Egress = ruch wychodzący; w R2 za pobrania nie płacisz

Backblaze B2 + Bunny CDN — *storage + cache przy dużych plikach*

- B2 = storage, Bunny = CDN/cache przy użytkowniku
- B2 free 10 GB + Bunny ~\$0.01/GB transferu
- Bandwidth Alliance: B2 ↔ Bunny = transfer za 0 zł
- Dobre dla ciężkich plików: video, ISO, paczki

Storage / CDN: VPS i homelab

 **Dysk VPS + Caddy** — *0 zł ponad cenę VPS-a*

- Pliki na lokalnym dysku, Caddy serwuje z cache-headers
- Caddy file_server + gzip + ETag = wystarczający CDN dla MVP
- Backup: rsync/restic do dowolnego storage (B2, R2, Storage Box)

Kiedy to ma sens?

- Masz VPS z wystarczającym dyskiem
- Kilka GB zdjęć, dziesiątki użytkowników, nie setki tysięcy
- Chcesz najprostszy możliwy setup bez zewnętrznych zależności

Reverse Proxy / WAF

Reverse proxy = brama HTTP przed aplikacją.

WAF = Web Application Firewall.

Reverse proxy

VPS/Homelab: Caddy lub nginx + Cloudflare.

SSL/TLS

VPS i Homelab: Certbot albo Caddy.

WAF / DDoS

VPS: Cloudflare proxy.
Homelab: Cloudflare Tunnel.



Web Server: nginx vs Caddy (przykład)

nginx

```
server {
    listen 443 ssl;
    ssl_certificate /etc/ssl/cert.pem;
    ssl_certificate_key /etc/ssl/key.pem;
    location / {
        proxy_pass http://app:3000;
    }
}
```

Caddy

```
myapp.com {
    reverse_proxy app:3000
}
```

Odpada ręczna konfiguracja certyfikatów SSL dla prostego reverse proxy.

Monitoring: SaaS free

Najmniej setupu

- *Better Stack*: uptime, logi, status page i alerty w jednym UI

Klasyczny zestaw

- *UptimeRobot*: sprawdza, czy strona odpowiada
- *Sentry*: błędy aplikacji i trace'y
- *Grafana Cloud*: metryki i logi, gdy potrzebujesz wykresów

Monitoring: self-hosted

 Self-hosted in a box — *jeden compose, dane u Ciebie*

- *Uptime Kuma* = uptime, *Netdata* = metryki, *GlitchTip* = błędy
- Wszystko w jednym `docker-compose.yml`
- brak limitów, brak vendor locka
- ⚠ Ty pilnujesz, że sam monitoring nie padnie

Kiedy to ma sens?

- Chcesz trzymać logi u siebie
- Masz już VPS-a
- Akceptujesz własne utrzymanie monitoringu

Security: sieć i dostęp

Cloudflare Free + hardening VPS

- *Cloudflare* proxy/WAF + UFW + fail2ban + auto-updates
- Niemal bezkosztowo, dobrze chroni przed botami i DDoS

Tailscale + Caddy + Authentik

- *Tailscale* free: nielimitowane urządzenia użytkowników, do 6 użytkowników
- SSH/admin tylko w prywatnej sieci VPN
- *Authentik/Authelia* = SSO/2FA przed aplikacjami
- Panel admin niewidoczny z publicznego internetu

Security: sekrety i backupy

Sekrety

- *SOPS* szyfruje `.env` w repo; age trzyma klucz lokalnie
- *1Password Developer* trzyma sekrety poza kodem

CrowdSec + restic + B2/R2

- *CrowdSec* = społecznościowe blocklisty zamiast samego fail2ban
- *restic* = backupy szyfrowane po stronie klienta
- Backup do R2/B2
- Codzienne backupy: konfigi, DB, sekrety, kod

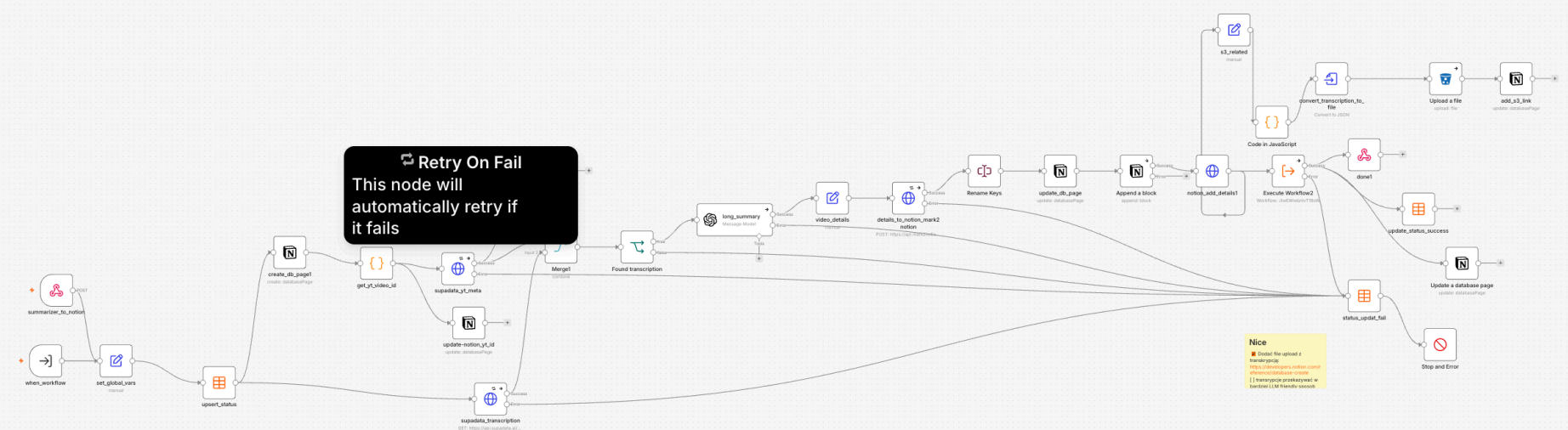


Antywzorce (czyli jak spalić cebulę)

- **Overengineering:** Kubernetes do MVP dla 5 użytkowników, Cache na Redis jak nie ma ruchu
- **Brak backupu::** infra, konfiguracje, dane, kod itd.
- **Sekrety w repo:** sekrety w repozytorium
- **Brak CI/CD:** ręczne testowanie i wdrażanie
- **Brak logowania:** „Zrobię logi później” (czytaj: nigdy)
- **Brak zabezpieczeń:** API, serwer, baza danych, sekrety, anty-spam itd.
- **Brak monitorowania:** metryk, kosztów i zużycia free tieru

Czy musisz programować?

Może można to gdzieś wyklikać i będzie działało?



🔄 🔍 🔍 ↶ 🖌️

Execute workflow from summarizer_to_notion

Logs

Triki oszczędzania na subskrypcjach

- Sprawdź, czy w Twojej firmie wykupione subskrypcje mają opcję dołączania kont prywatnych za darmo (mój case: 1Password)
- Jeśli używasz licencji premium, kliknięcie rezygnacji często pokazuje kod promocyjny
- Subskrypcje „organizacyjne” czy „rodzinne”: np. Spotify Duo, Google Workspace, dzielone subskrypcje YT
- Projekty "open-source", "fundacyjne" lub "non-profit" mają często darmowe subskrypcje np. IntelliJ
- Studenci mają często darmowe subskrypcje np. IntelliJ, Google Workspace
- Karty prepaid/wirtualne: Revolut, Zen, Wise

Pieter Levels

- Shipuj szybko, często i publicznie – „make something people want”
- Minimalny stack, iteracje po feedbacku
- Dystrybucja > produkt: SEO, social, content
- Automatyzuj wszystko, co się da (zapisy, płatności, wsparcie)
- Build once, sell many (SaaS, subskrypcje, side-projects)
- Zobacz: <https://levels.io/>





Obserwuj

@levelsio ✓

@levelsio

Pokaż tłumaczenie

[PhotoAI.com](#) \$100K/m

[RemoteOK.com](#) \$44K/m

[Vibej.am](#) \$39K/m

[InteriorAI.com](#) \$35K/m

[levelsio.com](#) + [@X](#) \$14K/m

[Nomads.com](#) \$10K/m

[Pieter.com](#) \$0/m

[My book](#) \$3K/m [readMAKE.com](#) Dołączył/a lipiec 2013 >

@levelsio

+7

projektów

+860 tys.

obserwujących

5. Backup

Shit happens....



Pożar OVH, 2021

W Strasburgu spłonęło centrum danych. Część klientów straciła dane, których nie miała poza tą lokalizacją.

Zasada backupu 3-2-1

3

kopie danych

1 oryginał + 2 kopie zapasowe

2

różne nośniki

np. komputer + dysk
zewnętrzny

1

**kopia poza główną
lokalizacją**

np. chmura lub inna lokalizacja

Cel: ochrona przed awarią, usunięciem danych i kradzieżą

GitLab.com, 2017

Inżynier przypadkowo usunął główną bazę danych.

Backupy nie działały albo nie były gotowe do szybkiego użycia.

18 godzin outage, około 6 godzin utraconych danych.

[Postmortem GitLab](#)

MIEDZY BOGIEM A PRAWDA



GRATIS TO UCZCIWA CENA

Kiedy przestać cebulić?

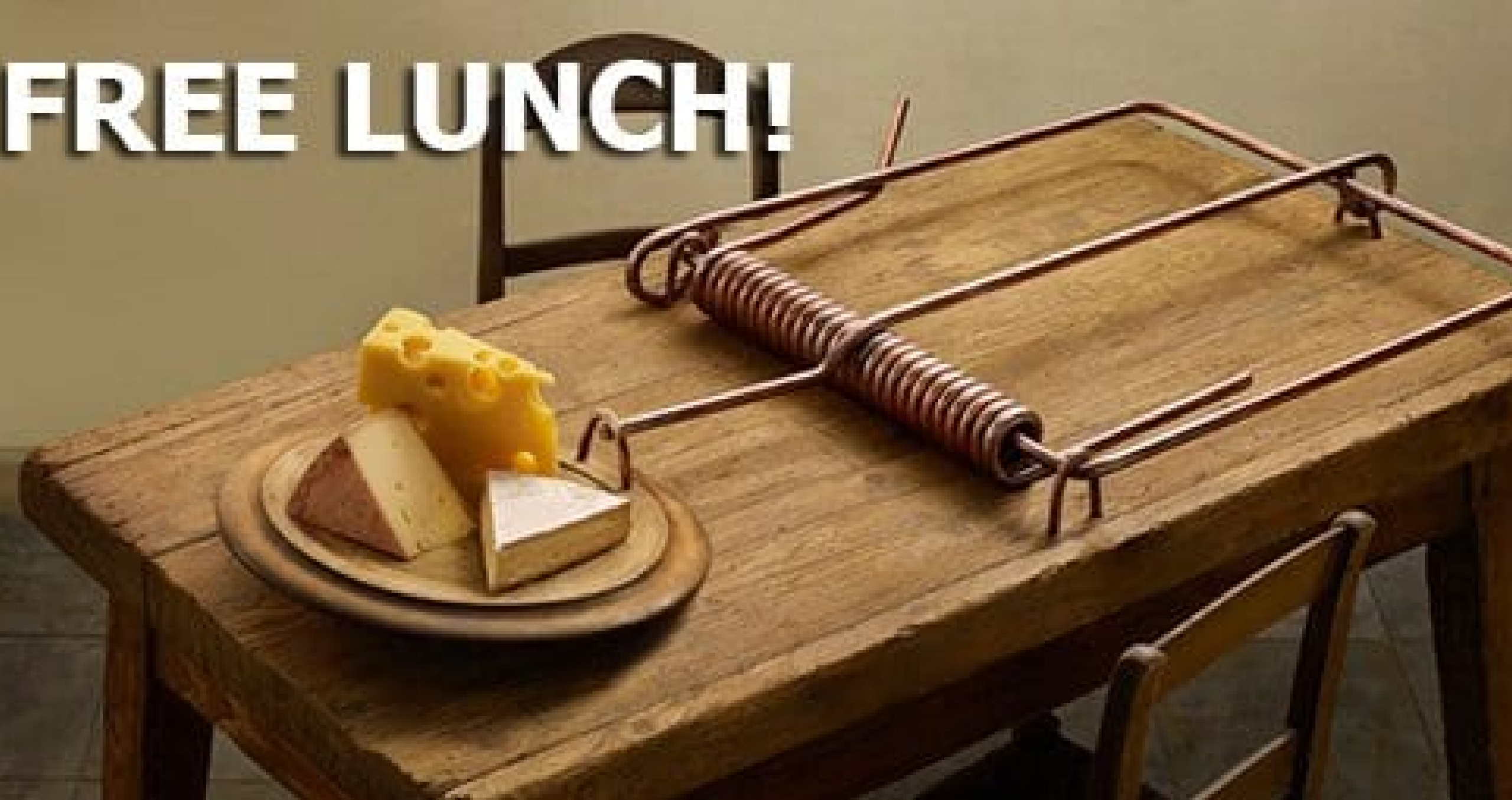
Kiedy **koszt Twojego czasu** > koszt usługi.

Kiedy **pierwszy płaćący klient** puka do drzwi.

Kiedy **awaria o 3 w nocy** kosztuje więcej niż \$10/mies. za managed service.

Cebula to świadomy etap. Nie tożsamość.

FREE LUNCH!



Zadanie na dzisiaj:

Weź swój bieżący projekt. Policz **cebulo-metrykę**.

Znajdź **jedną** usługę, za którą płacisz, a mógłbyś nie płacić.



Bonus:

Kontrowersyjne triki ciemnej strony cebuli - Darth Samsiad

Oszczędności, które mogą kosztować reputację, konto, dane albo zgodność z regulaminem.

Grupowe subskrypcje

Dzielenie kont premium/enterprise z innymi osobami. Oszczędność jest realna, ale regulaminowo i reputacyjnie ryzykowna.

- [promptowy](#)
- [oliwier1pl](#)
- [Lenny's newsletter](#)

Tymczasowy e-mail do testów i skrzynka SMS

- tymczasowe skrzynki e-mail: niepodam.pl
- skrzynka mailowa `.edu` i `.com` : Temp Mail
- międzynarodowa bramka SMS: JuicySMS
- Email alias przez np. iCloud

Q&A

Twitter: MichałMzr

LinkedIn: michmzr

GitHub: michmzr

Materiały dodatkowe: <https://bit.ly/javeloper2026-cebula>

Dzięki za uwagę! Michał Mazur

