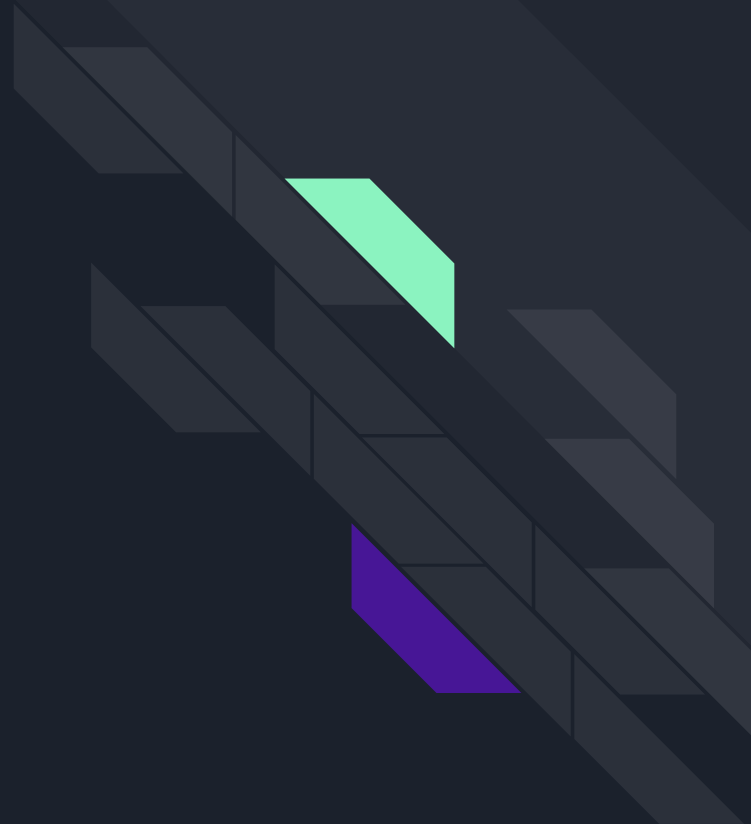



dr. Radics Ottó

# WP + Docker alapok

**A weboldal-költöztetés ne legyen fejfájás!**

Mi kell ahhoz, hogy egy  
weboldalt publikáljunk?





Mi kell ahhoz, hogy egy weboldalt publikáljunk?

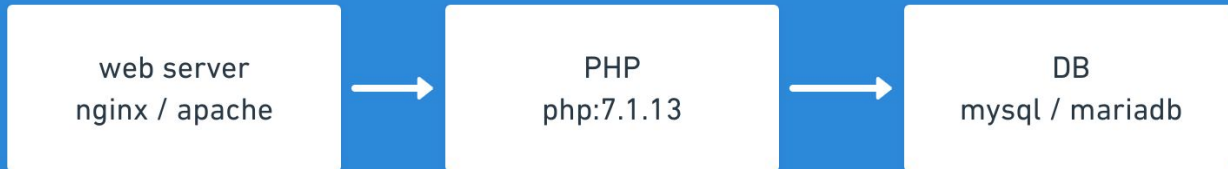
1. Netkapcsolat
2. Szerver
3. Operációs rendszer
4. Webes stack

Milyen elemekből épül fel a stack, ami kiszolgálja a látogatóinkat?



Milyen elemekből épül fel a stack, ami kiszolgálja a látogatóinkat?

b.hu



# Megszokott tárhely megoldások





Megszokott tárhely megoldások

## Szolgáltatások

### “Admin felületek”

1. cPanel
2. DirectAdmin
3. Plesk
4. Virtualmin
5. etc.

### Funkciók

1. FTP hozzáférés
2. Levelezés
3. Naptár szinkron
4. Backup
5. DNS kezelés
6. etc.




Megszokott tárhely megoldások

## Korlátozások

1. Domain, aldomain
2. Adatbázis
3. Lemezhasználat
4. Memória
5. E-mail fiókok
6. FTP fiókok
7. Entry processes
8. IOPS, I/O limit
9. Szoftver verzió





Megszokott tárhely megoldások

## Hátrányok/hiányok

1. Mi alkalmazkodunk
2. A pontos beállítások nem mindig ismertek
3. Nehéz reprodukálni a környezetet és az “egzotikus” hibákat (php.ini és társai)
4. Nem szabadon választott a stack
5. Korlátozások
6. SSH?
7. git?
8. CI/CD?
9. staging környezet?

Megszokott tárhely megoldások

## Igények

### Ne legyen:

- a stack-en belül programverzió eltérés a dev és a prod között
- konfigurációbeli eltérés
- jogosultság probléma

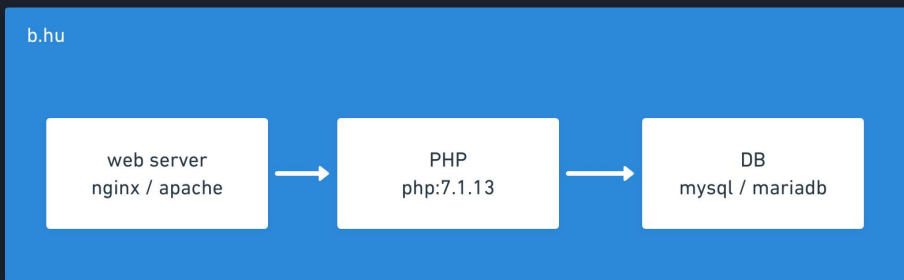
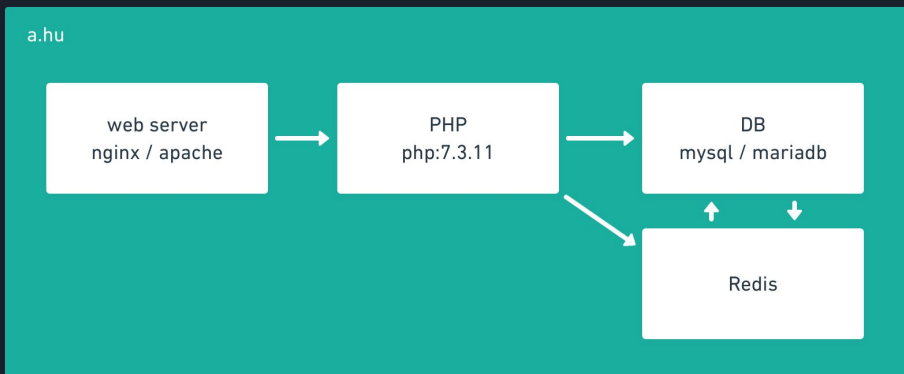
### Legyen:

- a konfiguráció része a projektnek
- minden cronjob a projekt “része”
- lehetőség eltérő projekt követelményeket kielégíteni “ugyanazon a gépen”

Reprodukálni “ugyanazt” nehéz: kattintgatás, nem kódvezérelt, nehezen automatizálható.

Megszokott tárhely megoldások

## Elvárás



# Mi a Docker?

## Alapfogalmak, jellemzők




## Mi a Docker? Alapfogalmak, jellemzők

- Docker
- Docker Compose
- Image
- Container
- Service
- Volume
- Network
- Lifecycle
- “VM”
- Reprodukálható
- Könnyű
- Skálázható
- Eldobható
- Determinisztikus (jó esetben)

Részletek: <https://docs.docker.com/engine/docker-overview/>

Hogyan ad a Docker a  
fenti problémákra  
megoldást?






Hogyan ad a Docker a fenti problémákra megoldást?

## **Stack-en belüli programverzió eltérések**

A megépített image-ben szerepel “minden”: a futtatókörnyezet és a tényleges alkalmazás egyaránt.




Hogyan ad a Docker a fenti problémákra megoldást?

## Konfigurációbeli eltérések & a konfiguráció nem része a projektnek

A konfig fájlokat a projekt könyvtárában tároljuk, így azok is verziókövetve vannak, az image buildelése során másolódnak be a megfelelő helyre. (kb. mint a `.htaccess` fájlok)






Hogyan ad a Docker a fenti problémákra megoldást?

## **Az esetleges cronjobok nem részei a "projektnek"**

A cronjobok egy konfig fájlban tárolódnak "valahol" (crontab), így nem idegen megoldás, ha a cron beállításokat konfig fájlok módjára kezeljük: külön service, ami futtatja a cronjobokat, melyeket egy fájlban/könyvtárban tárolunk - szintén a repository-ban.




Hogyan ad a Docker a fenti problémákra megoldást?

## Jogosultság problémák

Hamarabb kijönnek: ha az oldal megfelelően működött a konténerben, lokálisan, akkor jól fog működni az élesen is\*.

\* Feltéve, hogy container image-t terítünk, nem hibrid megoldást - erről később.



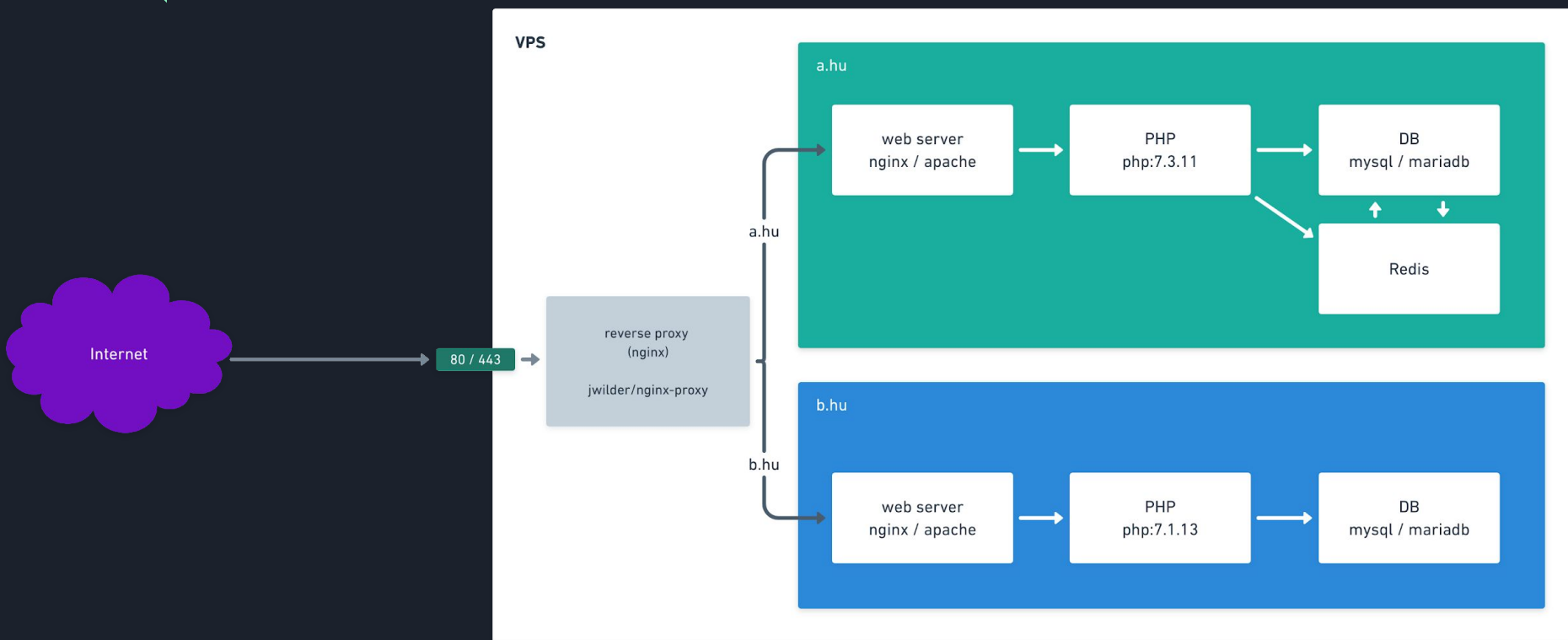
Hogyan ad a Docker a fenti problémákra megoldást?

## Eltérő projekt követelmények ugyanazon tárhelyen

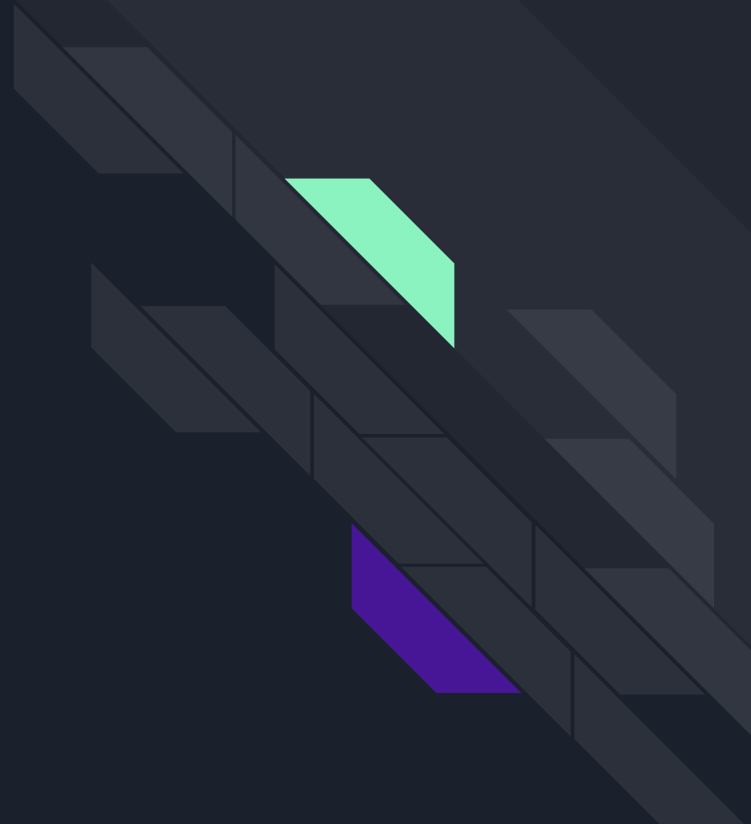
Reverse proxy: [jwilder/nginx-proxy](#)

Hogyan ad a Docker a fenti problémákra megoldást?

## Eltérő projekt követelmények ugyanazon tárhelyen



Hogyan csináljuk?



Hogyan csináljuk?

## Leszakadás a tárhely szolgáltatótól

### Feladatok

- Domain regisztrátor
- “Vas”
- Adatbázis
- E-mail fiókok
- DNS
- Registry (erről később)
  
- Repository
- Backup
- Levélküldés

### Javaslat

- .hu: mindegy, egyéb: Cloudflare
- DigitalOcean, (G/DO, linode, etc.) Kubernetes Engine
- VPS / managed DB (DO, GCE, etc.)
- G Suite
- Cloudflare (.hu is)
- Publikus: Docker Hub,  
Privát: GitLab
- GitLab (tök jó a CI/CD rendszerük)
- Object storage (S3, DO Spaces, etc.)
- Mailgun (10 ezer e-mail havonta free)



Hogyan csináljuk?

## Deployment

### Ideális

- teljes image-t terjesztünk
- futtatókörnyezet + fájlok

### Hibrid

- image egy alap
- csak futtatókörnyezet
- fájlokat csatoljuk

# Nehézségek és megoldási javaslatok







Nehézségek és megoldási javaslatok

## Mi legyen a feltöltött fájlokkal?

Javasolt megoldás valamilyen object storage:

- Amazon S3
- DigitalOcean Spaces
- Google Cloud Storage,
- etc.



Nehézségek és megoldási javaslatok

## Mi legyen a feltöltött fájlokkal?

Roszzabb\* esetben “valahol” tároljuk:

- NFS
- host

\* backup, hogyan reprodukáljuk, etc.



Nehézségek és megoldási javaslatok

## Verziózás

Akarjuk vagy nem, egy sima WP oldal esetén is egyedi alkalmazást fejlesztettünk.

Érdemes scriptet írni, a verziót valahol tárolni kell:  
`package.json` jó kiindulási alap lehet.

Statikus assetek verziózására “rendes” build pipeline.



Nehézségek és megoldási javaslatok

## Környezeti változók eltérésével mit lehet kezdeni?

Környezeti változók - wat?

Build során kerüljenek bele a container image-be

Használjuk a `getenv()`-et

*(De jó lenne, ha menne... :P)*

[roots.io/bedrock](https://roots.io/bedrock)

Nehézségek és megoldási javaslatok

## Privát registry

Vannak ingyenes privát registry-k, használjuk azokat:

- Google Container Registry (gcr.io)
- GitLab Registry

(Vagy üzemeljünk be sajátot. 🐱)

Demo :)





GitLab repository:

<https://gitlab.com/webmenedzser/wpdockerdemo/>





## Olvasnivaló

[Dockerise your PHP application with Nginx and PHP7-FPM](#)

[Making your dockerised PHP application even better](#)

[Getting started with Docker for the Inquisitive Mind](#)

[Don't use FTP — Here's why](#)

[Weboldalak biztonsági mentése - hogyan csináld jól szabadúszóként/ügynökségként?](#)

[Docker alapú VPS tárhely éles környezetben](#)

[GitLab CD beállítása avagy autodeploy ingyér'](#)





Köszönöm a  
figyelmet!