

Staging környezetek WordPress-hez

Milyen lehetőségeink vannak a párhuzamos
fejlesztésre, tesztelésre

Rólam

Rottenbacher Tamás

- » WordPress Magyarország csapatához tartozom 2016 óta és a Nyílt Web Alapítvány egyik alapítója vagyok.
- » 2006 óta foglalkozok honlapkészítéssel, 2008 óta SEO-val és WordPress fejlesztéssel, 2010 óta WordPress oldalak üzemeltetésével, jó ideje saját vállalkozásban.
- » 2018 óta több szerverrel rendelkezem (Cloud VPS-el már).
- » Több hazai WP-s projektet vezetek, ami a közösséget támogatja, például TrustedWP.hu alapító, szerkesztő vagy a WordPress Budapest MeetUp előadásorozat egyik szervezője, előadója vagyok.



Nyílt Web Alapítvány



Mi az a staging környezet?

- » Az éles, működő honlapnak egy olyan másolata, mely nem csak magának a honlapnak tartalmazza a másolatát (fájlok és adatokat egyaránt), hanem a szoftveres környezetnek is egy pontos letükrözése.

Mi tartozhat ide? Például:

PHP verzió beállításokkal, adatbázis verzió beállításokkal, Apache vagy Nginx rendszer, hardveres és szoftveres limitek



**A staging környezet lehetőséget
biztosít, hogy az elefántot
bezavarhassuk a porcelánboltba,
hogy megnézzük a
következményeket.
Mindezt kockázat nélkül.**

Miért van szükség a staging környezetre?

- » Az éles weboldal érték, pénzt termel
Nem rizikózunk vele!
- » Általában 3-4* bővítményt tesztel egy felhasználó, mire megtalálja melyik kell.
Közben az adatbázisban gyűlik a szemét..
- » Ha hiba történik és összeomlik az oldal...
Nem veszítünk pénzt, káromkodunk és újra próbáljuk.
- » Tesztelni, ellenőrizni folyamatosan szükséges.
(Elég csak WP frissítésre gondolni.)
Ha akarjuk, ha nem és mindig van esély a bajra.



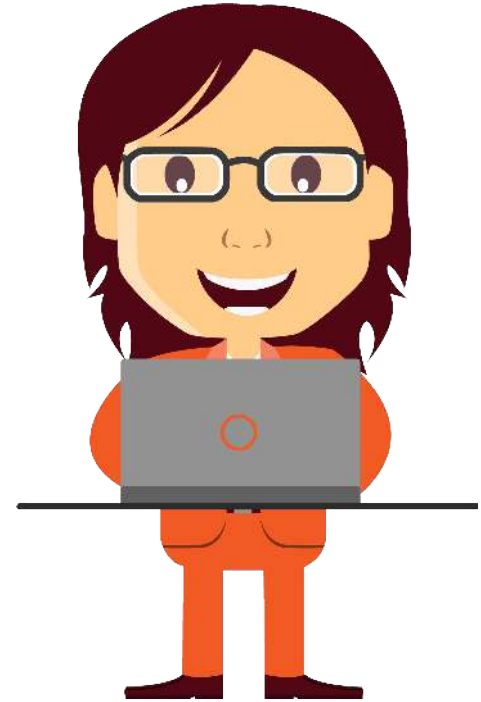
Milyen lehetőségeink vannak?

A következőkben bemutatok több lehetőséget, amik között a hétköznapi felhasználó éppúgy, mint a kezdő fejlesztő vagy a haladó senior tud választani számára ideálisat.

Minden esetben mérlegeljük, hogy mik a lehetőségeink és azzal párosítható megoldást válasszunk.

Haladjunk a favágós megoldástól indulva az ideálisabb megoldások felé.

- 1. Tárhelyen belüli**
- 2. Szerveren belüli**
- 3. Lokális staging**
- 4. Konténerizáció**



Tárhelyen belüli

Ennek során, a meglévő hostingot / tárhelyet használjuk és gyakorlatilag egy mappába másoljuk a honlapot, egy ugyanazon szolgáltatónál lévő másolt adatbázissal és így egy domain.hu/proba (vagy aldomain-en) érhetjük el a **(nem valódi)** staging honlapunk.

Ennek vannak előnyei és hátrányai...

Pro

- » Nem igényel szakmai ismeretet
- » A PHP beállítások, verzió, az adatbázis és a limitek is azonosak
- » Költséghatékony
- » A tárhelykezelők WP telepítőivel + WP költöztetéssel kényelmes megoldást ad

Kontra

- » Az átlagos shared hosting limit alacsony, így lassíthatja éles honlapunkat és dupla tárhelyméret szükséges
- » Ha a másolt honlapot nem gondozzuk, veszélybe sodorjuk az élelet is
- » Vannak tárhelykezelők amik ilyen telepítésnél saját PHP beállítás fájlt hoznak létre
- » Az éles oldalt több szempontból is érintheti (pl. biztonság, sebesség)
- » Nem tudunk eltérő verziókat tesztelni, csak azonosakat

Szerveren belüli

Itt a meglévő tárhely beállításával azonos “tárhelycsomagot” hozunk létre (vagy rendelünk meg) ugyanazon szerveren (verziók miatt ez fontos) és ezen környezetben végezzük a fejlesztést, tesztelést.

Egyszerűsítve: mintha vásárolnánk egy második tárhelycsomagot.

Pro

- » Nem igényel szakmai ismeretet
- » A PHP beállítások, verzió, az adatbázis és a limitek is azonosak
- » Tárhely backup-al könnyen duplikálható a teljes tárhely tartalma
(Sok tárhelyszolgáltatót meg is lehet kérni, hogy ezt elvégezze, aldomain beállítással együtt.)
- » Eltérő PHP verziókat, beállításokat tesztelhetünk
- » Adhatunk csak a tesztre hozzáférést
- » Szolgáltatótól függő: akár Git is beköthető

Kontra

- » Dupla tárhely / szerver költség
- » A második tárhelycsomagban lévő honlapunk is karban kell tartanunk (például frissíteni) és ügyelni a biztonságára
- » Szükségünk lesz egy aldomain-re vagy külön domain-re (szolgáltató függő)
- » Fontos, hogy a keresőrobotokat tiltsuk ki erről az oldalról, SEO hiba
- » Ha külsősnek hozzáférést adunk, akkor ügyelnünk kell, hogy más jelszavakat használjunk és a fontos adatokat töröljük

**A legtöbb céges honlaphoz,
kisebb webáruházhoz ezt a
megoldást tudnám javasolni.**

Local staging

Ennek során, a szükséges szoftveres környezetet a saját számítógépünkre telepítjük, ezáltal egy kis szervert csinálunk belőle saját magunknak, mely lehetővé teszi, hogy internet hozzáférés nélkül is tesztelhetünk, fejleszthetünk.

Itt is említhetünk előnyöket, hátrányokat...

Pro

- » Könnyen telepíthetőek (*majdnem mind*)
- » Bármilyen veszély nélkül tesztelhetünk, fejleszthetünk
- » Tesztelhetünk eltérő környezetet, így például jelenleg 7.4-en futó weboldalunk PHP 8.1-es környezetben ellenőrizhetjük
- » Költséghatékony, mivel ezek általában ingyenesek
- » Elég sok anyag érhető el online, ami segíthet
- » Offline is tudjuk használni

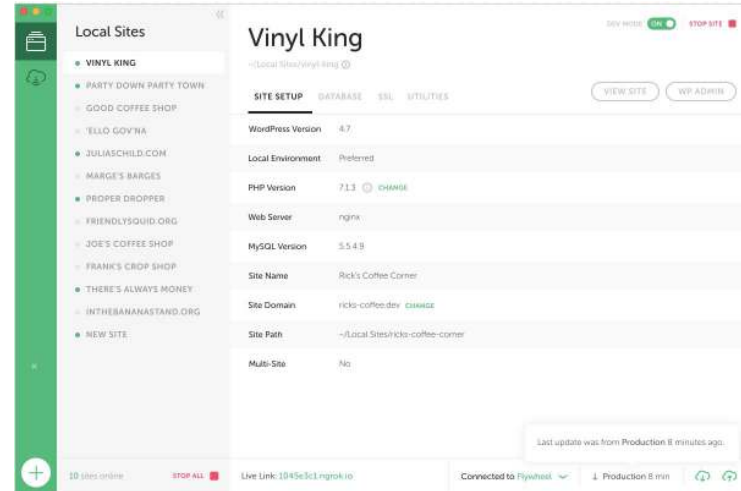
Kontra

- » A környezet (pl. PHP) beállításokat saját magunknak kell beállítani, hogy azonos legyen.
- » Nem minden esetben tudunk pontosan azonos verziókat beállítani
- » Ezek a megoldások általában lassabbak, gyorsításhoz komolyan bele kell nyúlni a beállításokba
- » Több plugin, theme igényel külsős kapcsolatot (például curl-t) és az local környezetbe nem mindig megoldható
- » Ha offline használjuk, nem fogunk tudni net-en képeket keresni, fórumokat olvasni

LocalWP (www.localwp.com)

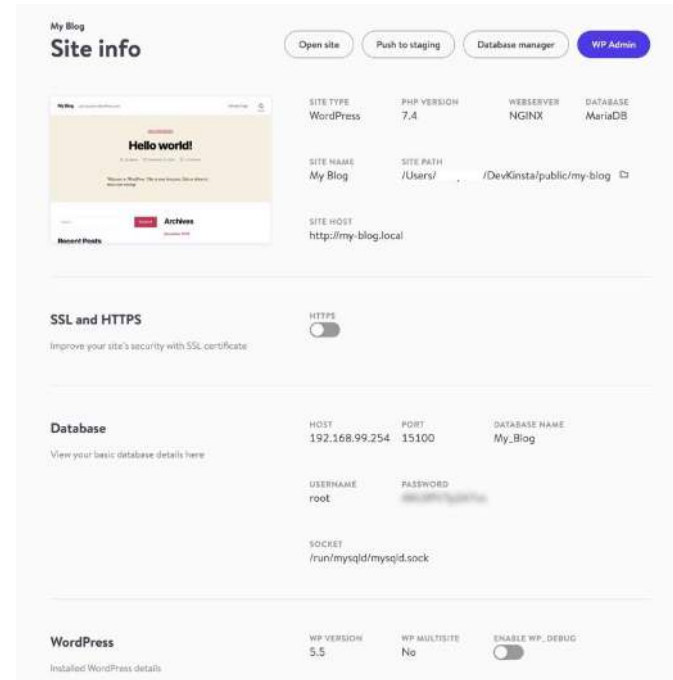
Egyszerű telepítője van, tartalmazza szükséges elemeket, ingyenesen elérhető, egész részletes a PHP verzió választó, van bépített képtimalizáló és a wpengine.com vagy flywheel.com-on host-olás esetén közvetlen a gépünkről kiküldhetjük a honlapunkat és a program felmásolja (deploy) nekünk akár online staging vagy éles közegbe. Windows-ra, MAC-re, Linux-ra egyaránt elérhető. Van saját HUB-ja is, illetve könnyen átlátható felülettel rendelkeznek és van WP-CLI támogatása.

Leírásában csak PHP 7-ig jelölik, ám az újabban a 8-as verzió is választható.



DevKinsta (www.kinsta.com/devkinsta)

Egyszerű telepítője van, tartalmazza szükséges elemeket, ingyenes. Van beépített levelező (csak helyileg). A Kinsta.com szolgáltatóé, így természetesen az ott host-olt honlap közvetlen mehet élesbe gombnyomásra. Rendelkezik egy egyszerű phpmyadmin-hoz hasonló felülettel, ami nagy segítség tud lenni. Windows-ra, MAC-re, Linux-ra egyaránt elérhető. Van külön online webinar hozzá (angolul). Ez Docker alapú, azt is telepíti nekünk. Szintén egyszerű, jól átlátható felülettel rendelkezik. Van WP-CLI támogatása.



The screenshot displays the 'Site info' dashboard for a WordPress site named 'My Blog'. At the top, there are four buttons: 'Open site', 'Push to staging', 'Database manager', and 'WP Admin'. The dashboard is divided into several sections:

- Site info:** A small preview of the website shows a 'Hello world!' message. To the right, a table lists site details:

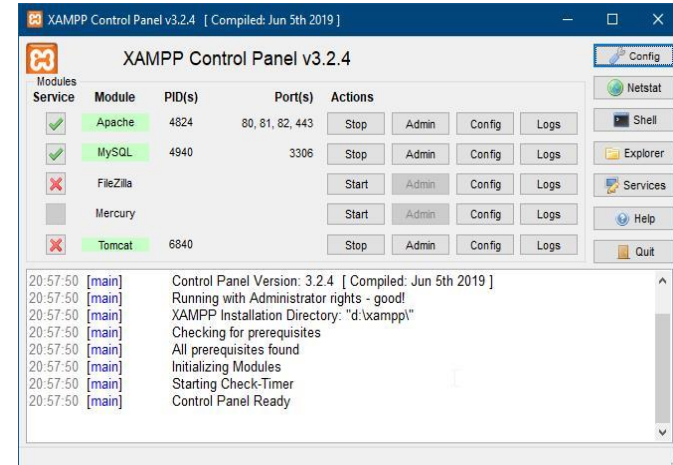
PROPERTY	VALUE
SITE TYPE	WordPress
PHP VERSION	7.4
WEBSEVER	NGINX
DATABASE	MariaDB
SITE NAME	My Blog
SITE PATH	/Users/.../DevKinsta/public/my-blog
SITE HOST	http://my-blog.local
- SSL and HTTPS:** A section with a toggle switch for 'HTTPS' and the text 'Improve your site's security with SSL certificate'.
- Database:** A section with the text 'View your basic database details here' and a table of database details:

PROPERTY	VALUE
HOST	192.168.99.254
PORT	15100
DATABASE NAME	My_Blog
USERNAME	root
PASSWORD	[REDACTED]
SOCKET	/run/mysqld/mysqld.sock
- WordPress:** A section with the text 'Installed WordPress details' and a table of WordPress details:

PROPERTY	VALUE
WP VERSION	5.5
WP MULTISITE	No
ENABLE WP_DEBUG	[Toggle]

XAMPP (www.apachefriends.org)

Az előzőekhez képest picit nehezebb a telepítés és kezelés, például manuálisan kell elindítanunk az egyes szolgáltatásokat mindig. A PHP (és további) beállításokat config fájlban tudjuk módosítani, szóval kicsit ijesztőbbek lehetnek felhasználóknak. Cserébe kevesebb erőforrást esznek és pont a config fájl szerkesztés révén nagyon pontosan beállíthatóak a paraméterek. Van hozzá lokális levelezés és még FTP megoldás is, továbbá lehet saját online szerveren is futtatni. Nagy hibája, hogy csak egy PHP verziót képes kezelni egy telepítéssel és a WP-CLI-t telepíteni kell még hozzá.



Továbbiak

A sort végtelenségig lehetne sorolni. WAMP, LAMP, MAMP, Varying Vagrant Vagrants (VVV) és így tovább. Ezeket nem részletezném.

És a végén... a nagyágyú...



A konténerizáció az előzőekhez képest komplexebb megoldás. Minden szükséges környezeti elemet ami a weboldalhoz kell egy csomagba (konténerbe) kerül és ezt egészbe másoljuk, mozgatjuk, írjuk felül. Ez már egy másik szint...

Pro

- » Szinte bármilyen méretű, rendszerű weboldalhoz használható
- » Bármilyen veszély nélkül tesztelhetünk, fejleszthetünk
- » Legapróbb részig testreszabhatjuk
- » Git-et köthetünk be
- » Elég sok anyag érhető el online, ami segíthet
- » Offline is tudjuk használni
- » Oldalak a környezettel és mindennel együtt másolhatóak

Kontra

- » Komolyabb ismereteket igényel
- » Kisebb honlapokhoz teljesen felesleges, "ágyúval verébre" lenne
- » Normál tárhelyszolgáltatók kiesnek, mivel konkrét szerver, VPS szükséges
- » Rendszergazdai ismeretek is kellenek
- » Ha offline használjuk, nem fogunk tudni net-en képeket keresni, fórumokat olvasni

Köszönöm a figyelmet!

Kérdésekre az utolsó előadást követően szívesen válaszolok!

Rottenbacher Tamás

www.rotisoft.hu

tamas@rotisoft.hu

+3630 387 39 69