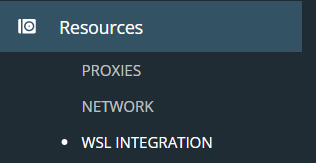
Katkıda Bulunanlar: Süleyman Yiğit, Gökhan Sevinç, Ömer Gürel, Mehmet Durmaz, Naime Damataşı

# Ubuntu Distrosunu kurunuz. Bkz: TMM-DEV-T01

# Docker Desktop yükleme

Docker Desktop Kurulum dosyası <https://www.docker.com/products/docker-desktop>

* Yükleme tamamlandıktan sonra  ayarlara tıklayınız
*  seçiniz
*  seçeneğinin aktif olduğunda ve aşağıda;
*  aktif olduğundan emin olunuz ve
*  düğmesine basınız.
* Ubuntu konsolunda docker ps ve docker-compose komutlarını deneyiniz ve çalıştığından emin olunuz.

# Homewbrew yükleme

## Ubuntu konsolda aşağıdaki komutu çalıştırınız, Homewbrew’i yükleyin;

/bin/bash -c "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh>)"

## Yükleme bittiğinde aşağıdaki komutları çalıştırınız:

echo 'eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"' >> /home/**<kullanıcı ismi>**/.profile

eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"

## Gerekli ek yükleme için aşağıdaki konuyu çalıştırınız

brew install gcc

# DDEV Yükleme (homebrew vasıtası ile)

## DDEV yüklemesi için aşağıdaki komutu çalıştırınız:

brew install drud/ddev/ddev

## DDEV’in sağlıklı çalışması için son olarak;

sudo apt-get update && sudo apt-get install -y xdg-utils

# Sertifikasyon Yardımcısı Yükleme

## Windows tarafında chocolatey yüklenmesi

Powershellden aşağıdaki komutu çalıştırınız;

Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))

## Mkcert yüklenmesi

choco install -y mkcert

## Sertifika oluşturulması (hem powershell hem de ubuntu konsolda yazılacak)

mkcert -install

## Sertifikanın WSL tarafından tanınmasını sağlamak için

setx CAROOT "$(mkcert -CAROOT)"; If ($Env:WSLENV -notlike "\*CAROOT/up:\*") { setx WSLENV "CAROOT/up:$Env:WSLENV" }.

# Site Klonunun çekilmesi ve Workflow için ayarlanması

Kurulumun ilk aşamalarında çalışmalarımızı workflow’a uyumlu çalışacak şekilde yapıyoruz. Örnek için TMM-DEV-T03 dosyasını referans alabilirsiniz.

# Lokal Wordpress Kurulumu

* Proje dizininde olduğunuzdan emin olun

cd ~/hotcoldshop

## Enviroment dosyası

Bu dosya ddev kurulumunun docker üzerinde oluşturacağı konteynırları bizim kullanabileceğimiz şekilde ayarlar. DDev başlatılmadan evvel bu adım gerçekleştirilmelidir.

Her proje için referans .env dosyası .env.example adı altında bulunmaktadır. Edinmek için;

cp .env. example.env

yada kendiniz oluşturabilirsiniz.

nano .env

### Single Site Enviroment dosyası

DB\_NAME=db  
DB\_USER=db  
DB\_PASSWORD=db  
DB\_HOST=db  
# DB\_PREFIX='wp\_'   
  
WP\_ENV=development  
WP\_HOME= https://br-hcshop.ddev.site  
WP\_SITEURL="${WP\_HOME}/wp"  
WP\_DEBUG\_LOG=/path/to/debug.log

### Multi Site Enviroment dosyası

DB\_NAME=db  
DB\_USER=db  
DB\_PASSWORD=db  
DB\_HOST=db  
DB\_PREFIX='wp\_'  
  
WP\_ENV=development  
WP\_HOME= https://vivit.ddev.site  
WP\_SITEURL="${WP\_HOME}"  
WP\_DEBUG\_LOG=/path/to/debug.log  
DOMAIN\_CURRENT\_SITE='vivit.ddev.site'  
PATH\_CURRENT\_SITE='/'  
SITE\_ID\_CURRENT\_SITE=1  
BLOG\_ID\_CURRENT\_SITE=1

ADMIN\_COOKIE\_PATH','/');

COOKIE\_DOMAIN','');

COOKIEPATH','');

SITECOOKIEPATH','');

## .htaccess oluşturulması

Eğer projedizini/.ddev/confi.yaml dosyanız aşağıdaki gibi ise apache web sunucu aktiftir;

webserver\_type: apache-fpm

Öyle ise uygun bir .htaccess proje\_dizini/www/ klasörü altında bulunmalıdır:

Her proje için referans .htaccess dosyası .htaccess.example adı altında bulunmaktadır. Edinmek için;

cp . htaccess.example . htaccess

yada kendiniz oluşturabilirsiniz: nano . htaccess

### Single site için:

# BEGIN WordPress  
# The directives (lines) between "BEGIN WordPress" and "END WordPress" are  
# dynamically generated, and should only be modified via WordPress filters.  
# Any changes to the directives between these markers will be overwritten.  
<IfModule mod\_rewrite.c>  
RewriteEngine On  
RewriteRule .\* - [E=HTTP\_AUTHORIZATION:%{HTTP:Authorization}]  
RewriteBase /  
RewriteRule ^index\.php$ - [L]  
RewriteCond %{REQUEST\_FILENAME} !-f  
RewriteCond %{REQUEST\_FILENAME} !-d  
RewriteRule . /index.php [L]  
</IfModule>  
# END WordPress

### Multi site için:

# BEGIN WordPress  
# The directives (lines) between "BEGIN WordPress" and "END WordPress" are  
# dynamically generated, and should only be modified via WordPress filters.  
# Any changes to the directives between these markers will be overwritten.  
<IfModule mod\_rewrite.c>  
RewriteEngine On  
RewriteRule .\* - [E=HTTP\_AUTHORIZATION:%{HTTP:Authorization}]  
RewriteBase /  
RewriteRule ^index\.php$ - [L]  
RewriteRule ^([\_0-9a-zA-Z-]+/)?wp-admin$ $1wp-admin/ [R=301,L]  
RewriteCond %{REQUEST\_FILENAME} -f [OR]  
RewriteCond %{REQUEST\_FILENAME} -d  
RewriteRule ^ - [L]  
RewriteRule ^([\_0-9a-zA-Z-]+/)?(wp-(content|admin|includes).\*) wp/$2 [L]  
RewriteRule ^([\_0-9a-zA-Z-]+/)?(.\*\.php)$ wp/$2 [L]  
RewriteRule . /index.php [L]  
</IfModule>  
# END WordPress

## Docker İmajlarının Kurulması

$ ddev start

*Dikkat: Eğer docker ile ilgili hata verirse “docker ps” komutunu çalıştırınız. O da hata verirse Ubuntu konsolonu kapatatınız, docker desktopun açık olduğundan ve ayarlarının doğru olduğundna emin olunuz ve birdaha deneyiniz.*

*Dikkat: DDev varyantında (devilbox, warden gibi) araçların conatinerlerin kapalı durumda olduğunda emin olunuz.*

## Composer ile dependency kurulumu

Gerekli dependencylerin alınması için composer çalıştırın

ddev ssh

composer install

## Wordpress Veritabanı kurulumu

### Yeni Wordpress Veritabanı kurulumu

ddev ssh

wp core install --url=<https://br-hcshop.ddev.site> --title=Hot-Cold-Shop --admin\_user=mehmet.durmaz --admin\_password=Tmm112255 [--admin\_email=mehmet.durmaz@temax.us](mailto:--admin_email=mehmet.durmaz@temax.us)

### Mevcut Veritabanı ile kurulumu

ddev import-db -f<veritabanı yolu ve dosyası>

ör: ddev import-db -f../vivit-21-08-10-13-00.sql.gz (veri tabanının proje dizini değilde bir üst dizinde olması faydalıdır)

ddev ssh

### Veritabanlarındaki değişiklikler

Veritabaında şağıdaki adınları kontrol edip gerekli müdaheleleri yapınız: "/wp/" girdileirni kaldıracağız: [Gokhan Sevinc](http://jira.temax.us:7070/secure/ViewProfile.jspa?name=gokhan.sevinc) uyardı:

a. wp\_options tablosunda;  
siteurl gidisinde /wp/ eki varsa kaldırın yanlızca şöyle gözükmeli https://\{http\_host} örneğin; eğer <http://develop.maxteddy.eu/wp> ise <https://develop.maxteddy.eu> olmalıdır

home gidisinde /wp/ eki varsa kaldırın yanlızca şöyle gözükmeli https://\{http\_host} örneğin; eğer <http://develop.maxteddy.eu/wp> ise <https://develop.maxteddy.eu> olmalıdır

b. wp\_sitemeta tablosunda;  
siteurl gidisinde /wp/ eki varsa kaldırın yanlızca şöyle gözükmeli https://\{http\_host} örneğin; eğer <http://develop.maxteddy.eu/wp> ise <https://develop.maxteddy.eu> olmalıdır

### Komut satırıdan weri tabanının aitliğinin sağlanması

wp search-replace --all-tables <veritabanının çekildiği site domaini> <kurulum yaptığını lokal domain>

ör: wp search-replace --all-tables develop.vivit.bio vivit.ddev.site

#### DİKKAT Multisite ise --network eklemeniz gerekir

## DDev ateşleyelim

ddev launch

## Workflow ile çalışmak

Kurulumdan sonraki ilerleyen aşamalarda çalışmalarımızı workflow’a uyumlu şekilde yapıyoruz. Örnek için TMM-DEV-T03 dosyasını referans alabilirsiniz.

## Multsisite’de Bedrock tabanı problemi

Roots/Bedrock kurulumunda wordpressin çekirdek kurlumu /wp alt dizini içinde olduğundan kurulumda sorun çukabilmektedir.

# EKLER

## Site Back-up’ı gelirse yüklemek için

ddev import-files

## Eğer WP Yüklemesini sil baştan yapmak isterseniz

ddev stop --remove-data

## Eğer containerleri tamamen kapatmak isterseniz

ddev stop --all --stop-ssh-agent

## Nezaman Linux distronuza bir yazılım yüklerseniz peşinden aşağıdaki betikleri çalıştırınız

sudo apt update && sudo apt upgrade

### Homebrew güncelleme kontrollerini alışkanlık haline getiriniz:

brew update && brew upgrade