

Docker で Laravel を使うための導入手順(windows)

はじめに

- ・ 事前に DockerToolBox をインストール
- ・ Docker Quickstart Terminal の使用を進める(現在の git のブランチが常に表示されるため)
- ・ Terminal 上で今いるフォルダより下のフォルダを探す際は ls コマンドが便利
- ・ git add .などを行った後、git commit する前に git status を行くと add したファイルを確認できる。

Docker 導入手順

1. git clone https://github.com/motokuraken/docker_env (C 直下で)
2. docker-compose.yml の volumes(12 行目)を変更、(server から前を変更)
この場合は、/c/docker_env/server:/var/www/html/public
(使用中のパソコンに絶対パスを合わせる)
3. Oracle VM VirtualBox に行き、default の設定を行う
 - (1)設定から共有フォルダを選び、共有フォルダを追加
 - ・ フォルダのパスを選ぶ(今回は C 直下の docker_env)
 - ・ フォルダ名の先頭に c/を付ける(今回の場合は c/docker_env)
 - ・ 自動マウント、および永続化には必ずチェックを入れる
 - (2)共有フォルダの設定が終わったら、一度 VirtualBox を閉じる(電源オフ)
(ここを忘れると、Laravel のインストールができなくなる)
4. コマンドで server に移動し、git clone https://github.com/motokuraken/laravel_env.git
→docker_env の server 下に laravel_env ができていれば OK
できてなければ、git clone のし直し
5. 任意ではあるが、laravel_env を laravel にリネーム(したほうがいいのかも)

6. ターミナルで docker-compose up -d を入力

- ・ 3 の前に行ってもよい
- ・ うまくいった場合は、

```
Creating docker_db          ... done
Creating docker_web         ... done
Creating docker_phpmyadmin ... done
```

のように表示される。

このように表示されていた場合、無事にコンテナが起動している。

- ・ docker-compose ps でコンテナの起動状況を確認できる
(うまくいっていれば Up、エラーだと Exit が表示される)

7. docker-compose exec web bash で web コンテナ内に移動

→root@c55b2c63c592:/var/www/html/public#が表示される

8. cd で laravel_env(リネームした場合はそのフォルダ名)に移動

→ここで、No such file or directory が出た場合は、

- ①docker-compose.yml の 12 行目の volumes の絶対パスが間違っている(3 で設定したやつ)
- ②Oracle VM VirtualBox の共有フォルダの設定ができていない
- ③共有フォルダを設定した後に一度「電源オフ」をしていない

以上 3 つのどれかが原因である。(再起動は忘れがち)

9. 移動したら、ターミナルで composer update を入力(時間がかかります)

→composer update を行うことで Laravel が使えるようになる。

10. exit を入力して web bash から脱出し、docker-compose build で image 作成

11. 再度、docker-compose up -d を実行

→6 同様、web、db、phpmyadmin の 3 つとも done になれば OK
(docker-compose ps で確認してみるのもよい)

12. ブラウザに行き、自分の IP アドレスの後ろに「:70」を付けて URL を入力

→おなじみの Laravel の画面が出れば OK

・例：111.111.91.100()

・ Docker Quickstart Terminal を使っている場合、

```
## .
## ## ## ==
## ## ## ## ===
/"/""""""""""""""""""""¥_/ ===
~~~ { ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ / === - ~ ~ ~
¥_____ o _____/
¥ ¥ _____/
¥_____¥_____/
```

起動したときに出る↑のクジラの下の方の IP の後の数字が自分の IP

docker is configured to use the default machine with IP ここが自分の IP

13. 同じく自分の IP アドレスの後に「:7070」を付けて URL を入力

→phpmyadmin が表示されれば OK

14. 「自分の IP アドレス:70/test」と URL を入力

→画面に「test」と表示されれば画面遷移の問題は OK

15. phpmyadmin に docker という名前で DB を作成

16. docker-compose exec web bash で web bash に入り、php artisan migrate 実行

→migration が上手くいった場合、DB との連携は OK

・ DB 設定の方法

config/database.php

51 行目 'database' => env('DB_DATABASE', 'docker')←docker 部分を DB 名に

.env

12 行目 DB_DATABASE=docker docker 部分を DB 名に