

Apple Silicon Mac 時代の PHP 開発環境構築 2021

2021/12/21 @shin1x1

@shin1x1

- 新原（しんばら） 雅司
- 1 × 1 株式会社
 - Web アプリケーション開発
 - 技術サポート
- PHP の現場
 - <https://php-genba.shin1x1.com/>

M1 Mac を買いました！

macOS Monterey

バージョン 12.1

MacBook Pro (14インチ、2021)

チップ Apple M1 Pro

メモリ 32 GB

PHP 開発環境

- Mac アプリ
- Vagrant
- Docker

Mac アプリ

Mac アプリの種類

- Intel Mac 用 (x86_64)
- Apple Silicon Mac(M1 Mac) 用 (aarch64)
- Universal
 - x86_64 と aarch64 アプリが同梱

いずれも Apple Silicon Mac で動作

- x86_64 は Rosseta 2 により aarch64 に自動変換して動作

file コマンドで確認

- Apple Silicon Mac 用 (aarch64)

```
$ file Docker
Docker: Mach-0 64-bit executable arm64
```

- Universal

```
$ file iTerm2
iTerm2 (for architecture x86_64): Mach-0 64-bit executable x86_64
iTerm2 (for architecture arm64): Mach-0 64-bit executable arm64
```

- Intel Mac 用 (x86_64)

```
$ file Dropbox
Dropbox: Mach-0 64-bit executable x86_64
```

主なアプリ対応状況

- aarch64 / x86_64
 - Docker Desktop
- Universal (これが多い)
 - PhpStorm
 - Visual Studio Code
 - iTerm 2
- x86_64 のみ
 - Evernote
 - Dropbox

Homebrew

- Apple Silicon Mac(3.0.0) + Monterey(3.3.0) 対応済
- aarch64 アプリは、`/opt/homebrew` にインストールされる
 - 環境変数 PATH に `/opt/homebrew/bin` `/opt/homebrew/sbin` を追加
- x86_64 アプリは、`/usr/local`

Apple Silicon Mac + Mac アプリ

- 全く問題無い！
- 2021/12 時点で多くのアプリで Apple Silicon 対応が行われている
- Universal 版が多い印象
 - ダウンロード選択肢が無いからといって Apple Silicon 未対応ではない
- x86_64 アプリのみでも、Rosetta 2 で普通に動作する
 - 今のところアプリで困ったことは無い

仮想環境 - Vagrant

Apple Silicon Mac + Vagrant + VirtualBox

- Vagrant は動作する (x86_64)
- 現時点では VirtualBox は Apple Silicon Mac で動作しない
 - Apple Silicon サポートについてオフィシャルには情報が無い

Apple Silicon Mac + Vagrant + VMWare Fusion

- VMware Fusion for Apple silicon Public Tech Preview
 - <https://blogs.vmware.com/teamfusion/2021/09/fusion-for-m1-public-tech-preview-now-available.html>
- Vagrant を使いたいなら選択肢の一つ
- 有償

Apple Silicon Mac + Vagant

- VirtualBox が動作しないので厳しい
- Apple Silicon Mac 以前に Docker 環境を利用していたので、個人的にはモチベーションはあまりない
- このタイミングで Docker に移行するプロジェクトもいくつかある

仮想環境 - Docker

Apple Silicon Mac + Docker

- Docker Desktop アプリは Apple Silicon 対応済
- Docker イメージにも種類がある
 - **linux/amd64**: Intel Mac で利用するイメージ
 - **linux/arm64**: Apple Silicon Mac で利用するイメージ
- Docker Desktop では **amd64** イメージも動作する
 - QEMU によるエミュレート
 - 動作は遅くなったり、挙動が不安になるケースも
 - 特に JVM をランタイムとしたイメージは要注意 (**arm64** は問題無し)

Docker イメージの対応状況

- オフィシャルイメージは **amd64** / **arm64** 両対応が多い
 - php, postgres, redis, nginx, composer 等
 - mysql は **amd64** のみ
 - mysql/mysql の 8.0 は **arm64** イメージあり
- 独自ビルドイメージは **amd64** のみが多い傾向
 - これまで amd64 のみ利用する場面が多かったので

docker コマンド, Dockerfile, docker-compose.yml

- docker コマンド操作は基本そのまま
- Dockerfile や docker-compose.yml はそのまま OK
 - CPU アーキテクチャ固有の情報以外
- Docker イメージ
 - **amd64/arm64** : Intel=**amd64** / Apple Silicon=**arm64** が動く
 - **amd64** のみ: Intel=**amd64** / Apple Silicon=**amd64** が動く
 - いかにかに 両サポートイメージを作るかがポイント

Intel Mac と Apple Silicon Mac との共存

- 案1. Mac で利用するイメージは、自身の Mac でビルド
 - Intel Mac では **amd64**、Apple Silicon Mac では **arm64** をビルド

```
services:  
  php:  
    build: ./docker/php # docker/php/Dockerfile をビルド
```

- 案2. Docker Buildx でマルチアーキテクチャビルドして共有
 - GitHub Actions 上でのマルチアーキテクチャビルド例
 - <https://blog.shin1x1.com/entry/docker-multi-arch-build-on-github-actions>

amd64/arm64 イメージ速度差

- M1 MacBook Pro では **arm64** イメージの方が速い
- phpunit 実行時間 (php + postgres)
 - M1 MBP (php: **arm64**): 2.36 minutes
 - M1 MBP (php: **amd64**): 4.90 minutes
 - [参考] Intel MBP (php: **amd64**) : 2.23 minutes

arm64 で発生: postgres イメージ接続遅延

- php([arm64](#)) + postgres([arm64](#)) で発生
- 時折、PHP から PostgreSQL への接続に 5 秒かかる現象
 - pg_connect / PDO とともに
- `php:X-apache-buster` ベースで発生
- `php:X-apache-bullseys` や `amazonlinux:2` では発生せず

Apple Silicon Mac + Docker

- 基本は **arm64** イメージを使えば問題無し
- **amd64** はうまく動かない場合がある (ex. JVM 系)
- **amd64** と **arm64** 両イメージを用意しておくスムーズ
 - チームで Intel Mac と Apple Silicon Mac が混在

Apple Silicon Mac 時代の PHP 開発環境 2021

- Mac アプリ + Docker で構成
- マルチアーキテクチャ対応の Docker イメージを用意するのがポイント
- Mac だけではなく、本番環境含めたマルチアーキテクチャ時代に