

to production

Browse files

master

Ruslan authored and Ruslan committed 4 minutes ago 1 parent 55e804f commit 0c5c5a3a0cece19b0325336c4cf34d328a8fbd66

Showing 10 changed files with 292 additions and 14 deletions.

Unified Split

BIN +0 Bytes (100%) .DS\_Store

Binary file not shown.

BIN +191 KB Illustrations/IMPROVE\_docker\_laravel\_commit.pdf

Binary file not shown.

6 laravel/.dockerignore

@@ -0,0 +1,6 @@

```
1 + vendor/
2 + node_modules/
3 + tests/
4 + .git/
5 + .idea/
6 + .env
```

36 laravel/docker-compose.yml

```
3
4 services:
5   php:
6 -   build:
7 -     context: ./docker # look for Dockerfile in this directory
8 -     # dockerfile: AnotherNameDockerFile # This is the alternative
9 -     container_name: customphp
10
11 volumes:
```

```
3
4 services:
5   php:
6 +   # -- FOR PROD --
7 +   image: atabegruslan/docker4laravel-php:0.0.1
8 +   environment:
9 +     # APP_DEBUG: "true"
10 +     APP_KEY: "base64:SWXterQT+3H2DQNwvOS+k00haIccLOfd8+z+P2qId3c="
11 volumes:
12 +   - ./.env:/var/www/html/.env
13 +   # -- FOR LOCAL --
14 +   # build:
15 +   #   context: ./docker # look for Dockerfile in this directory
```

		16	+	# # dockerfile: AnotherNameDockerFile # This is the alternative
		17	+	# container_name: customphp
		18	+	# volumes:
		19	+	# - ./var/www/html
12		20		
13	- web: # web talks to php internally thru docker networks	21	+	web:
14	- image: nginx:1.17	22	+	# -- FOR PROD --
15	- working_dir: /var/www/html	23	+	image: atabegruslan/docker4laravel-nginx:0.0.1
16	- volumes:	24	+	# -- FOR LOCAL --
17	- - ./var/www/html	25	+	# image: nginx:1.17
18	- - ./docker/nginx_template_local.conf:/etc/nginx/conf.d/default.conf # Recommended example template: https://laravel.com/docs/7.x/deployment#nginx	26	+	# working_dir: /var/www/html
		27	+	# volumes:
		28	+	# - ./var/www/html
		29	+	# - ./docker/nginx_template_local.conf:/etc/nginx/conf.d/default.conf # Recommended example template: https://laravel.com/docs/7.x/deployment#nginx
19	ports:	30		ports:
20	- - "80:80" # outside port:container port. In production, use the standard port 80 for outside port	31	+	- "80:80"
21		32		
22	- # Need to run this first: `docker-compose run --rm node npm install`			
23	node:	33		node:
24	image: node:12	34		image: node:12
25	user: node # Node image already have an user called node	35		user: node # Node image already have an user called node

▼ 18 ■■■■■ laravel/docker/nginx_prod.Dockerfile 📄				
... @@ -0,0 +1,18 @@				
		1	+	# NPM dependencies.
		2	+	FROM node:12 AS npm-build
		3	+	
		4	+	WORKDIR /var/www/html
		5	+	
		6	+	COPY package.json package-lock.json webpack.mix.js /var/www/html/
		7	+	COPY resources /var/www/html/resources/
		8	+	COPY public /var/www/html/public/
		9	+	
		10	+	RUN npm ci
		11	+	RUN npm run production
		12	+	
		13	+	# Nginx production.
		14	+	FROM nginx:1.17
		15	+	
		16	+	COPY ./docker/nginx_template_prod.conf /etc/nginx/conf.d/default.conf
		17	+	COPY --chown=www-data --from=npm-build /var/www/html/public/ /var/www/html/public/
		18	+	COPY --chown=www-data . /var/www/html 🚫

▼ 36 ■■■■■ laravel/docker/nginx_template_prod.conf 📄				

... @@ -0,0 +1,36 @@

1

+ server {

2

+ listen 80;

3

+ server\_name \_;

4

+ root /var/www/html/public;

5

+

6

+ add\_header X-Frame-Options "SAMEORIGIN";

7

+ add\_header X-XSS-Protection "1; mode=block";

8

+ add\_header X-Content-Type-Options "nosniff";

9

+

10

+ index index.html index.htm index.php;

11

+

12

+ charset utf-8;

13

+

14

+ access\_log /dev/stdout;

15

+ error\_log /dev/stdout;

16

+

17

+ location / {

18

+ try\_files \$uri \$uri/ /index.php?\$query\_string;

19

+ }

20

+

21

+ location = /favicon.ico { access\_log off; log\_not\_found off; }

22

+ location = /robots.txt { access\_log off; log\_not\_found off; }

23

+

24

+ error\_page 404 /index.php;

25

+

26

+ location ~ \.php\$ {

27

+ fastcgi\_pass 127.0.0.1:9000;

28

+ fastcgi\_index index.php;

29

+ fastcgi\_param SCRIPT\_FILENAME \$realpath\_root\$fastcgi\_script\_name;

30

+ include fastcgi\_params;

31

+ }

32

+

33

+ location ~ /\.(?!well-known).\* {

34

+ deny all;

35

+ }

36

+ }

▼ 18 ■■■■■ laravel/docker/opcache.ini 📄

... @@ -0,0 +1,18 @@

1

+ [opcache]

2

+ opcache.enable=1

3

+ ; 0 means it will check on every request

4

+ ; 0 is irrelevant if opcache.validate\_timestamps=0 which is desirable in production

5

+ opcache.revalidate\_freq=0

6

+

7

+ ; Validate\_timestamps=0 means just cache the interpreted byte code, instead of

+ interpret the php upon every request

```
8 + opcache.validate_timestamps=0
9 +
10 + ; Limit max number of files. Can be checked by `find . -iname *.php | wc -l`
11 + opcache.max_accelerated_files=10000
12 +
13 + ; Can be checked by observing php container's MEM USAGE by `docker stats` while the php
    container is in action, eg `ab -n 500 -c 100 http://domain/someLaravelRoute` (giving the
    Laravel app some work: 500 requests, 100 requests in each go concurrently)
14 + opcache.memory_consumption=128
15 +
16 + opcache.max_wasted_percentage=10
17 + opcache.interned_strings_buffer=16
18 + opcache.fast_shutdown=1
```

48 laravel/docker/php\_prod.Dockerfile

@@ -0,0 +1,48 @@

```
1 + # Composer dependencies.
2 + FROM composer AS composer-build
3 +
4 + WORKDIR /var/www/html
5 +
6 + COPY composer.json composer.lock /var/www/html/
7 +
8 + RUN mkdir -p /var/www/html/database/{factories,seeds} \
9 +   && composer install --no-dev --prefer-dist --no-scripts --no-autoloader --no-progress
    --ignore-platform-reqs
10 +
11 + # NPM dependencies.
12 + FROM node:12 AS npm-build
13 +
14 + WORKDIR /var/www/html
15 +
16 + COPY package.json package-lock.json webpack.mix.js /var/www/html/
17 + COPY resources /var/www/html/resources/
18 + COPY public /var/www/html/public/
19 +
20 + RUN npm ci
21 + RUN npm run production
22 +
23 + # Actual production image.
24 + FROM php:7.4-fpm
25 +
26 + WORKDIR /var/www/html
27 +
28 + RUN apt-get update \
29 +   && apt-get install --quiet --yes --no-install-recommends \
30 +     libzip-dev \
31 +     unzip \
```

```
32 + && docker-php-ext-install pcntl opcache zip pdo pdo_mysql \  
33 + && pecl install -o -f redis-5.1.1 \  
34 + && docker-php-ext-enable redis  
35 +  
36 + # Use the default production configuration  
37 + RUN mv $PHP_INI_DIR/php.ini-production $PHP_INI_DIR/php.ini  
38 +  
39 + # Override with custom opcache settings  
40 + COPY ./docker/opcache.ini $PHP_INI_DIR/conf.d/  
41 +  
42 + COPY --from=composer /usr/bin/composer /usr/bin/composer  
43 +  
44 + COPY --chown=www-data --from=composer-build /var/www/html/vendor/ /var/www/html/vendor/  
45 + COPY --chown=www-data --from=npn-build /var/www/html/public/ /var/www/html/public/  
46 + COPY --chown=www-data . /var/www/html  
47 +  
48 + RUN composer dump -o && composer check-platform-reqs
```

131 laravel/good_notes.md			
76	- Use the php image's `docker-php-ext-enable` helper to enable the Redis dependency.	76	- Use the php image's `docker-php-ext-enable` helper to enable the Redis dependency.
77	- `docker-php-ext-install` installs and enables, but pecl only installs without enabling.	77	- `docker-php-ext-install` installs and enables, but pecl only installs without enabling.
78		78	
79	- - <u>Install composer using a multi-stage image.</u> Copy this binary: `/usr/bin/composer` from this composer image: https://hub.docker.com/_/composer to this place in our container: `/usr/bin/`	79	+ - Copy this binary: `/usr/bin/composer` from this composer image: https://hub.docker.com/_/composer to this place in our container: `/usr/bin/`
80	- https://dev.to/jonesrussell/install-composer-in-custom-docker-image-3f71	80	- https://dev.to/jonesrussell/install-composer-in-custom-docker-image-3f71
81		81	
82	...	82	...
108	user: "\${UID}:\${GID}"	108	user: "\${UID}:\${GID}"
109	...	109	...
110	- https://stackoverflow.com/questions/56844746/how-to-set-uid-and-gid-in-docker-compose	110	- https://stackoverflow.com/questions/56844746/how-to-set-uid-and-gid-in-docker-compose
		111	+
		112	+ Rest of the major changes:
		113	+
		114	+ https://github.com/Ruslan-Aliyev/Docker/blob/master/Illustrations/IMPROVE_docker_laravel_commit.pdf
		115	+
		116	+ ## Tutorial 3 - To Production
		117	+
		118	+ In production
		119	+ - Better not to map volumes, so the image should have everything it need within it (except some private things)
		120	+ - Change build steps for dependencies
		121	+ - Better to use https://hub.docker.com/_/php 's ` \$PHP_INI_DIR/php.ini-production `, which in turn needs OPcache for custom configs.
		122	+ - https://www.scalingphpbook.com/
		123	