

2	APP_ENV=local	2	APP_ENV=local
3	APP_KEY=	3	APP_KEY=
4	APP_DEBUG=true	4	APP_DEBUG=true
5	APP_URL=http://localhost	5	APP_URL=http://localhost
6		6	
7	LOG_CHANNEL=stack	7	LOG_CHANNEL=stack
8	LOG_LEVEL=debug	8	LOG_LEVEL=debug
9		9	
10	DB_CONNECTION=mysql	10	DB_CONNECTION=mysql
11	DB_HOST=db	11	DB_HOST=db
12	DB_PORT=3306	12	DB_PORT=3306
13	DB_DATABASE=laraapp_db	13	DB_DATABASE=laraapp_db
14	DB_USERNAME=root	14	DB_USERNAME=root
15	DB_PASSWORD=	15	DB_PASSWORD=
16		16	
17	BROADCAST_DRIVER=log	17	BROADCAST_DRIVER=log
18	CACHE_DRIVER=file	18	CACHE_DRIVER=file
19	QUEUE_CONNECTION=sync	19	QUEUE_CONNECTION=sync
20	SESSION_DRIVER=file	20	SESSION_DRIVER=file
21	SESSION_LIFETIME=120	21	SESSION_LIFETIME=120
22		22	
23	MEMCACHED_HOST=127.0.0.1	23	MEMCACHED_HOST=127.0.0.1
24		24	
25	- REDIS_HOST=127.0.0.1	25	+ REDIS_HOST=redis
26	REDIS_PASSWORD=null	26	REDIS_PASSWORD=null
27	REDIS_PORT=6379	27	REDIS_PORT=6379
28		28	

▼ 103 ■■■■■ laravel/docker-compose.yml 📄			
...	@@ -1,23 +1,92 @@		
1	- version: '3.7'	1	+ version: '3'
2	- services:	2	+ # Version 3 is the standard (of 2019) that every docker-compose should have
3	- app:	3	+
		4	+ services:
		5	+ php:
4	build:	6	build:
5	- context: .	7	+ context: ./docker # look for Dockerfile in this directory
6	- dockerfile: docker/Dockerfile	8	+ # dockerfile: AnotherNameDockerFile # This is the alternative
7	- image: 'laravelapp'	9	+ container_name: customphp
8	- container_name: laraapp	10	+ volumes:
9	- ports:	11	+ - ./:/var/www/html
10	- - 8080:80	12	+
		13	+ web: # web talks to php internally thru docker networks
		14	+ image: nginx:1.17
		15	+ working_dir: /var/www/html
11	volumes:	16	volumes:
12	- ./:/var/www/html	17	- ./:/var/www/html
		18	+ - ./docker/nginx_template_local.conf:/etc/nginx/conf.d/default.conf # Recommended example template: https://laravel.com/docs/7.x/deployment#nginx

		<div>19 + ports: 20 + - "80:80" # outside port:container port. In production, use the standard port 80 for outside port 21 + 22 + # Need to run this first: `docker-compose run --rm node npm install` 23 + node: 24 + image: node:12 25 + user: node # Node image already have an user called node 26 + working_dir: /assets 27 + volumes: 28 + - ./assets 29 + command: npm run watch # Default command if nothing is specified 30 +</div>
13	db:	31 db:
14 - image: mysql:5.7		32 + image: mysql:8
15 - restart: always		33 + restart: unless-stopped # By default, a container does not restart under any circumstance.
16 - ports:		34 + tty: true # enable shell-like interface to interact with this container
17 - - "3307:3306"		
18 environment:		35 environment:
19 - MYSQL_DATABASE: 'laraapp_db'		36 + MYSQL_ALLOW_EMPTY_PASSWORD: "yes" # Allow connecting to it with empty passwords
20 - MYSQL_ALLOW_EMPTY_PASSWORD: 1		37 + MYSQL_ROOT_HOST: "%" # Allow connections from any host
21 - MYSQL_ROOT_PASSWORD: ""		38 + MYSQL_ROOT_PASSWORD: "\${DB_PASSWORD}" # Ensure these match .env
		39 + MYSQL_DATABASE: "\${DB_DATABASE}"
		40 + # MYSQL_USER: "\${DB_USERNAME}" # <a href="https://stackoverflow.com/questions/66831863/mysql-docker-container-keeps-restarting/66910240#66910240">https://stackoverflow.com/questions/66831863/mysql-docker-container-keeps-restarting/66910240#66910240</a>
		41 + MYSQL_PASSWORD: "\${DB_PASSWORD}"
		42 + volumes:
		43 + - mysqldata:/var/lib/mysql # Persist data, as containers don't keep data when restarted.
		44 +
		45 + redis:
		46 + image: redis:5
22 volumes:		47 volumes:
23 - - ./db:/var/lib/mysql ⊖		48 + - redisdata:/data
		49 +
		50 + phpmyadmin:
		51 + image: phpmyadmin/phpmyadmin
		52 + restart: always
		53 + container_name: phpmyadmin
		54 + depends_on:
		55 + - db
		56 + ports:
		57 + - "8089:80"
		58 + environment:
		59 + PMA_HOST: db
		60 + MYSQL_ROOT_PASSWORD: "\${DB_PASSWORD}"
		61 +

```
62 + # composer:
63 + #   image: composer:latest
64 + #   container_name: composer
65 + #   volumes:
66 + #     - ./:/var/www/html
67 + #   working_dir: /var/www/html
68 + # With the above composer container: `docker-compose run --rm composer require
  blah/blah`
69 + # Or do like this: https://www.codegrepper.com/code-
  examples/shell/Composer+install+with+dockerfile
70 +
71 + # artisan:
72 + #   build:
73 + #     context: ./docker
74 + #     dockerfile: ./docker/Dockerfile
75 + #   container_name: artisan
76 + #   volumes:
77 + #     - ./:/var/www/html
78 + #   depends_on:
79 + #     - mysql # else migrations wont happen properly
80 + #   working_dir: /var/www/html
81 + #   entrypoint: ['php', '/var/www/html/artisan'] # Runs `php /var/www/html/artisan`
  instead of `artisan` when `docker-compose run artisan` is called.
82 +
83 + # mailhog:
84 + #   image: mailhog/mailhog:latest
85 + #   container_name: mailhog
86 + #   ports:
87 + #     - 1025:1025
88 + #     - 8025:8025
89 +
90 + volumes: # Named volumes
91 +   mysqldata:
92 +   redisdata: ⊖
```

▼ 33 ■■■ laravel/docker/Dockerfile 📄

...	@@ -1,23 +1,20 @@	1	+ FROM php:7.4-fpm
1	- FROM php:7.4.1-apache	2	
2		3	WORKDIR /var/www/html
3	WORKDIR /var/www/html	4	
4		5	+ RUN apt-get update \
5	- RUN apt-get update && apt-get install -y \	6	+ && apt-get install --quiet --yes --no-install-recommends \
6	-     libpng-dev \	7	+     libzip-dev \
7	-     zlib1g-dev \	8	+     unzip \
8	-     libxml2-dev \	9	+     && docker-php-ext-install pcntl zip pdo pdo_mysql \
9	-     libzip-dev \	10	+     && pecl install -o -f redis-5.1.1 \
10	-     libonig-dev \	11	+     && docker-php-ext-enable redis
11	-     zip \		

12	- curl \		
13	- unzip \		
14	- && docker-php-ext-configure gd \		
15	- && docker-php-ext-install -j\$(nproc) gd \		
16	- && docker-php-ext-install pdo_mysql \		
17	- && docker-php-ext-install mysqli \		
18	- && docker-php-ext-install zip \		
19	- && docker-php-source delete		
20		12	
21	- COPY docker/vhost.conf /etc/apache2/sites-available/000-default.conf	13	+ COPY --from=composer /usr/bin/composer /usr/bin/composer
22		14	
23	- RUN curl -sS https://getcomposer.org/installer   php -- --install-dir=/usr/local/bin --filename=composer	15	+ RUN groupadd --gid 1000 appuser \
		16	+ && useradd --uid 1000 -g appuser \
		17	+ -G www-data,root --shell /bin/bash \
		18	+ --create-home appuser
		19	+
		20	+ USER appuser ⊖

▼ 37 ■■■■■ laravel/docker/nginx_template_local.conf 📄			
...	@@ -0,0 +1,37 @@		
		1	+ server {
		2	+ listen 80;
		3	+ server_name _; # Can be anything
		4	+ root /var/www/html/public; # Have correct folder here
		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	+ # Need to redirect logs so that we can see the logs
		15	+ access_log /dev/stdout;
		16	+ error_log /dev/stdout;
		17	+
		18	+ location / {
		19	+ try_files \$uri \$uri/ /index.php?\$query_string;
		20	+ }
		21	+
		22	+ location = /favicon.ico { access_log off; log_not_found off; }
		23	+ location = /robots.txt { access_log off; log_not_found off; }
		24	+
		25	+ error_page 404 /index.php;
		26	+
		27	+ location ~ \.php\$ {

```
28 +         fastcgi_pass php:9000; # Define how nginx container is talking to php container
29 +         fastcgi_index index.php;
30 +         fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
31 +         include fastcgi_params;
32 +     }
33 +
34 +     location ~ /\.(!well-known).* {
35 +         deny all;
36 +     }
37 + } ❌
```

▼ 11 ■■■■■ laravel/docker/vhost.conf 📄



Load diff  
This file was deleted.

▼ 110 ■■■■■ laravel/good\_notes.md 📄

... @@ -0,0 +1,110 @@

```
1 + # Very good tutorial's notes
2 +
3 + https://www.youtube.com/playlist?list=PL1LQwTE3lBhSnaL7j90AUJyvC9mFCKhZm
4 +
5 + ## Tutorial 1 - Install Laravel from a composer container
6 +
7 + ### Basic
8 +
9 + 1. Getting composer image (https://hub.docker.com/_/composer) - most simple: `docker
  run composer`
10 + 2. Delete containwr when no longer needed: `docker run --rm composer`
11 + 3. Map volume: `docker run --rm -v $(pwd):/app composer`
12 + 4. Set the working directory in the container: `docker run --rm -v $(pwd):/app -w /app
  composer`. Any commands you pass into the container will be run from the `/app`
  directory.
13 + 5. Install Laravel: `docker run --rm -v $(pwd):/app -w /app composer composer create-
  project laravel/laravel composer4laravel`
14 + 6. Use your user instead of `root`: `docker run -u $(UID):$(UID) --rm -v $(pwd):/app -
  w /app composer composer create-project laravel/laravel composer4laravel`
15 +
16 + This is slow because composer is downloading each dependency from scratch, in this
  newly created filesystem, without cache.
17 +
18 + ### Fast
19 +
20 + https://hub.docker.com/r/clevyr/prestissimo install Composer dependencies in parallel
```