

in WordPress plugin and theme development

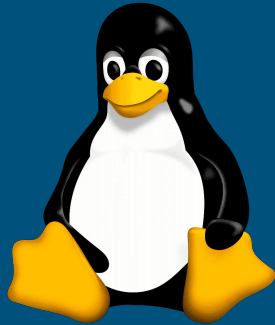
Rade Jekić

@rjelic,
@drummerche

What is Docker?

“Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.”

Docker is not a virtualization software.



Important Docker related terms

- image
- container
- docker file
- docker-compose
- docker-compose.yml
- writable container layer
- volumes
 - named
 - path based

Image

“A Docker image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.”

Docker images can be built from a local code, or downloaded from the Docker image hub (local or on the Internet).

Official download location - <https://hub.docker.com/>

Container

When a user runs an image using Docker Engine, it becomes a container.

Docker file

Text file containing instructions for building Docker image. The Docker image is built using the following command:

```
docker build -t image_name:tag .
```

docker-compose

“Docker Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration.”

docker-compose.yml

Text file containing instructions for creating and maintaining container services (applications). Instead of starting every single container in a separate command or one long terminal command, only one short command needs to be issued in the folder where docker-compose.yml file is located, and that command is:

```
docker-compose up
```

That's it! You've successfully started your application.

Writable container layer

“An image is made of a stack of immutable or read-only layers. When the Docker engine creates a container from such an image, it adds a writable container layer on top of this stack of immutable layers.”

- Size of this layer directly interferes with the size of the container itself
- This kind of container storage “lives” as long as the container
- Upon destroying the container this storage is also destroyed/deleted

Volumes

“Volumes are the preferred mechanism for persisting data generated by and used by Docker containers.”

- All data “lives” inside the volume, and doesn’t interfere with the size of the container itself.
- If the container is destroyed and deleted, the volumes stay intact - they have to be manually deleted.
- Useful for storing data that have to be changed more often
- Volumes can be shared between container instances
- Useful for storing db data and wp-content folder

Volumes

There are two main types of volumes:

- Named volumes
- Path based volumes

Named volumes

Named volumes look like this:

```
db_data:/var/lib/mysql
```

If you're using Docker Compose, it will automatically create the volume for you when you first do a

```
docker-compose up
```

When you use a volume like this, Docker will manage the volume for you. On Linux, that volume will get saved to

```
/var/lib/docker/volumes/db_data/_data
```

Path based volumes

Path based volumes serve the same purpose as named volumes, except you're responsible for managing where the volume gets saved on the Docker host.

For example:

```
./:/var/www/html/wp-content
```

will mount contents of the current folder to the `/var/www/html/wp-content` in the container.

Important note: deleting the file inside current folder deletes the file in the mounted folder inside container!

Pros

- Small footprint (CPU, RAM, and disk) compared to similar solutions
- Easy to maintain (backup, restore, update)
- Fast
- Scalable
- Easy to learn after you get to know basic concepts and terms

Cons

- Usage of terminal
- Docker file and docker-compose.yml both have formatting rules that must be applied

Main Alternatives



XAMPP



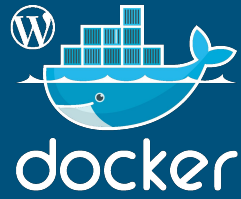
MAMP



VAGRANT



LOCAL



SETUP AND USAGE

Wordpress Docker container

- Wordpress image
- Database engine image
 - mysql
 - mariadb
- Server engine image
 - apache
 - nginx

Basic usage

1. Create a folder that will host your Wordpress project files.
2. Open the newly created folder.
3. Open the terminal (console) inside that folder.
4. Copy and paste this line of code inside terminal (the dot in the end is very important. Also, notice the space between dot and “git”)

```
git clone https://github.com/rjekic/docker-wordpress.git .
```

5. Voilà! You have everything you need to start using Docker Wordpress development environment :)

Basic usage (continued)

6. Copy and paste this line of code in the terminal

```
docker-compose up -d
```

7. That's it! Grab a cup of coffee and wait for the process to finish.
8. After the process is finished, open the browser and copy this line of code in the address bar

```
localhost
```

7. Press ENTER and you will see the Wordpress installation screen
8. You can manage from here on your own, right? :)

Other useful docker-compose commands

```
docker-compose stop # Stops all that is started using docker-compose.yml from the current dir
```

```
docker-compose down # Stops all that is started using docker-compose.yml from the current dir.  
Also, this command removes all docker containers listed in docker-compose.yml file
```

```
docker-compose ps # List all running containers
```

Useful links

- <https://docs.docker.com>
- <https://hub.docker.com>
- <https://nickjanetakis.com/blog/tag/docker-tips-tricks-and-tutorials>
- Everything that you can find related to Docker :)



That's all Folks!