

PROJECT REPORT

FY 2023



docker
Compose

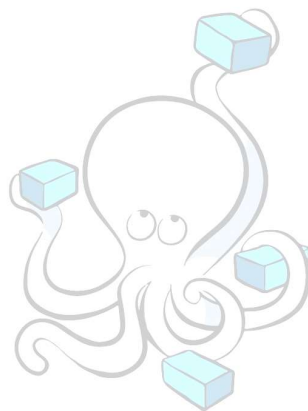
Docker Compose to easily run WordPress in an isolated environment built with Docker containers. This report guides to demonstrate how to use Compose to set up and run WordPress.

Tushar Chauhan

Trainer : Siddharth Sir

Contents

About Docker & its importance	1
Docker Compose & yaml files	2
About wordpress	3
Method to perform task	4
Code beakdown	10



docker
Compose

"You can use Docker playground to learn and do experiments in Docker environment"

DOCKER & IT'S IMPORTANCE

What is Docker?

Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called Containers. The service has both free and premium tiers. The software that hosts the containers is called Docker Engine. It was first started in 2013 and is developed by Docker, Inc.

Docker is a tool that is used to automate the deployment of applications in lightweight containers so that applications can work efficiently in different environments.

Operations

Docker can package an application and its dependencies in a virtual container that can run on any Linux, Windows, or macOS computer. This enables the application to run in a variety of locations, such as on-premises, in public (see decentralized computing, distributed computing, and cloud computing) or private cloud. When running on Linux, Docker uses the resource isolation features of the Linux kernel (such as cgroups and kernel namespaces) and a union-capable file system (such as OverlayFS) to allow containers to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines. Docker on macOS uses a Linux virtual machine to run the containers.

Importance

Docker is an essential tool for developers of any technology as it **makes it easier to develop, ship, and run applications by using containers**. Containers are small (megabytes), start instantly, have built-in mechanisms for versioning and component reuse and can be easily shared via the public Docker Hub or private repository. Additionally, Docker provides encapsulation, isolation, portability and control which are all essential benefits for developers.

Docker compose & yaml/yml files

YAML/YML FILES

YAML is a human-readable data serialization language that is often used for writing configuration files. Depending on whom you ask, YAML stands for yet another markup language or YAML ain't markup language (a recursive acronym), which emphasizes that YAML is for data, not documents.

YAML is a popular programming language because it is designed to be easy to read and understand. It can also be used in conjunction with other programming languages. Because of its flexibility and accessibility, YAML is used by the Ansible automation tool to create automation processes, in the form of Ansible Playbooks.

YAML files use a “.yml” or “.yaml” extension, and follow specific syntax rules.

YAML has features that come from Perl, C, XML, HTML, and other programming languages. YAML is also a superset of JSON, so JSON files are valid in YAML.

Docker Compose

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.

Compose works in all environments: production, staging, development, testing, as well as CI workflows. It also has commands for managing the whole lifecycle of your application:

- Start, stop, and rebuild services
- View the status of running services
- Stream the log output of running services
- Run a one-off command on a service

WordPress


What is WordPress?

WordPress was initially a simple blog platform, however it soon developed into a more feature-rich Content Management System (CMS).

The software is written in the PHP scripting language. All website content, including settings, blog posts and pages, are stored in a database. MySQL is the most popular choice of database, but some website owners choose to use an alternative solution such as MariaDB or SQLite.

People use WordPress to create sites for the following purposes:

- Personal blogs
- Corporate blogs
- Portfolios
- Business sites
- Ecommerce sites
- Newsletters
- Podcasts
- Landing pages
- Video website
- School websites
- Etc.



docker
Compose

Method to perform task

TASK:

Using Docker compose to make a WordPress environment with MariaDB.

1. Open your window terminal (make sure Docker is installed first) or you can simply just try this in Docker playground.
2. As I have used the Docker playground for this report so I will be using linux commands.
3. Make a directory in your machine with 'mkdir' command as shown in picture below.

```
#####  
# WARNING!!!! #  
# This is a sandbox environment. Using personal credentials #  
# is HIGHLY! discouraged. Any consequences of doing so are #  
# completely the user's responsibilities. #  
# #  
# The PWD team. #  
#####  
[node1] (local) root@192.168.0.28 ~  
$ ls  
[node1] (local) root@192.168.0.28 ~  
$ mkdir wordpress-docker  
[node1] (local) root@192.168.0.28 ~  
$ ls  
wordpress-docker  
[node1] (local) root@192.168.0.28 ~  
$ cd wordpress-docker  
[node1] (local) root@192.168.0.28 ~/wordpress-docker  
$
```

4. Now change directory to that folder by "cd wordpress-docker" in my case.
5. Make a file inside the folder you are in with "touch" command so we can edit it afterwards.
6. Make sure the file have an extension name of ".yaml" or ".yml"

```
[node1] (local) root@192.168.0.28 ~/wordpress-docker  
$  
[node1] (local) root@192.168.0.28 ~/wordpress-docker  
$ touch docker-compose.yaml
```

7. You can input the code into the file through 2 methods whether you can open the file through vim editor.

```
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$ vim docker-compose.yml
```

8. Or if you are working on Docker playground as me you can just click on editor button.

The screenshot shows the Docker Playground interface for an instance named `cgijg50s_cgijoaee69v000cl4i8g`. The instance details include IP `192.168.0.28`, Memory `33.87% (1.323GiB / 3.906GiB)`, and CPU `0.73%`. The SSH command is `ssh ip172-18-0-61-cgijg50sf2q000d373e0@direct.labs.play-`. Below the details are buttons for `DELETE` and `EDITOR`. A red arrow points to the `EDITOR` button. Below the buttons is a terminal window showing the following output:

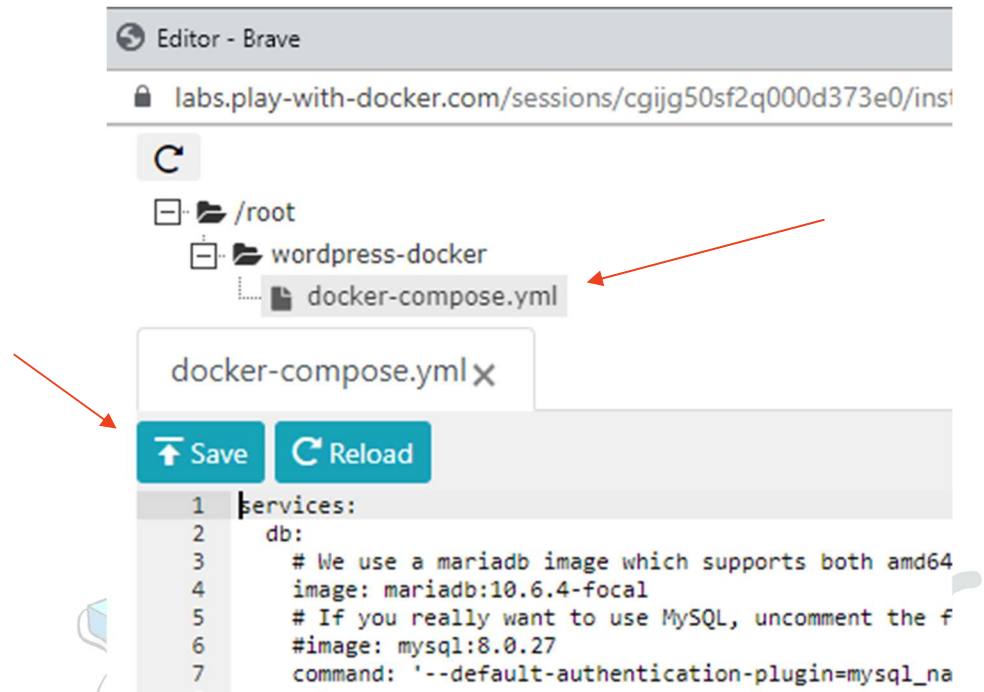
```
:: Network wordpress-docker_default Created 0.1s
:: Volume "wordpress-docker_db_data" Created 0.0s
:: Volume "wordpress-docker_wp_data" Created 0.0s
:: Container wordpress-docker-wordpress-1 Started 0.8s
:: Container wordpress-docker-db-1 Started 0.8s
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
```

9. A new pop up window will open.

The screenshot shows a browser window titled "Editor - Brave" with the URL `labs.play-with-docker.com/sessions/cgijg50sf2q000d373e0/instances/cgijg50s_...`. A notification bar at the top says "Create or upload files in the session terminal and then refresh". The file explorer on the left shows the following structure:

- /root
 - wordpress-docker
 - docker-compose.yml

10. Navigate to the file you made with the “.yaml” or “.yml” extension & then paste the code there then press save button. (Code is written in end of the report)



11. Your file is ready to be used, to run the file use “docker compose up -d”, shown below :

```

[node1] (local) root@192.168.0.28 ~/wordpress-docker
$ docker compose up -d

```

12. It will take some time and after that you will see the results like this as shown :

```

# 7b1a6ab2e44d Pull complete      11.9s      3.3s
# 034655750c88 Pull complete      3.4s
# f0b757a2a0f0 Pull complete      3.9s
# 5c37daf8b6b5 Pull complete      4.3s
# b4cd9409b0f6 Pull complete      4.4s
# dbcd906785eb Pull complete      5.0s
# a34cd90f184c Pull complete      5.2s
# fd6cef4ce489 Pull complete      5.3s
# 3cb89a1550ea Pull complete      11.5s
# df9f153bd930 Pull complete      11.7s
[+] Running 5/5
# Network wordpress-docker_default Created      0.1s
# Volume "wordpress-docker_db_data" Created      0.0s
# Volume "wordpress-docker_wp_data" Created      0.0s
# Container wordpress-docker-wordpress-1 Started      0.8s
# Container wordpress-docker-db-1 Started      0.8s
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$

```


13. Now to check whether the processes running or not, use “docker ps” to check.

```
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
f0564f90be4c   wordpress:latest "docker-entrypoint.s..." 9 minutes ago  Up 9 minutes  0.0.0.0:80->80/tcp       wordpress-docker-wordpress-1
d65ec15e8412   mariadb:10.6.4-focal "docker-entrypoint.s..." 9 minutes ago  Up 9 minutes  3306/tcp, 33060/tcp      wordpress-docker-db-1
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
```

14. Click on open port as shown in picture below to check the WordPress site.

cgijg50s_cgijoaee69v000cl4i8g

IP
192.168.0.28

Memory
20.66% (826.4MiB / 3.906GiB)

CPU
0.47%

SSH
ssh ip172-18-0-61-cgijg50sf2q000d373e0@direct.labs.play-

DELETE EDITOR

```
$
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
NAMES
f0564f90be4c   wordpress:latest "docker-entrypoint.s..." 9 minutes ago  Up 9 minutes  0.0.0.0:80->80/tcp
wordpress-docker-wordpress-1
d65ec15e8412   mariadb:10.6.4-focal "docker-entrypoint.s..." 9 minutes ago  Up 9 minutes  3306/tcp, 33060/tcp
wordpress-docker-db-1
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
```

15. A new small window will appear on top to put the port number, give 80 as input.

play-with-docker.com/p/cgijg50sf2q000d373e0#cgijg50s_cgijoaee69v000cl4i8g

cgijg50s_c

labs.play-with-docker.com says
What port would you like to open?

80

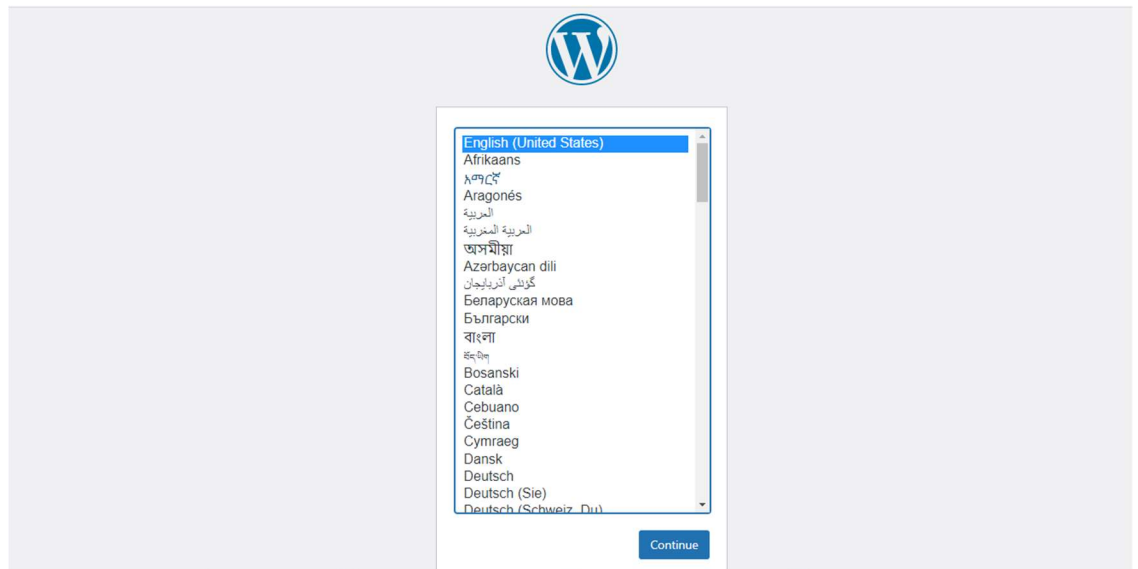
OK Cancel

SSH
ssh ip172-18-0-61-cgijg50sf2q000d373e0@direct.labs.play-

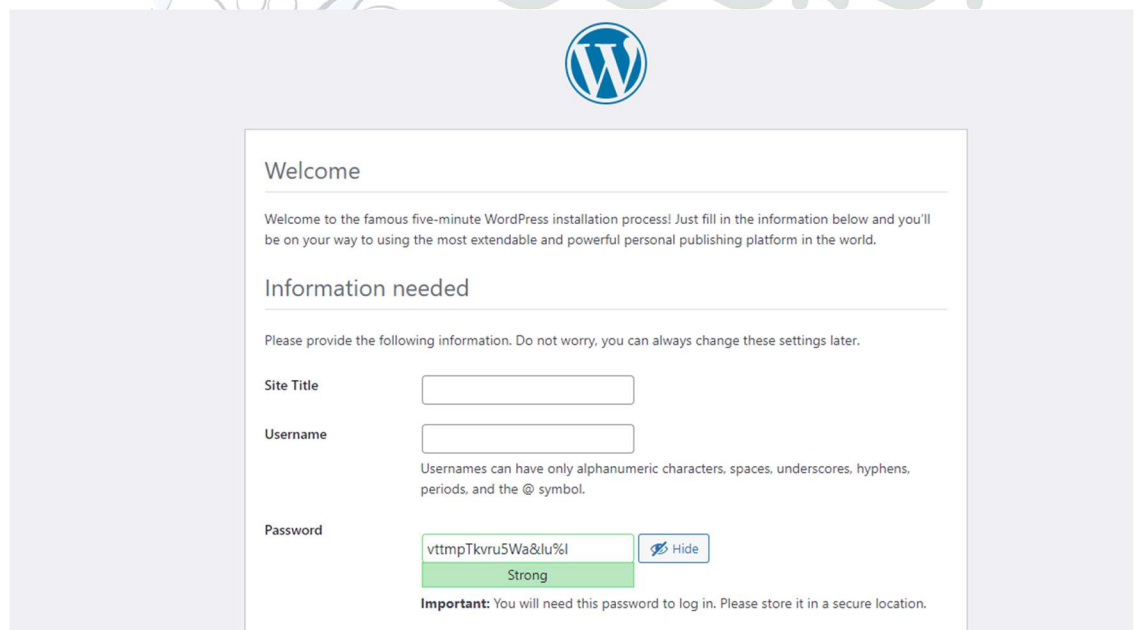
DELETE EDITOR

```
$
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
NAMES
f0564f90be4c   wordpress:latest "docker-entrypoint.s..." 9 minutes ago  Up 9 minutes  0.0.0.0:80->80/tcp
wordpress-docker-wordpress-1
```

16. You will be redirected to the new browser tab with will open the WordPress for you.



17. As you can see that Site running perfectly well.

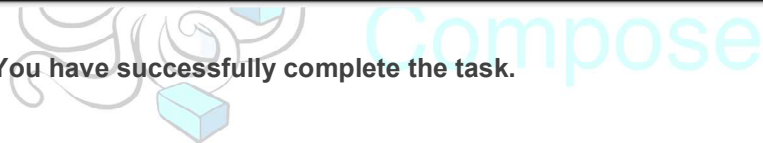


18. Now if you do not wish to continue & want to stop the process and shut down the database and docker compose use “docker compose down” which will stop the Docker compose and “docker compose down –volumes” command to stop & remove the databases attached with that Docker compose.

```
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$ docker compose down
[+] Running 3/3
  :: Container wordpress-docker-wordpress-1   Removed
  :: Container wordpress-docker-db-1          Removed
  :: Network wordpress-docker_default         Removed
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
```

```
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$ docker compose down --volumes
[+] Running 2/2
  :: Volume wordpress-docker_wp_data   Removed
  :: Volume wordpress-docker_db_data   Removed
[node1] (local) root@192.168.0.28 ~/wordpress-docker
$
```

19. You have successfully complete the task.



Yaml code & breakdown

```
1.1 services:
1.2   db:
1.3     # We use a mariadb image which supports both amd64 & arm64 architecture
1.4     image: mariadb:10.6.4-focal
1.5     # If you really want to use MySQL, uncomment the following line
1.6     #image: mysql:8.0.27
1.7     command: '--default-authentication-plugin=mysql_native_password'
1.8     volumes:
1.9       - db_data:/var/lib/mysql
1.10    restart: always
1.11    environment:
1.12      - MYSQL_ROOT_PASSWORD=somewordpress
1.13      - MYSQL_DATABASE=wordpress
1.14      - MYSQL_USER=wordpress
1.15      - MYSQL_PASSWORD=wordpress
1.16    expose:
1.17      - 3306
1.18      - 33060
1.19    wordpress:
1.20      image: wordpress:latest
1.21      volumes:
1.22        - wp_data:/var/www/html
1.23      ports:
1.24        - 80:80
1.25      restart: always
1.26      environment:
1.27        - WORDPRESS_DB_HOST=db
1.28        - WORDPRESS_DB_USER=wordpress
1.29        - WORDPRESS_DB_PASSWORD=wordpress
1.30        - WORDPRESS_DB_NAME=wordpress
1.31    volumes:
1.32      db_data:
1.33      wp_data:
```

