

Migrating From WordPress to Hugo

Adrian Kosmaczewski

2022-04-22

I've been migrating old blog posts (2004-2014) to this blog lately, and you can find them by clicking the “Next” button at the bottom of the index.

I found the corresponding old MySQL database backup at the very bottom of an old backup disk, and to export them into Markdown, I had to do a few things. Maybe these instructions will be useful for somebody else.

Requirements

These are the only technical requirements for this operation:

- Docker or Podman;
- A web browser.

1. Run MariaDB and WordPress

The first thing is to run a combo MariaDB and WordPress; let's use this simple `docker-compose.yml` file:

```
version: '3.3'

services:
  db:
    image: mariadb:latest
    container_name: mariadb
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: myrootpass
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    container_name: wordpress
```

```
depends_on:
  - db
image: wordpress:latest
ports:
  - "80:80"
restart: always
environment:
  WORDPRESS_DB_HOST: db:3306
  WORDPRESS_DB_USER: wordpress
  WORDPRESS_DB_PASSWORD: wordpress
  WORDPRESS_DB_NAME: wordpress
```

```
volumes:
  db_data:
```

And now:

```
$ docker-compose up
```

Boom, MariaDB and WordPress are running.

2. Import Dump

We can restore the database dump, but before we must do some cleanup:

- First unzip the backup file, which yields a backup file with plenty of SQL commands.
- Then remove those tables from the SQL backup file created by plugins and completely useless in this context, like `wp_twitter`, `wp_woocommerce_*`, `wp_yoast_seo_*`, and others.

But there are a few more issues:

- It turns out that I had exported the database with the wrong collation; hence I edited the backup SQL file to correct encoding errors for old articles, in particular accents in articles in Spanish and French. *Le sigh.*
- I did not remember the password used to access WordPress back then, so I simply used this tool to create a new hash for a new password of the `admin` user. Then I copied the new hash in the `wp_users` table at the bottom of the dump. Et voilà.
- Finally, to avoid WordPress needlessly redirecting URLs, in the `wp_options` table I changed the entries from the old URL to `http://localhost` (no `https!` Just plain `http...`) in the `siteurl` and `home` entries.

Before importing the dump, check that you have an empty database called `wordpress`:

```
$ docker exec -it mariadb mysql --user=root --password=myrootpass
Welcome to the MariaDB monitor.  Commands end with ; or \g.
```

```
Your MariaDB connection id is 3
Server version: 10.7.3-MariaDB-1:10.7.3+maria-focal mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]> show databases;
+-----+
| Database          |
+-----+
| information_schema|
| mysql             |
| performance_schema|
| sys               |
| wordpress         |
+-----+
5 rows in set (0.001 sec)
```

```
MariaDB [(none)]> use wordpress
Database changed
MariaDB [wordpress]> show tables;
Empty set (0.001 sec)
```

```
MariaDB [wordpress]>
```

And now import the data, just piping from stdin:

```
$ cat data | docker exec -i mariadb mysql --user=root \
    --password=myrootpass --database wordpress
```

This only concerns the text; since I did not find a backup of the `wp-content` folder, many images are missing at the moment. I'll add those as soon as I find them.

3. Access WordPress Console

Now, the moment of truth; open the `http://localhost/wp-admin/` URL, and login:

1. Upgrade the database as requested by the latest WordPress; in my case, incredibly enough, it worked without a glitch.
2. Log in as `admin` with the new password, the one hashed in a previous step.
3. Change the theme to “Twenty Twenty” (I personally had no idea what theme I used a decade ago...)

And lo and behold, I could browse my old writing from that point on.

4. Export to Jekyll / Hugo

Finally, I wanted to get all of this content out of the database, in individual files ready for Hugo:

1. I installed the WordPress to Jekyll Exporter plugin (which conveniently enough has absolutely no settings or options)
2. Selected the “Tools” > “Export to Jekyll” option. A zip file downloaded with all posts ready to be imported in Markdown format.

In your `~/Downloads` folder you’ll find a nice compressed archive with all of your stuff ready to add to Hugo.

5. Cleanup

Shut down the containers:

```
$ docker-compose down -v
```

The `-v` option removes all volumes attached to the containers.

I’m in the process of reviewing and editing the archives, but many are already in place, stretching back to 2004.