My Wordpress-Docker

Paul Huygen

7th June 2023

Abstract

This document generates a Docker image that contains te Wordpress website of CLTL.nl. It is derived from a back-up fom the actual website.

1 Introduction

This document constructs a restauration of a Wordpress website from a back-up. A Docker image with the restauration of the site is made. It serves the following purposes:

1. Provide proof that the back-up is complete. In other words, it is guaranteed that, if a disaster happens, the website can be restored.
2. Describe how it works. After a while, the knowledge on how to use the software instruments for the restauration becomes rusty. Hopefully this document provides clear instructions.
3. The software on the original site has not been updated for a long time. When a Docker with a duplicate exists, this can serve as a template to test upgrading.

1.1 Tasks to be performed

First We have to do the following:

1. Construct a Docker image with the correct version of the operating system on it. The original server uses a very old version of Ubuntu linux: `14.04.5 LTS, Trusty Tahr`.
2. Restore the accounts of the users that where known in the original server.
3. Install the database and restore the WordPress part on it.
4. Install the Apache web-server.
5. Restore the WordPress site. This is of a very old version too.
6. Test whether it works.

When this works, we will bother about upgrading the operating system and the Wordpress version to the latest.

2 Construct the docker image

The following rudimentary Dockerfile generates an image for an Ubuntu 14.04 server. After you run the image, you can contact it via the terminal. When you stop it, all modifications are lost.

"Dockerfile" 1≡

```
FROM ubuntu:14.04
EXPOSE 80
⟨ copy stuff to the image 2d ⟩
⟨ "run" commands in Dockerfile 2e ⟩
CMD ["/bin/bash"]
◇
```

"doit" 2a≡
```
#!/bin/bash
# doit -- generate the image
⟨ init doit 3a ⟩
make sources
⟨ build the Docker image 2b ⟩
# ⟨ finit doit ? ⟩
◇
```

⟨ build the Docker image 2b ⟩ ≡
```
docker build -t ubuntu-docker .
◇
```
Fragment referenced in 2a.

To restore the Wordpress-site on the image, `RUN` a script with instructions.

"restore" 2c≡
```
#!/bin/bash
⟨ restore instructions 2f ⟩
◇
```

⟨ copy stuff to the image 2d ⟩ ≡
```
COPY --chmod=755 restore /root/restore
◇
```
Fragment referenced in 1.

⟨ "run" commands in Dockerfile 2e ⟩ ≡
```
RUN /root/restore
◇
```
Fragment referenced in 1.

3 Connect to the back-up of the original source

We used backup2l to back-up the server, so we need this program in our image te restore things:

⟨ restore instructions 2f ⟩ ≡
```
apt-get update
apt-get install backup2l
◇
```
Fragment referenced in 2c.
Defines: `backup2l` Never used.

Mount the directory of the back-up on directory `/backup`. We assume that the back-up files that backup2l has made are available on directory `/home/paul/mnt/b2l` on the Docker host. The `docker run` instruction contains a mount option that connects this directory to the local `/backup` directory.

⟨ *init doit* 3a ⟩ ≡
```
source  $HOME/bin/local_wpbak_setup
```
◇

Fragment referenced in 2a.

4 Run the Docker image

"run_the_image" 3b≡
```
#!/bin/bash
# run_the_image -- start a container with the cltl image
docker run   -it --mount type=bind,src=/home/paul/mnt/b2l,target=/backup  ubuntu-docker
```
◇

5 Indexes

5.1 Filenames

5.2 Macro's

5.3 Variables