

Narration&Reference Easy realization of Pandoc's operating environment with Docker

Base File Name: NarrationReference_Docker_Pandoc_ja_07_en

2018.11.23

Easy realization of Pandoc's operating environment with Docker

by *Shuichi Ohtsu*

Operating environment

This time, I will introduce an example of easily creating Pandoc's environment by using Docker on WSL.

I mentioned how to install Pandoc on Ubuntu on WSL and add LaTeX and cross reference tool *Pandoc-Crossref* etc, but this installation takes a lot of time.

Also, due to dependency, we also needed to install Haskell and Cabal.

With Docker you can save time and effort of these installations at once.

This time, I will show you how to achieve the same function by using the Docker container.

As a working environment, Docker installed on Ubuntu on WSL is required.

If you have not installed it please refer to other videos and install Docker in advance.

Download of Docker image (for Pandoc)

First, open Ubuntu on the WSL.

Then create a specific directory for running Pandoc.

First, enter `cd /mnt/c` and move to Windows C drive.

Then create a specific directory.

In this example, I created a directory called `v1.0` in `__myprg/Docker/Ubuntu/Pandoc` and created a sample Markdown file there.

Next, in this directory, enter `code .` to start the Visual Studio Code.

First, download the image file of Docker.

Open the Docker Hub in your browser.

That URL is `https://hub.docker.com/u/ohtsu/`.

When you open this page, an image called `pandoc_full` is registered, so download this image with the pull command.

Return to the VS Code and enter `docker pull ohtsu/pandoc_full:1.0` in the terminal window.

The last number is the version number.

Next, click the *Docker icon* on the left side to see the pull image.

Sure, it is being downloaded.

Starting the Docker image

Right-click on this image, open the pop-up menu and select *Run Interactive*.

A new container is created and executed.

Then the terminal window opens and the bash prompt opens.

The default directory is `/home`.

Now, type `ls` and try to display the file.

Then you will see the style file for the bibliography list, the literature file for BibTeX, the JavaScript file for Node.js for creating PDF from Markdown and the sample Markdown file.

Type `exit` here and exit this container.

Background execution of Docker container

Next, start this container in the background and accept commands all the time.

Also, give a name to the container so that you can call it by its name.

```
docker container run --name "test01" -itd ohtsu/pandoc_full:1.0.
```

test01 is the container's distinguished name.

-itd means to output results to the console and run in the background.

Click the Docker icon on the left side and try opening *Containers*.

Then you can see that *test01* container is indeed working.

The green icon indicates that it is in operation.

Checking the Markdown file

Next, display the Markdown file in the current directory.

This file is not just a Markdown file, it is a file premised on Pandoc's extension function.

In other words, LaTeX command is inserted everywhere, and header, footer, table of contents, bibliography list are automatically inserted.

LaTeX is particularly good at displaying mathematical expressions, and it seems to be a useful tool for scientific and technical papers.

Copy Markdown file to container

Next we need to copy this Markdown file to the *test 01* container.

The convenience at this time is the *docker container cp* command.

Type `docker container cp ./docker_pandoc_sample01.md test01:/home/sample01.md`

The meaning is to copy the `docker_pandoc_sample01.md` file in the current directory as a file name `sample01.md` in the `/home` directory of the container with identifier *test01*.

Next, check whether it was able to actually copy.

From the *Containers* list on the left, select *test01*, right click to display the pop-up menu and select *Attach Shell*.

When a new terminal window opens, type `ls` to display the file.

The extension of `sample01` was insufficient. I will fix this.

It is OK.

Close the terminal while keeping the container running

Close this terminal window by clicking the trash can icon on the upper right.

Conversion from Markdown to PDF

Since we could copy the Markdown file to the container, give the command from outside the container and generate the PDF file from the Markdown file.

Type `docker container exec -it test01 node makepdf01 sample01`.

This means that you must start the *test01* container, use Node.js, start `makepdf01.js`, and convert the `sample01.md` file into a `sample01.pdf` file.

Next, check whether the `sample01.pdf` file is actually generated inside the container.

From the *Containers* list on the left, select *test01*, right click to display the pop-up menu and select *Attach Shell*.

If you type `ls` and display the file, you can confirm that the `sample01.pdf` file is actually generated.

Close this terminal window by clicking the trash can icon on the upper right.

Copy the PDF file inside the container to the host side

Next, copy the PDF file generated inside the container to the current directory.

Again, we are going to use the *docker container cp* command.

Note, however, that the order of the arguments is reversed.

That is, `docker container cp copy source container identification name:inside container path name copy destination path name`.

Type `docker container cp test01:/home/sample01.pdf ./sample01.pdf`.

Open the current directory.

Certainly `sample01.pdf` has been copied.

Display of generated PDF file

Here, right click on `sample01.pdf` on VS Code, and select *Display by explorer* from the pop-up menu.

Then double-click `sample01.pdf`.

The PDF file will be displayed.

First, the table of contents is displayed.

A table is also displayed.

Equations including root symbols etc. are also displayed properly.

Footnotes are OK too.

A bibliography list is also automatically generated.

For the users of LaTeX, these are obvious, but by using Pandoc on Docker this time, we have broken the technical barriers of the preliminary environment setting.

For science and technology experts, it seems to be very useful.

Thank you for your watching.

Reference

Docker

- "Docker Community Edition for Windows",
<https://store.docker.com/editions/community/docker-ce-desktop-windows>
- "Docker/Kubernetes 実践コンテナ開発入門",
<http://amazon.co.jp/o/ASIN/4297100339/>
- "プログラマのためのDocker教科書 第2版 インフラの基礎知識&コードによる環境構築の自動化",
<http://amazon.co.jp/o/ASIN/4798153222/>

Pandoc

- "Sample data and program(Ohtsu/pandoc-o2-makepdf)",
<https://github.com/Ohtsu/pandoc-o2-makepdf>
- "citation-style-language/styles",
<https://github.com/citation-style-language/styles/>
- "Pandoc ユーザーズガイド 日本語版",
<http://sky-y.github.io/site-pandoc-jp/users-guide/>
- "Pandoc User's Guide",
<https://pandoc.org/MANUAL.html>
- "WSL (Windows Subsystem on Linux) で pandoc メモ",

<https://qiita.com/miyamiya/items/4d2e93ad7895e302c27e>

- "プログラミングPandoc",
https://www.amazon.co.jp/dp/4274067815/ref=sxbs_sxwds-stvp_1?pf_rd_m=AN1VRQENFRJN5&pf_rd_p=14895845-6b63-47e2-b967-96bf0ca66fcb&pd_rd_wg=ZJDGq&pf_rd_r=0649V2CWECEG2NZ0KXHKB&pf_rd_s=desktop-sx-bottom-slot&pf_rd_t=301&pd_rd_i=4274067815&pd_rd_w=zZKAt&pf_rd_i=Pandoc&pd_rd_r=7e5f7e6c7-4c7f-9e7e-a4ef2c2531f2&ie=UTF8&qid=1541321932&sr=1
- "化学系だけど Markdown でレポートを書いて Pandoc を使った",
<http://pinkmagenta.hatenablog.jp/entry/2017/12/20/124911>
- "Pandocで相互参照",
<http://pinkmagenta.hatenablog.jp/entry/2017/12/20/124911>
- "MarkdownとPandocを使って論文っぽい文章を書く",
<https://inody1991.tumblr.com/post/134742076815/markdown%E3%81%A8pandoc%E3%82%9C>

Others

- "Containerizing Angular with Docker - Dan Wahlin",
<https://www.youtube.com/watch?v=cLT7eUWKZpg&t=1140s>
- "Deploy Angular 5 app in Docker Container in under 10 mins - For local development",
<https://www.youtube.com/watch?v=L2UkQ2CND68&t=178s>
- "Angular5, Angular6, Angular7 Custom Library: Step-by-step guide",
<https://www.udemy.com/angular5-custom-library-the-definitive-step-by-step-guide/>
- "Angular5, Angular6, Angular7用 カスタムライブラリの作成: 完全ステップ・バイ・ステップ・ガイド",
<https://www.udemy.com/angular5-1/>