

Narration&Reference DockerでPandocの稼働環境を簡単実現

Base File Name: NarrationReference_Docker_Pandoc_ja

2018.11.22

DockerでPandocの稼働環境を簡単実現

by *Shuichi Ohtsu*

動作環境

今回は、WSL上のDockerを利用することにより、Pandocの環境を簡単に作成する例をご紹介します。

以前、WSL上のUbuntuにPandocをインストールし、さらにLaTeXや相互参照ツール *Pandoc-Crossref* などを追加する方法をご紹介しましたが、このインストールには大変な時間を要します。

また、依存性の関係で、HaskellやCabalをインストールする必要もありました。

Dockerを利用するとこれらのインストールの手間を一気に省くことができます。

今回は、Dockerのコンテナを利用することにより、同様の機能を実現する方法をご紹介します。

なお、稼働環境として、WSL上のUbuntuにインストールされたDockerが必要となります。

まだインストールされていない方は、他のビデオを参照して、Dockerをインストールしておいてください。

Dockerイメージ(Pandoc用)のダウンロード

まず、WSL上のUbuntuを開きます。

そして、Pandoc稼働用に特定ディレクトリを作成します。

まず、`cd /mnt/c`と入力して、WindowsのCドライブに移ります。

そして特定のディレクトリを作成します。

ここでは、`__myprg/Docker/Ubuntu/Pandoc`内に`v1.0`というディレクトリを作成し、そこにサンプル用のMarkdownファイルを作成しました。

次に、このディレクトリ内で、`code .`と入力して、Visual Studio Codeを起動します。

まず、Dockerのイメージファイルをダウンロードします。

ブラウザで、Docker Hubを開きます。

そのURLは、<https://hub.docker.com/u/ohtsu/>です。

このページを開くと、pandoc_fullというイメージが登録されていますので、このイメージをpullコマンドでダウンロードします。

VS Codeに戻り、ターミナル・ウィンドウで`docker pull ohtsu/pandoc_full:1.0`と入力します。

最後の数字は、バージョン番号です。

次に左側のDockerアイコンをクリックして、pullしたイメージを確認します。

確かに、ダウンロードされています。

Dockerイメージの起動

このイメージを右クリックして、ポップアップメニューを開き、*Run Interactive*を選択します。

すると新たにコンテナが生成され、それが実行されます。

そしてターミナル・ウィンドウが開き、bashのプロンプトが開きます。

デフォルトのディレクトリは、`/home`となっています。

ここで、`ls`と入力して、ファイルを表示してみます。

すると、文献リスト用のスタイルファイル、BibTeX用の文献ファイル、MarkdownからPDFを作成するためのNode.js用のJavaScriptファイルやサンプルのMarkdownファイルが表示されます。

ここで`exit`と入力して、このコンテナを一旦終了します。

Dockerコンテナのバックグラウンド実行

次に、このコンテナをバックグラウンドで起動し、常時コマンドを受け付けるようにします。

また、コンテナに名称をつけ、その名称で呼び出すことができるようにします。

`docker container run --name "test01" -itd ohtsu/pandoc_full:1.0`と入力します。

`test01` はコンテナの識別名です。

`-itd` はコンソールに結果を出すとともに、バックグラウンドで実行することを意味します。

左側のDockerアイコンをクリックし、*Containers*を開いてみます。

すると*test01*コンテナが確かに稼働しているのがわかります。

緑色のアイコンは、稼働中であることを示しています。

Markdownファイルの確認

次にカレントディレクトリにあるMarkdownファイルを表示してみます。

このファイルは、単なるMarkdownファイルではなく、Pandocの拡張機能を前提としているファイルとなっています。

すなわち、随所にLaTeXのコマンドを挿入し、ヘッダ、フッタ、目次、さらに参考文献リストを自動挿入するようになっています。

LaTeXは、特に数式の表示に長けており、科学技術系の論文を作成する方にとっては、便利なツールになると思われます。

Markdownファイルのコンテナへのコピー

次にこのMarkdownファイルを*test01*コンテナにコピーする必要があります。

この時に便利なのが、*docker container cp*コマンドです。

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

と入力します。

その意味は、カレントディレクトリにある、*docker_pandoc_sample01.md*ファイルを、識別子*test01*のコンテナの/*home*ディレクトリに*sample01.md*というファイル名としてコピーするということになります。

次に、実際にコピーできたかどうかを確認します。

左側の*Containers*リストから、*test01*を選択し、右クリックしてポップアップメニューを表示し、*Attach Shell*を選択します。

新たなターミナル・ウィンドウが開きましたら、*ls*と入力し、ファイルを表示します。

*sample01*の拡張子が不足していました。これを修正します。

OKです。

コンテナを起動したまま、ターミナルを閉じる

このターミナル・ウィンドウを、右上のごみ箱アイコンをクリックして閉じます。

MarkdownからPDFへの変換

PDFに変換するMarkdownファイルをコンテナにコピーできましたので、次にコンテナ外からコマンドを与えて、MarkdownファイルからPDFファイルを生成します。

`docker container exec -it test01 node makepdf01 sample01`と入力します。

この意味は、*test01*コンテナを起動し、Node.jsを利用して、`makepdf01.js`を起動し、`sample01.md`ファイルを`sample01.pdf`ファイルに変換せよ、ということになります。

次に実際にコンテナ内で、`sample01.pdf`ファイルが生成されているかを確認します。

左側のContainersリストから、*test01*を選択し、右クリックしてポップアップメニューを表示し、*Attach Shell*を選択します。

`ls`と入力してファイルを表示すると、実際に`sample01.pdf`ファイルが生成されていることを確認することができます。

このターミナル・ウィンドウを、右上のごみ箱アイコンをクリックして閉じます。

コンテナ内のPDFファイルをホスト側へコピー

次に、コンテナ内で生成されたPDFファイルをカレント・ディレクトリにコピーします。

この場合も、`docker container cp`コマンドを利用します。

但し、引数の順序が逆になることに注意してください。

すなわち、`docker container cp コピー元 コンテナ識別名:コンテナ内パス名 コピー先パス名`となります。

`docker container cp test01:/home/sample01.pdf ./sample01.pdf`と入力します。

カレント・ディレクトリを開いてみます。

確かに、`sample01.pdf`がコピーされています。

生成されたPDFファイルの表示

ここで、VS Code上で、`sample01.pdf`を右クリックし、さらにポップアップメニューから、*エクスプローラーで表示*を選択します。

そして、`sample01.pdf`をダブルクリックします。

するとPDFファイルが表示されます。

まず、目次が表示されます。

表も表示されます。

さらにルート記号などを含めた数式もきちんと表示されています。

脚注もOKです。

参考文献リストも自動的に生成されています。

LaTeXの利用者にとっては、これらは当然のことですが、今回のDocker上のPandocを利用することによって、前提となる環境設定の技術的障壁を崩したことになります。

科学技術系の専門家の方には、ぜひ利用していただきたいと思います。

ご清聴ありがとうございました。

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=zZKA&pf_rd_i=Pandoc&pd_rd_r=7e5fe6c7-4c7f-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/>