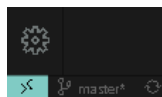


VS Code で Docker にリモート接続する (続き)

今回は, VS Code が提供している `Miniconda` 用の `Docker image` を用いて環境構築を行う.

まず, 2021 年 1 月 24 日の操作と同様に, 以下の手順を踏む.

1. 図の `><` ボタンを押す.



2. 1.表示された選択肢から, `Remote-Containers: Reopen in Container` を選択する.

```
Remote-Containers: Reopen in Container
Remote-Containers: Open Folder in Container...
Remote-Containers: Open Workspace in Container...
Remote-Containers: Clone Repository in Container Volume...
Remote-Containers: Attach to Running Container...
Remote-Containers: Add Development Container Configuration Files...
Remote-Containers: Try a Sample...
Remote-Containers: Getting Started
```

3. そこで表示された選択肢から `Miniconda` を選択する.

```
GitHub Codespaces - Stretch (Legacy)
Legacy Debian 9 / Stretch based Codespaces universal default image.
Anaconda (Python 3)
Develop Anaconda applications in Python3.
Miniconda (Python 3)
Develop Miniconda applications in Python 3.
Alpine
Simple Alpine container with Git installed.
Bazel (Community)
```

これにより、ワーキングディレクトリ上に以下のファイル構造を持つディレクトリが生成される。

```
.
|-- ./devcontainer
|   |-- library-scripts
|   |   |-- common-debian.sh
|   |   |-- node-debian.sh
|   |   |-- python-debian.sh
|   |-- devcontainer.json
|   |-- docker-compose.yml
|   |-- Dockerfile
|   |-- noop.txt
```

そのとき生成された `Dockerfile` を以下に示す。

```
# See here for image contents: https://github.com/microsoft/vscode-dev-containers/tree/v0.155.1/containers/python-3-miniconda/.devcontainer/base.Dockerfile

FROM mcr.microsoft.com/vscode/devcontainers/miniconda:0-3

# [Option] Install Node.js
ARG INSTALL_NODE="true"
ARG NODE_VERSION="lts/*"
RUN if [ "${INSTALL_NODE}" = "true" ]; then su vscode -c "umask 0002 && . /usr/local/share/nvm/nvm.sh && nvm install ${NODE_VERSION} 2>&1"; fi

# Copy environment.yml (if found) to a temp location so we update the environment. Also
# copy "noop.txt" so the COPY instruction does not fail if no environment.yml exists.
COPY .devcontainer/environment.yml .devcontainer/noop.txt /tmp/conda-tmp/
RUN if [ -f "/tmp/conda-tmp/environment.yml" ]; then /opt/conda/bin/conda env update -n base -f /tmp/conda-tmp/environment.yml; fi \
    && rm -rf /tmp/conda-tmp

# [Optional] Uncomment to install a different version of Python than the default
# RUN conda install -y python=3.6 \
#     && pip install --no-cache-dir pipx \
#     && pipx reinstall-all

# [Optional] Uncomment this section to install additional OS packages.
# RUN apt-get update && export DEBIAN_FRONTEND=noninteractive \
#     && apt-get -y install --no-install-recommends <your-package-list-here>
```

この `Dockerfile` では、`.yaml` ファイルを使用することで `conda` 環境を更新する機能がデフォルトで搭載されている。また、`.yaml` ファイルが存在しない場合にもエラーが発生しないよう、`noop.txt` ファイルが置かれている。

2. yaml ファイルを使った環境のアップデート

.yaml ファイルを使用して, conda 環境をアップデートする. そのために, まず Dockerfile を以下のように変更する.

```
# See here for image contents: https://github.com/microsoft/vscode-dev-containers/tree/v0.155.1/containers/python-3-miniconda/.devcontainer/base.Dockerfile
FROM mcr.microsoft.com/vscode/devcontainers/miniconda:0-3

# [Option] Install Node.js
ARG INSTALL_NODE="true"
ARG NODE_VERSION="lts/*"
RUN if [ "${INSTALL_NODE}" = "true" ]; then su vscode -c "umask 0002 && . /usr/local/share/nvm/nvm.sh && nvm install ${NODE_VERSION} 2>&1"; fi

# Copy environment.yml (if found) to a temp location so we update the environment. Also
# copy "noop.txt" so the COPY instruction does not fail if no environment.yml exists.
COPY .devcontainer/environment.yml .devcontainer/noop.txt /tmp/conda-tmp/
RUN if [ -f "/tmp/conda-tmp/environment.yml" ]; then /opt/conda/bin/conda env update -n base -f /tmp/conda-tmp/environment.yml; fi \
    && rm -rf /tmp/conda-tmp
```

さらに、ワーキングディレクトリ上に `environment.yml` を配置する.

```
.
|-- ./devcontainer
    |-- library-scripts
        |-- common-debian.sh
        |-- node-debian.sh
        |-- python-debian.sh
    |-- devcontainer.json
    |-- docker-compose.yml
    |-- Dockerfile
    |-- environment.yml
    |-- noop.txt
```

また, `environment.yml` には以下のように `conda` 環境に追加インストールするパッケージが記述されている.

```
# Docker 内の cnonda 上に構築する開発環境の設定

name: myenv
channels:
  - conda-forge
  - defaults
dependencies:
  - anaconda-client=1.7.2=py38_0
  - black=19.10b0=py_0
  - easydict=1.9=py_0
  - matplotlib=3.3.2=h06a4308_0
  - mypy=0.800=pyhd3eb1b0_0
  - numpy=1.19.2=py38h54aff64_0
  - pandas=1.1.5=py38ha9443f7_0
  - pillow=8.0.1=py38he98fc37_0
#   - pysimplegui=4.32.1=pyhd3deb0d_0=conda-forge
  - python=3.8.5=h7579374_1
  - pytorch=1.4.0=cuda101py38h02f0884_0
```

さらに、 `devcontainer.json` ファイルに仮想環境上の VS Code にインストールする追加パッケージを記述しておく。

```
// For format details, see https://aka.ms/devcontainer.json. For config options, see the README at:
// https://github.com/microsoft/vscode-dev-containers/tree/v0.155.1/containers/python-3-miniconda
{
  "name": "Miniconda (Python 3)",
  "build": {
    "context": "..",
    "dockerfile": "Dockerfile",
    "args": {
      "INSTALL_NODE": "true",
      "NODE_VERSION": "lts/*"
    }
  },
  // Set *default* container specific settings.json values on container create.
  "settings": {
    "terminal.integrated.shell.linux": "/bin/bash",
    "python.pythonPath": "/opt/conda/bin/python",
    "python.linting.enabled": true,
    "python.linting.pylintEnabled": true,
    "python.formatting.autopep8Path": "/usr/local/py-utils/bin/autopep8",
    "python.formatting.blackPath": "/usr/local/py-utils/bin/black",
    "python.formatting.yapfPath": "/usr/local/py-utils/bin/yapf",
    "python.linting.banditPath": "/usr/local/py-utils/bin/bandit",
    "python.linting.flake8Path": "/usr/local/py-utils/bin/flake8",
    "python.linting.mypyPath": "/usr/local/py-utils/bin/mypy",
    "python.linting.pycodestylePath": "/usr/local/py-utils/bin/pycodestyle",
    "python.linting.pydocstylePath": "/usr/local/py-utils/bin/pydocstyle",
    "python.linting.pylintPath": "/usr/local/py-utils/bin/pylint"
  },
  // Add the IDs of extensions you want installed when the container is created.
  "extensions": [
    "coenraads.bracket-pair-colorizer-2",
    "oderwat.indent-rainbow",
    "ms-ceintl.vscode-language-pack-ja",
    "ms-python.python"
  ],
  // Use 'forwardPorts' to make a list of ports inside the container available locally.
  // "forwardPorts": [],
  // Use 'postCreateCommand' to run commands after the container is created.
  // "postCreateCommand": "python --version",
  // Comment out connect as root instead. More info: https://aka.ms/vscode-remote/containers/non-root.
  "remoteUser": "vscode"
}
```

ここまでで仮想環境立ち上げの準備が整ったので、実際に仮想環境を立ち上げる。
立ち上げは、以下の手順で行う。

1. 図の >< ボタンを押す。



2. 1.表示された選択肢から, Remote-Containers: Reopne in Container を選択する。

```
Remote-Containers: Reopen in Container
Remote-Containers: Open Folder in Container...
Remote-Containers: Open Workspace in Container...
Remote-Containers: Clone Repository in Container Volume...
Remote-Containers: Attach to Running Container...
Remote-Containers: Add Development Container Configuration Files...
Remote-Containers: Try a Sample...
Remote-Containers: Getting Started
```

これで仮想環境が構築される。