To Clone the NEOH-Docker from GitHub

Navigate to the following URL https://github.com/UAH-NEOH/neoh-docker
Or
Enter the following command in your terminal

```
git clone https://github.com/UAH-NEOH/neoh-docker
```

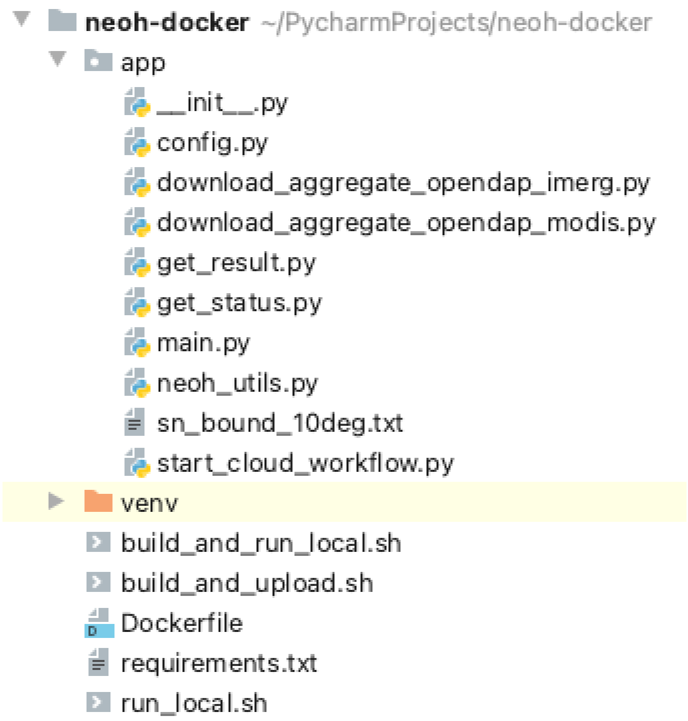
Pre-requisite software

1. Docker (https://www.docker.com/)
2. Postman (https://www.postman.com/) (Optional)
3. Python 3.9

System requirements

Minimum resources allocated to Docker Desktop
8 GB RAM
100 GB HDD (Volume-storage)

The directory structure would look similar to this



```
neoh-docker ~/PycharmProjects/neoh-docker
  app
    __init__.py
    config.py
    download_aggregate_opendap_imerg.py
    download_aggregate_opendap_modis.py
    get_result.py
    get_status.py
    main.py
    neoh_utils.py
    sn_bound_10deg.txt
    start_cloud_workflow.py
  venv
  build_and_run_local.sh
  build_and_upload.sh
  Dockerfile
  requirements.txt
  run_local.sh
```

The `app` directory contains the NEOH source code. The files can be modified to user/developer needs.

The `config.py` file contains the credentials information required for this app to work. To request data from NASA servers. It first needs to be authenticated by the Earthdata URS login (https://urs.earthdata.nasa.gov/) If you already have an account enter the username and password in the config.py file. Otherwise, Click Register on the Earthdata URS login page to get started.

Main.py Python file contains the Fast API declaration. The user can find all the end-point declarations. Based on the API endpoint, handlers are called to process the user payload. The process will be passed down to `start_cloud_workflow.py`

Start_cloud_workflow.py is where the user payload is gathered and processed (boundary files). The request-id is been generated and passed down to the download/aggregate step. The Request id is sent as a response to the user.

Download_aggregate_opendap_imerg.py handles the IMERG data download and aggregate process for Precipitation data.

Download_aggregate_opendap_modis.py handles the MODIS data download and aggregate process for Temperature and Vegetation data.

Get_result.py and **Get_status.py** is used to read the result and status of the requests inside the Docker

Sn_bound_10deg.txt boundary reference files used in the MODIS process

Neoh_utils.py The python helper library for creating directories and writing status information inside Docker.

Requirements.txt

The file contains the python dependency libraries for the NEOH process. This will be installed inside the Docker.

The Dockerfile has the step by step instructions for the installation and setup of the python tools in a virtual machine inside the Docker

The Bash scripts:

Build_and_upload.sh

Builds the Docker image and uploads it to the Docker Desktop app

Also, stops and removes the previous setup with the same name (if any). This step is required for continuous development and deployment.

Runs the Docker in the specified port. The default is 80:80.

Build_and_run_local.sh

Builds the Docker image and runs the service locally

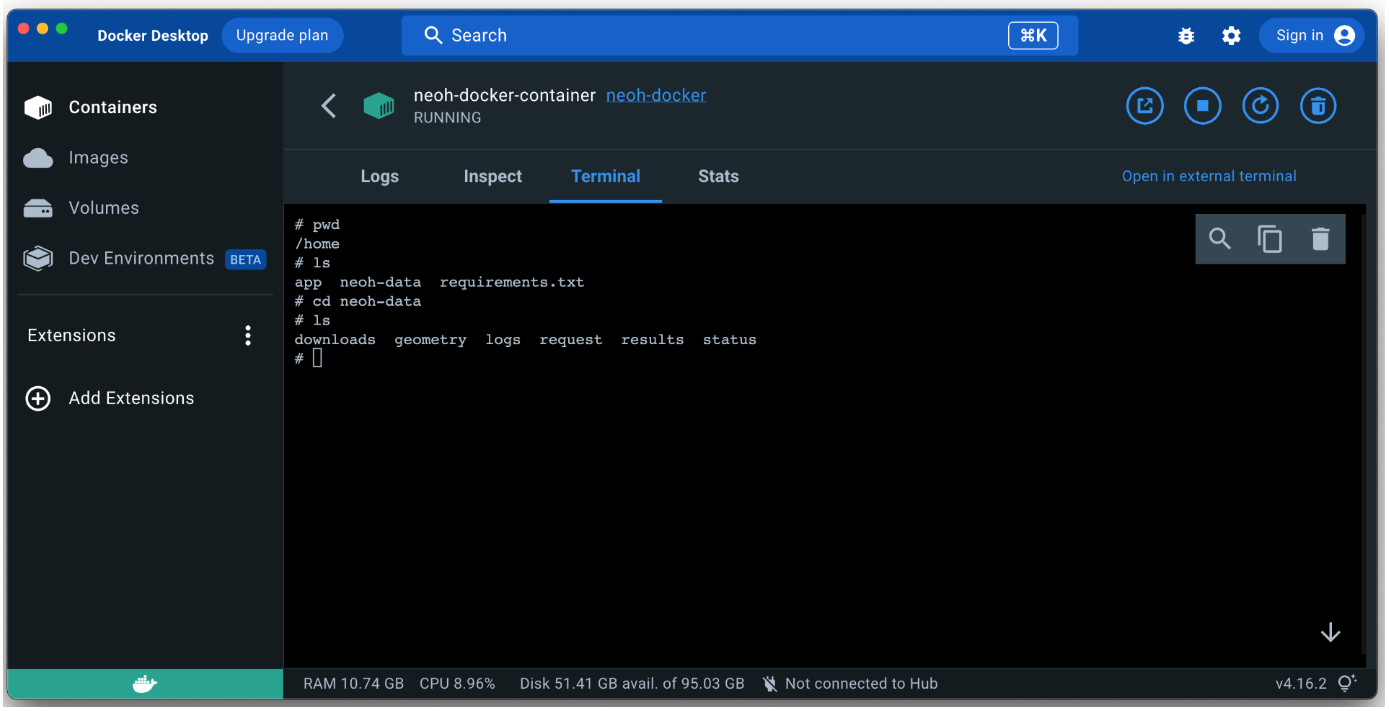
Run_local.sh

Run the Python server without installing the dependencies. The dependencies should be already installed

Inside the Docker:

The working/current directory inside the Docker is `/home`

Once the NEOH app gets initialized, a directory called `neoh-data` is created.



```
# pwd
/home
# ls
app  neoh-data  requirements.txt
# cd neoh-data
# ls
downloads  geometry  logs  request  results  status
# 
```

The `neoh-data` directory consists of 6 subdirectories.

Downloads

Contains downloaded data from NASA servers (if any). If a process uses OPENDAP, it directly reads the data rather than downloading it

Geometry

Contains the GeoJson files from the user input. The boundary files are stored in JSON format and used by the IMERG and MODIS handlers.

Logs

A python logger file is created. Neoh-data.log. It contains all the processing info happening within the NEOH processing code. This file may help to identify the issue/errors (if any) during the run-time.

Request

Consists of the JSON payload created by the user request. The file contains the dataset name and filters sent through the request

Results

The output generated from the NEOH output is stored in this directory. The output is in the format of DHIS2 Data Element entry.

Status

The NEOH get_status API uses the contents from this folder. The status of each request is stored under its request-id_status.json file.

Steps to use this application

1. Start the Docker Desktop Application
2. Open the terminal, navigate to the neoh-docker directory, execute the following command in the root directory

```
bash build_and_upload.sh
```

Note: Open the `Dockerfile`, and adjust the host and port values accordingly. Other services may use the same host and port values. It may create a conflict.

By default, it is using Localhost and port 80 in Docker.

```
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "80"]
```

Change the following line in Docker file to some different number, for eg:

```
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "90"]
```

Note: The Download won't initiate unless the Earthdata login is created and used in the following file.
app/config.py

Enter the following values

```
  Earthdata_username = 'XXXXXXX'
  Earthdata_password = 'XXXXXXX'
```

Note: Make sure you remember which host and port the Docker server is running. If you plan to use the NEOH-webapp, you need this information. The values need to be in **neoh-webapp/src/config.js**

Since the default host and port are localhost and port 80. `http://localhost:80` is used in the webapp

NEOH Webapp

To clone the NEOH Webapp

Navigate to the following URL https://github.com/UAH-NEOH/neoh-webapp
Or
Enter the following command in your terminal

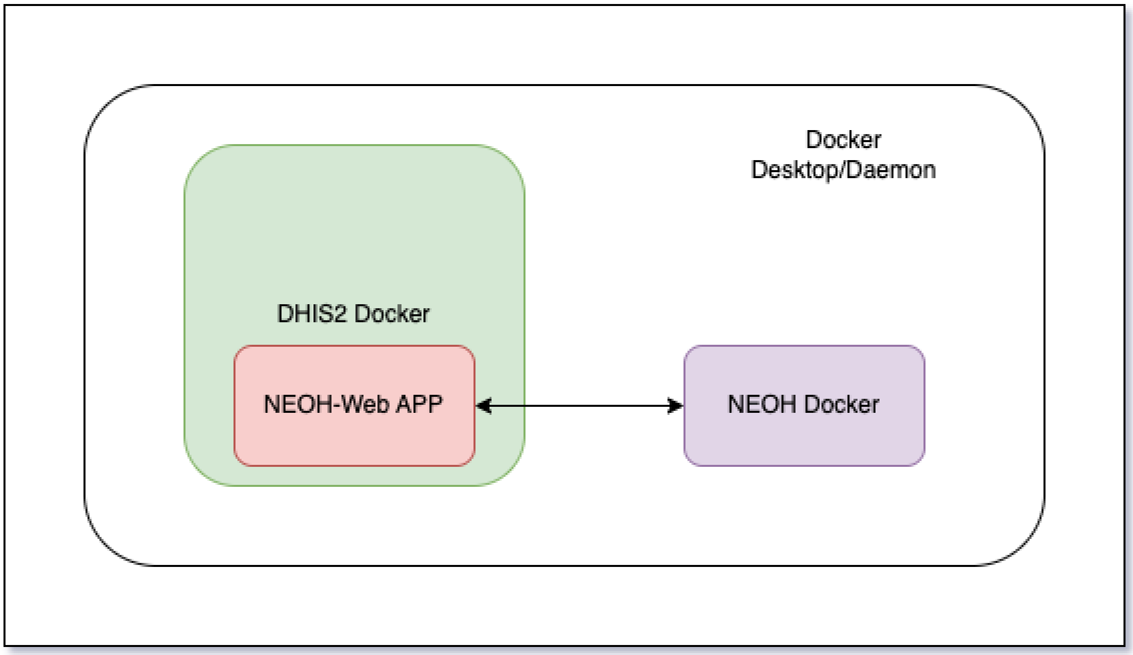
```
git clone https://github.com/UAH-NEOH/neoh-webapp
```

You need to make changes to the following files (if needed) to make sure the services are connected.

.env file in the root directory
Config.js file in **neoh-webapp/src/config.js**

If you plan to run the DHIS2 Server and NEOH webapp locally,

Follow the link to setup DHIS2 locally,
https://developers.dhis2.org/docs/guides/spin-up-local-instance/



The DHIS2 will run on the Docker if you decide to go with local deployment.

On the successful setup, you will be given which port the DHIS2 is running. Or You can specify which port the DHIS2 server needs to run.

You can find that out using the D2 CLI, by following the command

```
d2 cluster list  or
d2 cluster -help
```

You need to extract the port number, which is needed for compiling the NEOH- web app
The NEOH web app needs to communicate with DHIS2 instance to read/store data.

For eg: The DHIS2 is running in localhost port 8080, `http://localhost:8080`

Open the .env file from neoh-webapp root directory and enter the address as the DHIS2_BASE_URL value.

```
DHIS2_BASE_URL=http://localhost:8080
```

If you host the DHIS2 server in any server, enter the URL address of the server in the .env file

For eg:

```
DHIS2_BASE_URL=https://neoh-dhis2.itsc.uah.edu
```

(The URL without the trailing slash)

—

Next step would be to find out which host and port the NEOH Docker is running. By default, the NEOH Docker runs on localhost port 80 `http://localhost:80`

By default, the value `http://localhost:80` is present in the **neoh-webapp/src/config.js** file. If you plan to use different values (Running a server in different port), you need to update the values in **neoh-webapp/src/config.js**

Once you complete these steps, you are now ready to install, run, and build (package) the web app.

Pre-requisite softwares

1. Node JS (Version 14) https://nodejs.org/en/
2. Yarn https://yarnpkg.com/
3.

On the root directory of neoh-webapp, run the following command to install the dependency packages.

```
yarn install
```

Once the packages got installed, run the following command to start the application locally.

```
yarn start
```

The application will be served on `http://localhost:3000` by default.

Open the browser and load the `http://localhost:3000` URL to visit the NEOH-web app

To build the application, execute the following command on the root directory

```
yarn build
```

Enter the details on the prompt. On successful execution, a directory called dist is created.

The zip file will consist of the NEOH-webapp compiled source code. You can upload the zip file to your DHIS2 instance directly.