Disclaimer: This is a pre-release version of the AIDAmri pipeline. Bugs may appear and some functions as well as the GUI are not accessible yet. Please handle if with care and double-check resulting data.

Prerequisites

First install Docker by clicking **here** to get your Docker distribution. We recommend to follow the **getting started tutorial** to get comfortable with the basic Docker usage and terms.
Clone the AIDAmri git repository to get all files needed to initiate the AIDAmri pipeline:

```
git clone -b dockerdev \
https://github.com/aswendtlab/AIDAmri.git
```

This automatically moves your repository to the `dockerdev` branch. Please note that the other branches do not contain any files to set-up the Docker.

Quick start and usage

If you wish to quickly build and start the pipeline, a bash runfile (`docker_runfile.sh`) is provided. If you use a Windows system, you may need to install Bash or use PowerShell (click here for instructions to use Bash on Windows). Mac and Linux users should be able to execute the bash file via a terminal without further installations.
Open a terminal and switch to the AIDAmri directory:

```
cd PATH/TO/AIDAmri
```

There, execute the runfile by typing in the following command:

```
bash docker_runfile.sh -b -a PATH/TO/DATAFOLDER
```

or

```
bash docker_runfile.sh -ba PATH/TO/DATAFOLDER
```

The `-b` flag initializes the building process for the Docker image called "aidamri:dev" (see below for further details). The `-a` flag attaches your shell to the container called "aidamri", so you will be able to use the command line interface (CLI) of the AIDAmri pipeline. If you only like to run the container without attaching to it, do not enter the flags. To attach to the running container, (i.e. the active pipeline) afterwards, use the command:

```
docker attach aidamri
```

To enter the running container, use the following command:

```
docker attach aidamri
```

You can leave a running container without stopping by pressing `CTRL+P` and `CTRL+Q` consecutively. Alternatively, you can type in `exit`, however this will stop and delete the container, thus you need to re-run the container after stopping it. Keep in mind, that you need to re-run the container after reboot. Use

```
bash docker_runfile.sh PATH/TO/DATAFOLDER
```

to re-run the container (building the image is not necessary if no updates were made within the dockerdev git branch). Use the `-a` flag as mentioned above to directly attach to your pipeline.

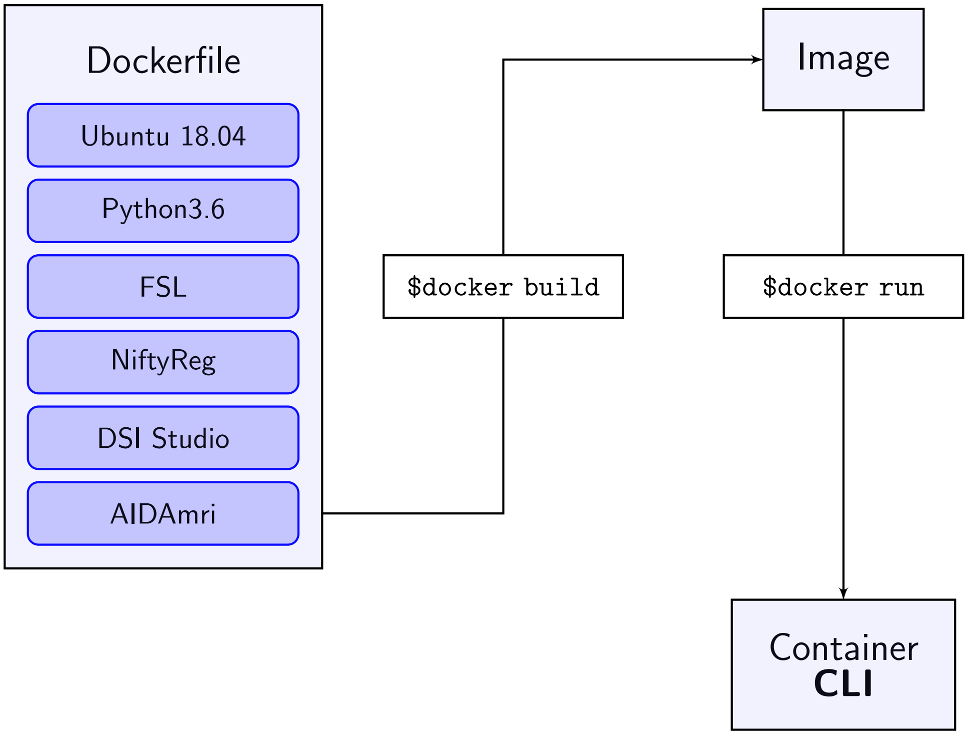

Figure 1: Docker architecture draft. The blue boxes within the Dockerfile box depict the main content layers. The boxes preceded with an $ are command line codes. Click on texts within the boxes for further information.

Creating image

To initiate the image building process, open your shell to grant yourself access to the command line interface (CLI). Change your directory to the cloned GitHub repository.

```
cd PATH/TO/AIDAmri
```

You can check the folder contents by using the `ls` command. A file named Dockerfile, as well as the `fslinstaller_mod.py`, a `bin/` and a `lib/` folder should be located in this directory. Then, launch the docker daemon to build the image:

```
docker build -t aidamri:dev -f Dockerfile .
```

The created image is a template for running so-called containers (see below section) and instantiate the pipeline. Be aware that the period at the end is part of the command and refers to the corresponding directory. The `-t` flag calls for the name and the tag that you wish to give your image, in this case it is called 'aidamri' and tagged 'latest'. You may change name and tag to your liking, but keep in mind to change them accordingly in later steps that invoke the image. The `-f` tag calls for the Dockerfile in the current directory. Let the process run. Docker desktop should show you the image within the image tap once it is built.
You only need to run the building process once initially or after updating the github after new changes were made. If an update occured, the building process will only update those layers that were changed, so the process will not take as long as the initial built.

Running container and mount data

Before running a container, make sure you know where exactly your data you wish to process is located on your host system. The container will be an instance written from the built image and serves as an environment for you to use the AIDAmri pipeline. To run the container, enter the following command in your command line:

```
docker run -dit \
--rm --name aidamri \
--mount type=bind,source=PATH/TO/DATA,target=/aida/DATA \
aidamri:dev
```

The first flag, -dit starts the container as a detached mode, hence the d, as well as an interactive mode, that allows you to actively use this environment, hence the it. Alternatively you can directly enter the container environment by only using the -it flag. The --rm flag will delete the container after stopping it. This flag may be optional but it is recommended to use this flag to avoid container housekeeping after you complete the task. With the --name tag you give your container a name, in this case it is called "aidamri". Note, that the image name and the container are named individually. It is recommended to give your container a different name from the image to avoid confusion. Note further that if you wish to use more than one instance of running containers (e.g. to process multiple datasets simultaneously) you need to name each container differently.

Binding mounts create a reference to a given directory, allowing the container to access and transmuting data on the host system. In this case it should allow the the pipeline to process your MRI data without copying or re-allocating your data. The --mount flag with the type=bind argument allows you to grant your container the access to your target directory. Type in the absolute path at the source placeholder. Do not use relative paths. Within the container environment, a working directory called aida/ was created. It is recommended to allocate your reference path within this directory by typing in the name of your data folder according to the target placeholder, i.e. /aida/DATA. You can give the target folder another name than the source directory as it still will refer to the source but it is advised to keep the name to avoid confusion. Be aware not to place any spaces between the commas.

Pass the name and tag of your image, i.e. here aidamri:dev at the end of the command line. After initializing a ID will appear and the Docker Desktop app should show the running container. You can also check whether the container is running by typing docker container ls.

To enter the running container, use the following command:

```
docker attach aidamri
```

You can leave a running container without stopping by pressing CTRL+P and CTRL+Q consecutively. Alternatively, you can type in exit, however this will stop and delete the container if the --rm flag was used in the running command, thus you need to re-run the container after stopping it.