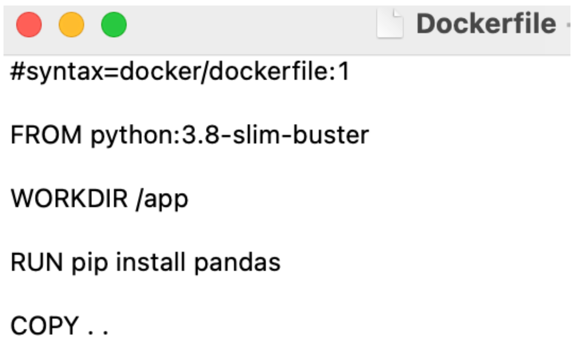
Run without sync to the local file system

If we want to provide someone with a container with a set of necessary files (so that they can run a complete process without configuring both the environment and having the necessary files), we can copy necessary files in the *Dockerfile* to the appropriate directory in the image with command `COPY . .`:



```
#syntax=docker/dockerfile:1

FROM python:3.8-slim-buster

WORKDIR /app

RUN pip install pandas

COPY . .
```

To create an image:

1. go to the folder where you have a set of necessary data
2. run a terminal in it, and in it run the command: `docker build -t bind_app1_app2:v2 .`

You can run the image created in this way in any directory - you do not need to have access to the necessary files in it:

`docker run -ti bind_app1_app2:v2 python3 run.py`.

Note that:

1. at the end of the command we added `python3 run.py`. This is because, according to the Dockerfile, our image does not run the `run.py` script.
2. running in a folder that does not have the necessary files does not generate errors
3. subsequent runs do not add any files to the local system.

Use cases

In particular, you can use the above mechanism in the following situations:

Sharing

I want to share an application that uses a database with others.

My application uses a database (such as Redis, PostgreSQL or MySQL). I want to share it with people who don't have it installed, nor can they configure or manage it.

I am using docker compose to create a multi-container application, containing both my application and the corresponding database.

Training.

I'm a trainer, and want to show someone a solution in jupyter notebook, using various libraries (such as PyCaret). The users do not have any environments installed.

I ask them to install Docker and then run my container.

Archiving the project

I want to ensure the reproducibility of my solution in the future.

I document and publish the docker image of my solution in the image repository accordingly.

Deployment

I want to simulate a production environment.

Our image analysis system is running on an NVidia Jetson Nano endpoint device. On my workstation, I want to simulate how the latest version of my model will behave on this device.

To do this, I am running a Docker container with an image of the end device.

I want to run multiple services or a sequence of services on the end device.

To do this, I use docker compose to create the launch of multiple services: in parallel or in the appropriate sequence.

I want to run my ML project in a scalable environment, such as Kubernetes, or at a service provider (GCP, MS Azure or Amazon EC2).

I create a separate image for each component of my machine learning process. I use the Kubeflow system to manage the full process in Kubernetes.

Useful sources

A very good presentation of the *bind mount* and *volume mount* methods can be found in the official docker documentation available here.