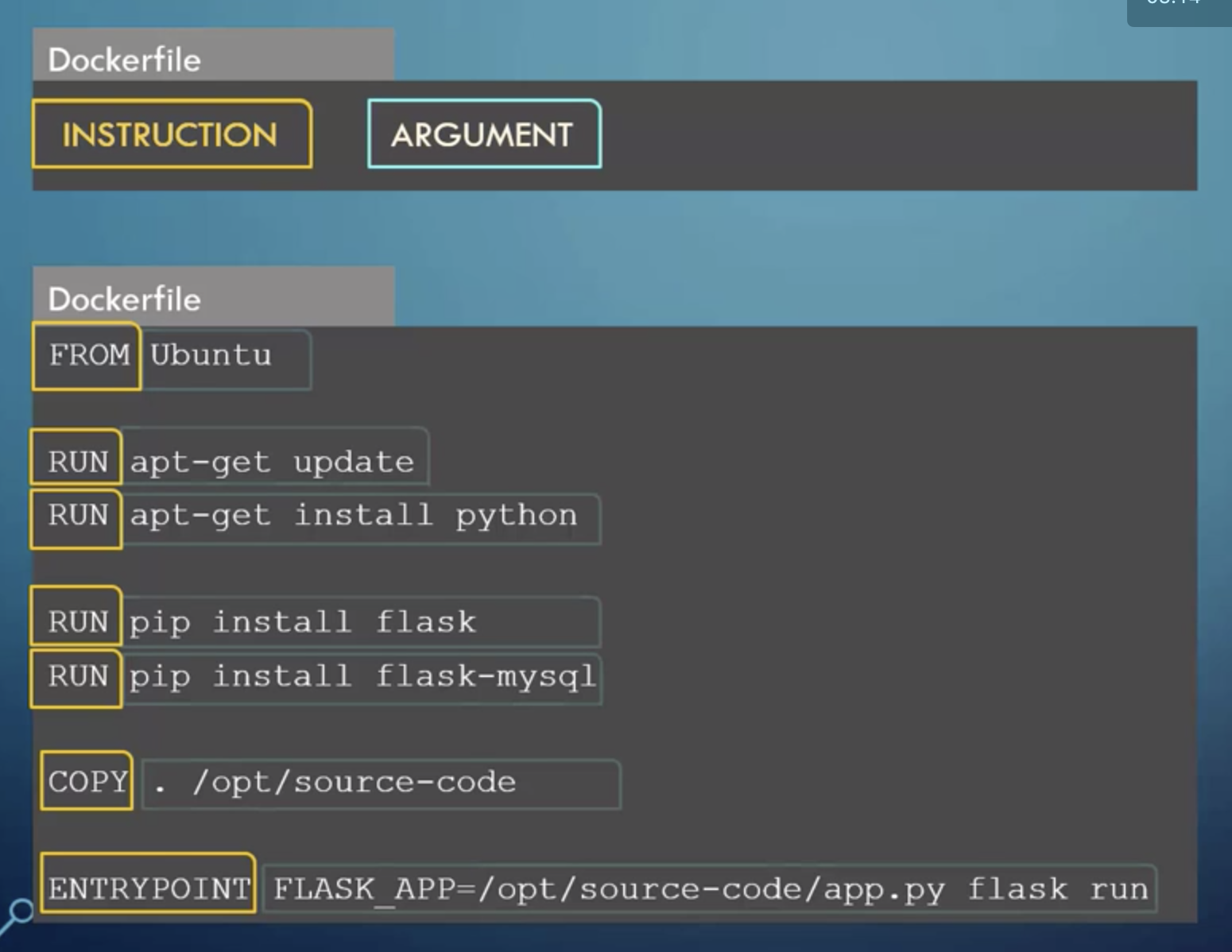
**Docker images**

* How to create your own image?
* Manual steps needed:

A picture containing screenshot

Description automatically generated

Docker File



$ docker build -t my-custom-app .

* First line: **FROM** ubuntu
  + Base OS for this container: Every docker image must be based-off of another image.
* RUN: Install all dependencies.
* COPY: Copy files from the local system to the docker image. In this case it will copy the source code from the local folder to /opt/source-code.

**Creating new Docker image**

* Source code of the web-app we want to containerize:

<https://github.com/mmumshad/simple-webapp-flask>

* **Step 1**: Create an ubuntu docker container and Install all dependencies.

$ docker run -it ubuntu bash

# apt-get update

# apt-get install -y python

# apt-get install -y python-pip

# pip install flask

* **Step 2: copy source code locally**

A screenshot of a cell phone

Description automatically generated

# cat > /opt/app.py

* **Step 3: Start webserver**

# FLASK\_APP=app.py flask run --host=0.0.0.0

\* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)

* + Remark: To access the web-app from outside the container, you need to map the port.
* **Step 4: Dockerize the application**
  + Create a Dockerfile inside folder ***my-simple-webapp***
  + Make sure source code app.py is inside ***my-simple-webapp***

A screenshot of a cell phone

Description automatically generated

* Build the image

$ cd my-simple-webapp

$ docker build . -t my-simple-webapp

* Port mapping so we can run it in our local host browser

$ docker run -p 5000:5000 my-simple-webapp

-Run in Google Chrome:

<http://localhost:5000/>

Welcome!

<http://localhost:5000/how%20are%20you>

I am good, how about you?

**Step 5: Push to docker hub**

* You need to build it with a **tag**

$ docker build . -t mehdiah/my-simple-webapp

* Enter login and password

$ docker login

* Push it to docker Hub

$ docker mehdiah/my-simple-webapp

**Command vs Entrypoint**

**A screenshot of a cell phone

Description automatically generated**