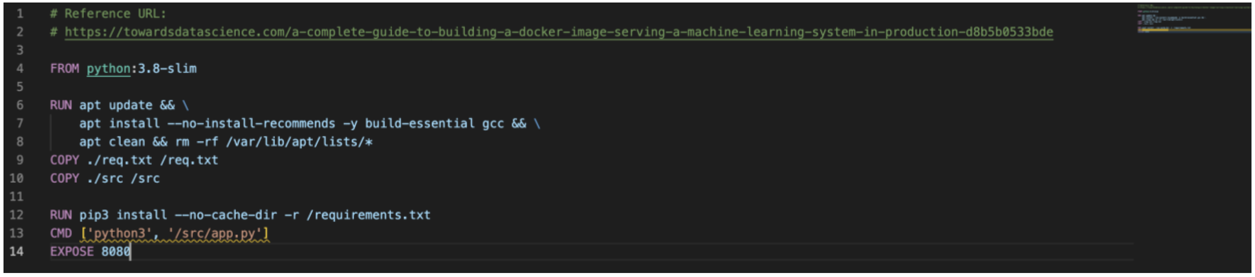
This’s my file I uploaded originally:

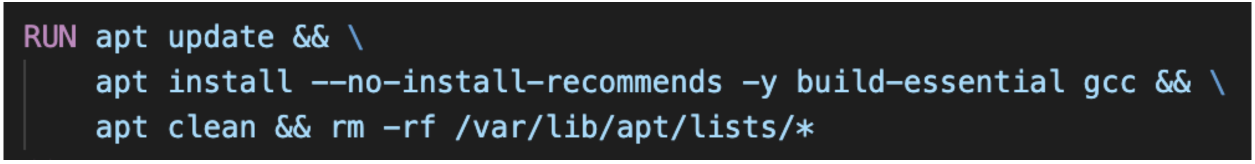


```
# Reference URL:
# https://towardsdatascience.com/a-complete-guide-to-building-a-docker-image-serving-a-machine-learning-system-in-production-d8b5b0533bde

FROM python:3.8-slim

RUN apt update && \
    apt install --no-install-recommends -y build-essential gcc && \
    apt clean && rm -rf /var/lib/apt/lists/*
COPY ./req.txt /req.txt
COPY ./src /src

RUN pip3 install --no-cache-dir -r /requirements.txt
CMD ['python3', '/src/app.py']
EXPOSE 8080
```

What I did is I followed the steps in the URL and tried to built a docker image to serve a machine learning system.

Major takeaways:

1. requirements.txt must always contains a python package version, don’t write a package name, as it will always install the latest package and during the process defeats the purpose of using docker.
2. Always group similar RUN command together and result in a single Docker layer:



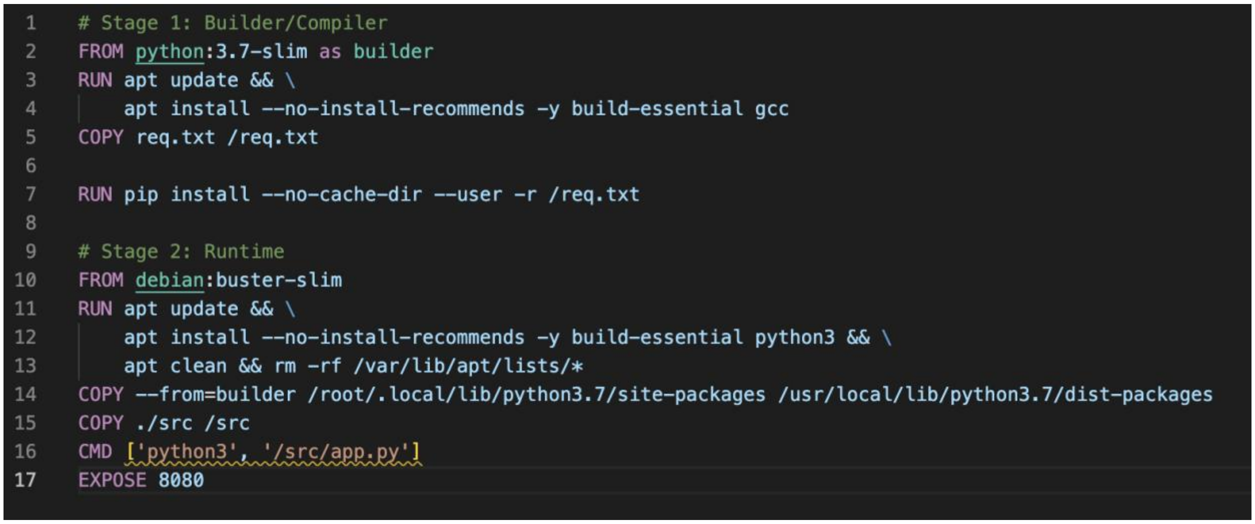
```
RUN apt update && \
    apt install --no-install-recommends -y build-essential gcc && \
    apt clean && rm -rf /var/lib/apt/lists/*
```

3. Use .dockerignore to avoid unnecessary build context.

Building a Docker image for any Python Project:

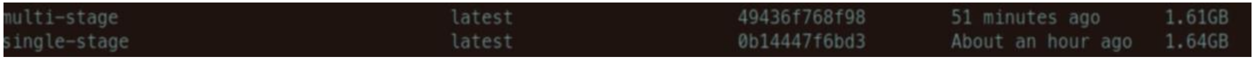
#Most of the time a ML system will be based on Python.

1. Single stage: the single-stage will perform all the task in the same/single docker build time.
2. Select a base image, install os packages, copy source, install packages, set entry point/other commands.
3. Multi stage: to optimize the dockerfile, use FROM statements and each FROM can use a different base, and each of them begins a new stage of build.



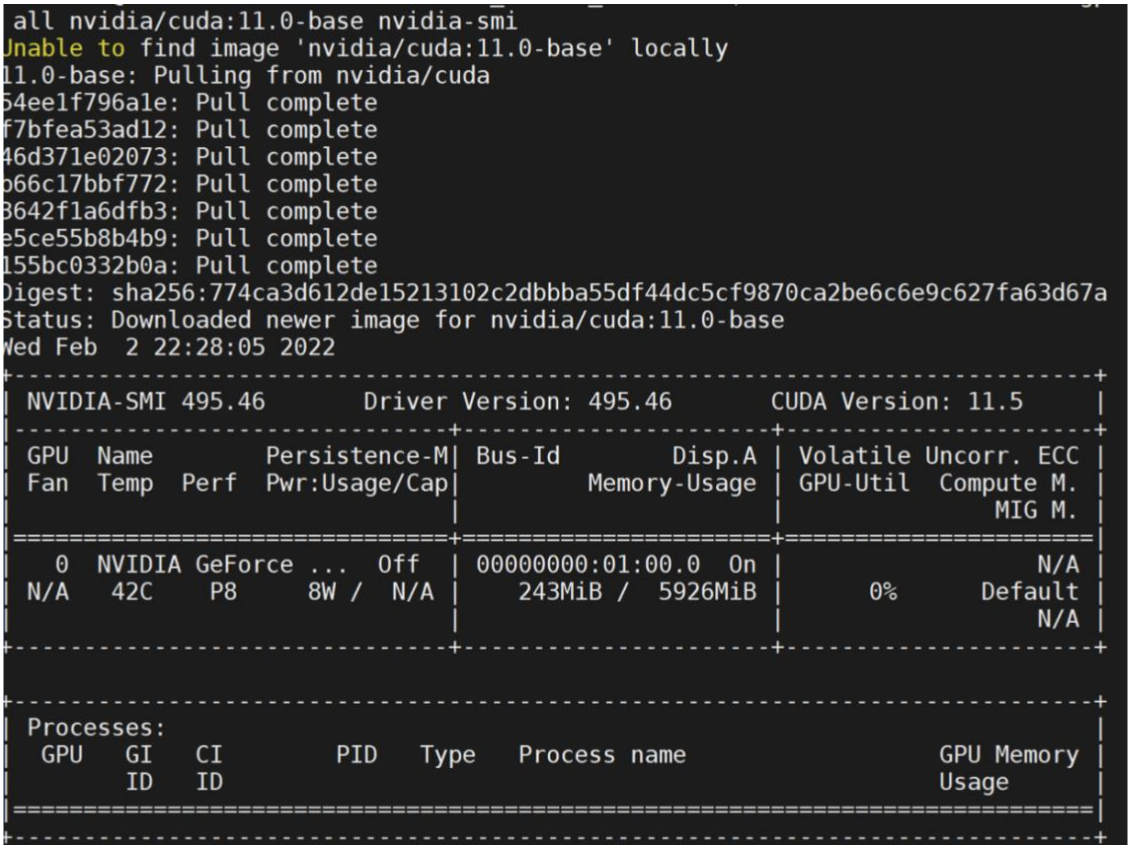
```
# Stage 1: Builder/Compiler
FROM python:3.7-slim as builder
RUN apt update && \
    apt install --no-install-recommends -y build-essential gcc
COPY req.txt /req.txt

RUN pip install --no-cache-dir --user -r /req.txt

# Stage 2: Runtime
FROM debian:buster-slim
RUN apt update && \
    apt install --no-install-recommends -y build-essential python3 && \
    apt clean && rm -rf /var/lib/apt/lists/*
COPY --from=builder /root/.local/lib/python3.7/site-packages /usr/local/lib/python3.7/dist-packages
COPY ./src /src
CMD ['python3', '/src/app.py']
EXPOSE 8080
```

Which reduced the docker image size:



```
multi-stage                          latest              49436f768f98        51 minutes ago       1.61GB
single-stage                         latest              0b14447f6bd3        About an hour ago    1.64GB
```

4. Then I started to experience nvidia-docker2 installation error
   Reference URL:
   https://docs.nvidia.com/ai-enterprise/deployment-guide/dg-docker.html
   and fixed it.



```
 all nvidia/cuda:11.0-base nvidia-smi
Unable to find image 'nvidia/cuda:11.0-base' locally
11.0-base: Pulling from nvidia/cuda
54ee1f796a1e: Pull complete
f7bfea53ad12: Pull complete
46d371e02073: Pull complete
b66c17bbf772: Pull complete
3642f1a6dfb3: Pull complete
e5ce55b8b4b9: Pull complete
155bc0332b0a: Pull complete
Digest: sha256:774ca3d612de15213102c2dbbba55df44dc5cf9870ca2be6c6e9c627fa63d67a
Status: Downloaded newer image for nvidia/cuda:11.0-base
Wed Feb  2 22:28:05 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 495.46       Driver Version: 495.46       CUDA Version: 11.5     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA GeForce ...  Off  | 00000000:01:00.0  On |                  N/A |
| N/A   42C    P8     8W /  N/A |    243MiB /  5926MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
+-----------------------------------------------------------------------------+
```