Docker Day 1 Assignment

Mohamed Emary

February 21, 2025

0.1 Question 1

> What is the difference between:
> - `CMD` & `ENTRYPOINT`

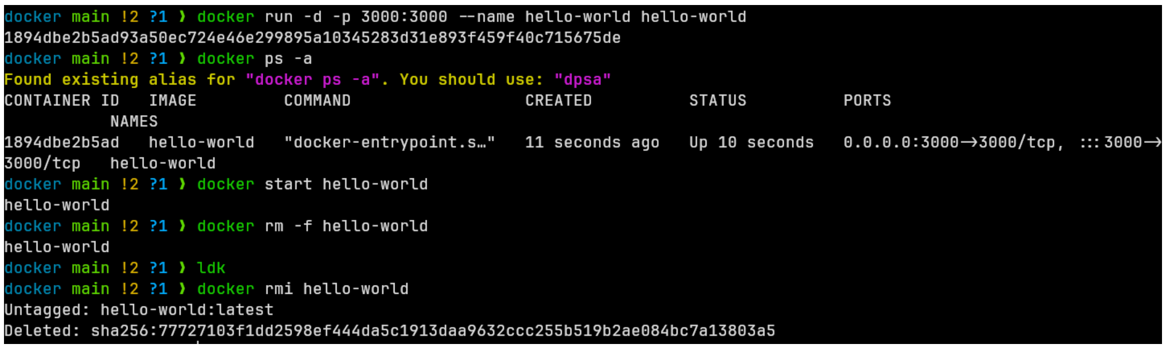
Feature	CMD	ENTRYPOINT
Primary Purpose	Provides default arguments	Configures container as executable
Override Behavior	Easily overridden by CLI args	Requires `--entrypoint` flag
Multiple Instructions	Only last CMD takes effect	Can combine with CMD args
Use Case	Default container behavior	When container acts as binary

> - `COPY` & `ADD`

Feature	COPY	ADD
Basic Function	Simple file copying	Advanced file operations
Local Files	Yes	Yes
Remote URLs	No	Yes
TAR Extraction	No	Yes (automatic)
Transparency	More transparent	More complex
Best Practice	Preferred for most cases	Use only when special features needed

0.2 Question 2

> - Run the container `hello-world`
> - Check the container status
> - Start the stopped container
> - Remove the container
> - Remove the image

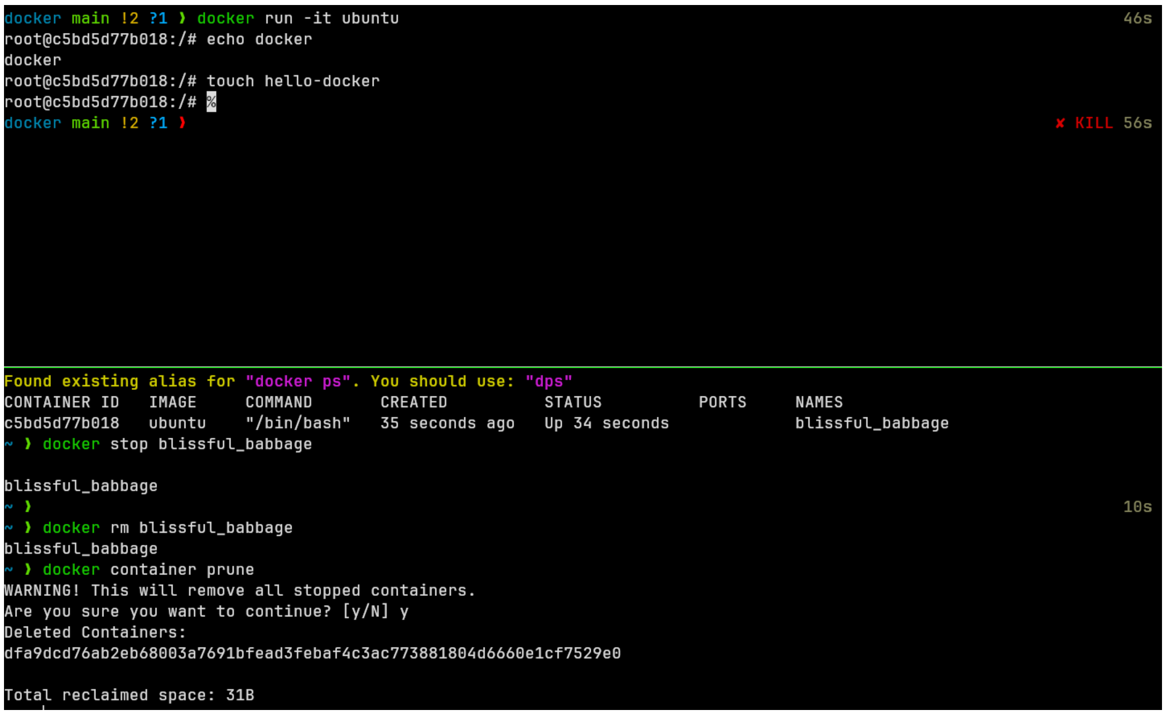


```
docker main !2 ?1 ❯ docker run -d -p 3000:3000 --name hello-world hello-world
1894dbe2b5ad93a50ec724e46e299895a10345283d31e893f459f40c715675de
docker main !2 ?1 ❯ docker ps -a
Found existing alias for "docker ps -a". You should use: "dpsa"
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS          PORTS
           NAMES
1894dbe2b5ad   hello-world   "docker-entrypoint.s…"   11 seconds ago   Up 10 seconds   0.0.0.0:3000→3000/tcp, :::3000→
3000/tcp   hello-world
docker main !2 ?1 ❯ docker start hello-world
hello-world
docker main !2 ?1 ❯ docker rm -f hello-world
hello-world
docker main !2 ?1 ❯ ldk
docker main !2 ?1 ❯ docker rmi hello-world
Untagged: hello-world:latest
Deleted: sha256:77727103f1dd2598ef444da5c1913daa9632ccc255b519b2ae084bc7a13803a5
```

Figure 1: Answer

0.3 Question 3

- Run container centos or ubuntu in an interactive mode
- Run the following command in the container `echo docker`
- Open a bash shell in the container and touch a file named `hello-docker`
- Stop the container and remove it. Write your comment about the file `hello-docker`
- Remove all stopped containers



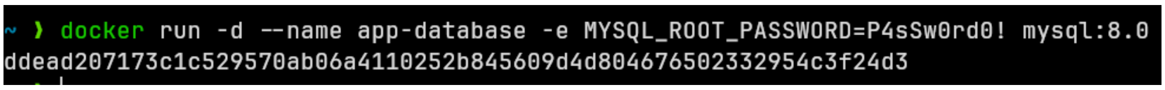
```
docker main !2 ?1 ❯ docker run -it ubuntu                                                                          46s
root@c5bd5d77b018:/# echo docker
docker
root@c5bd5d77b018:/# touch hello-docker
root@c5bd5d77b018:/# %
docker main !2 ?1 ❯                                                                                        ✘ KILL 56s
```

```
Found existing alias for "docker ps". You should use: "dps"
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS          PORTS     NAMES
c5bd5d77b018   ubuntu    "/bin/bash"   35 seconds ago   Up 34 seconds             blissful_babbage
~ ❯ docker stop blissful_babbage

blissful_babbage
~ ❯                                                                                                                10s
~ ❯ docker rm blissful_babbage
blissful_babbage
~ ❯ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
dfa9dcd76ab2eb68003a7691bfead3febaf4c3ac773881804d6660e1cf7529e0

Total reclaimed space: 31B
```

Figure 2: Answer

0.4 Question 4

- Deploy a MySQL database called `app-database`. Use the mysql latest image, and use the `-e` flag to set `MYSQL_ROOT_PASSWORD` to `P4sSw0rd0!`. The container should run in the background. I had `v8.0` of mysql installed, so I used it instead of `latest`.

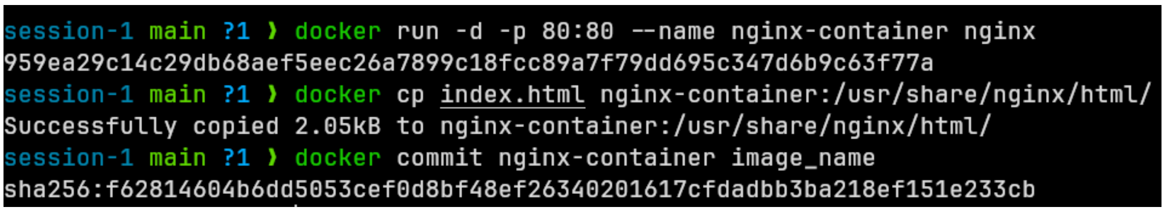


```
~ ❯ docker run -d --name app-database -e MYSQL_ROOT_PASSWORD=P4sSw0rd0! mysql:8.0
ddead207173c1c529570ab06a4110252b845609d4d804676502332954c3f24d3
```

Figure 3: Answer

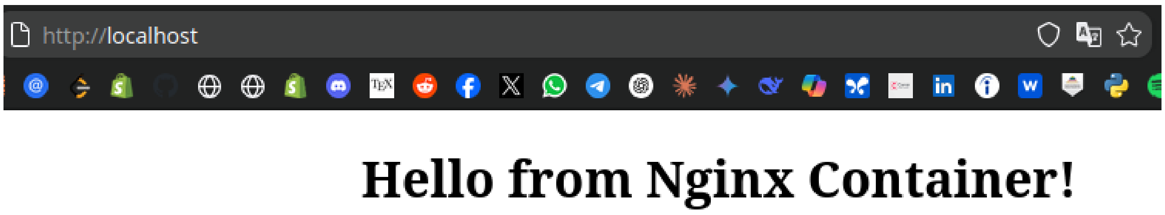
0.5 Question 5

- Run the image Nginx
- Add html static files to the container and make sure they are accessible
- Commit the container with image name `IMAGE_NAME`



```
session-1 main ?1 ❯ docker run -d -p 80:80 --name nginx-container nginx
959ea29c14c29db68aef5eec26a7899c18fcc89a7f79dd695c347d6b9c63f77a
session-1 main ?1 ❯ docker cp index.html nginx-container:/usr/share/nginx/html/
Successfully copied 2.05kB to nginx-container:/usr/share/nginx/html/
session-1 main ?1 ❯ docker commit nginx-container image_name
sha256:f62814604b6dd5053cef0d8bf48ef26340201617cfdadbb3ba218ef151e233cb
```

Figure 4: Commands



http://localhost

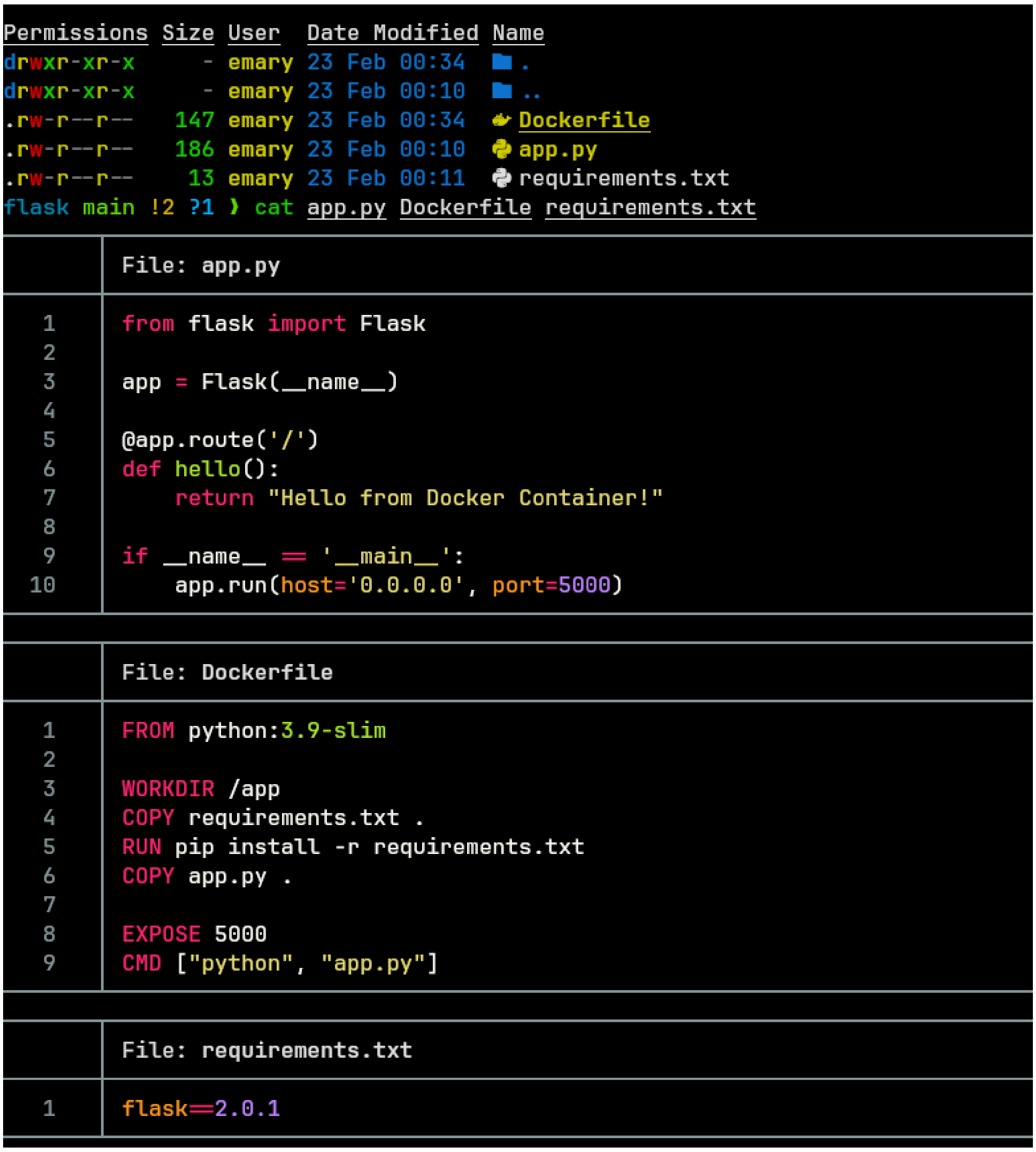
Hello from Nginx Container!

Figure 5: In Browser

0.6 Question 6

- Create a python simple app
- Create a dockerfile to containerize the python app
- Build the image and test it
- (Bonus) create a dockerfile for the same app in smaller size using multi staging
- Push the created imageinto your docker hub repo

Question 6



```
Permissions Size User  Date Modified Name
drwxr-xr-x     - emary 23 Feb 00:34  .
drwxr-xr-x     - emary 23 Feb 00:10  ..
.rw-r--r--   147 emary 23 Feb 00:34  Dockerfile
.rw-r--r--   186 emary 23 Feb 00:10  app.py
.rw-r--r--    13 emary 23 Feb 00:11  requirements.txt
flask main !2 ?1 ❯ cat app.py Dockerfile requirements.txt
```

```
File: app.py
```

```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello from Docker Container!"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

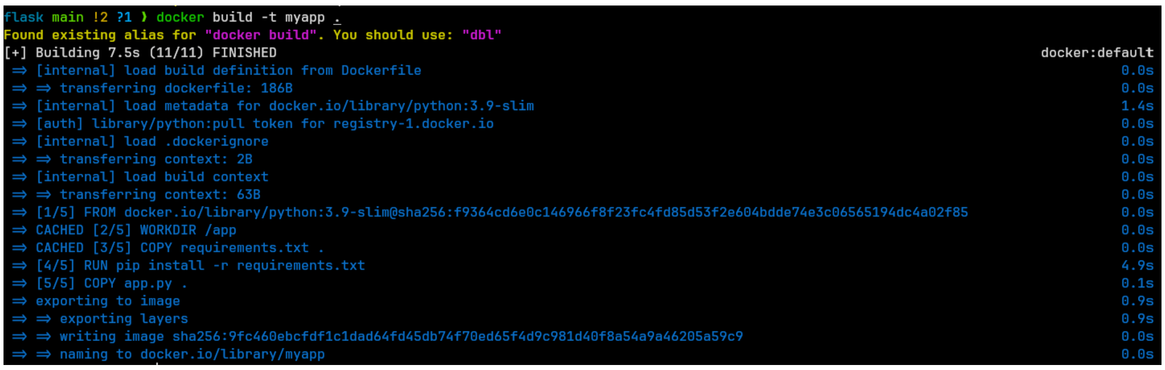
```
File: Dockerfile
```

```dockerfile
FROM python:3.9-slim

WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY app.py .

EXPOSE 5000
CMD ["python", "app.py"]
```

```
File: requirements.txt
```

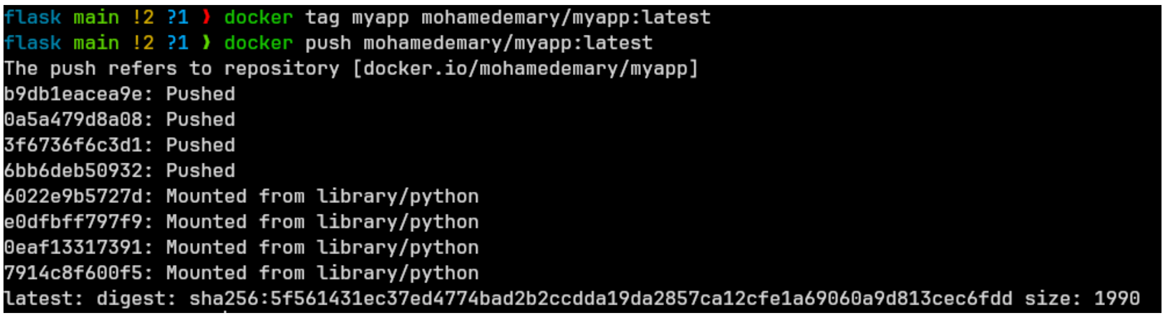
```
flask==2.0.1
```

Figure 6: Files & Their Content



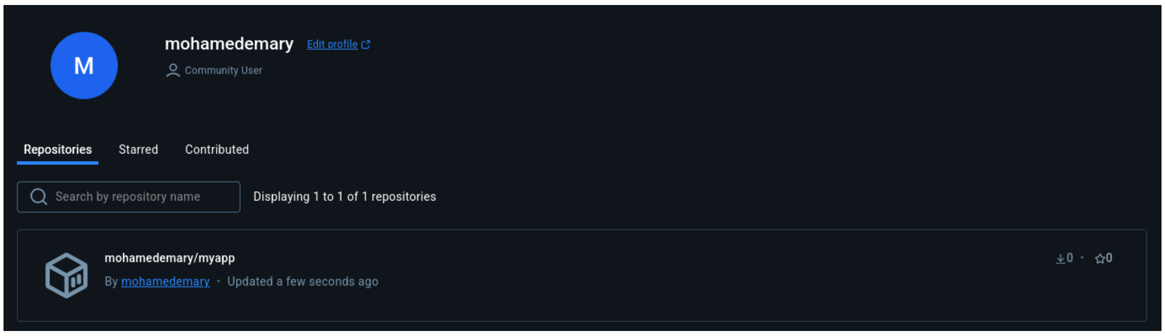
```
flask main !2 ?1 ❯ docker build -t myapp .
Found existing alias for "docker build". You should use: "dbl"
[+] Building 7.5s (11/11) FINISHED                                                                            docker:default
 => [internal] load build definition from Dockerfile                                                                    0.0s
 => => transferring dockerfile: 186B                                                                                    0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim                                                      1.4s
 => [auth] library/python:pull token for registry-1.docker.io                                                           0.0s
 => [internal] load .dockerignore                                                                                       0.0s
 => => transferring context: 2B                                                                                         0.0s
 => [internal] load build context                                                                                       0.0s
 => => transferring context: 63B                                                                                        0.0s
 => [1/5] FROM docker.io/library/python:3.9-slim@sha256:f9364cd6e0c146966f8f23fc4fd85d53f2e604bdde74e3c06565194dc4a02f85   0.0s
 => CACHED [2/5] WORKDIR /app                                                                                           0.0s
 => CACHED [3/5] COPY requirements.txt .                                                                                0.0s
 => [4/5] RUN pip install -r requirements.txt                                                                           4.9s
 => [5/5] COPY app.py .                                                                                                 0.1s
 => exporting to image                                                                                                  0.9s
 => => exporting layers                                                                                                 0.9s
 => => writing image sha256:9fc460ebcfdf1c1dad64fd45db74f70ed65f4d9c981d40f8a54a9a46205a59c9                            0.0s
 => => naming to docker.io/library/myapp                                                                                0.0s
```

Figure 7: Building The Image



```
flask main !2 ?1 ❯ docker tag myapp mohamedemary/myapp:latest
flask main !2 ?1 ❯ docker push mohamedemary/myapp:latest
The push refers to repository [docker.io/mohamedemary/myapp]
b9db1eacea9e: Pushed
0a5a479d8a08: Pushed
3f6736f6c3d1: Pushed
6bb6deb50932: Pushed
6022e9b5727d: Mounted from library/python
e0dfbff797f9: Mounted from library/python
0eaf13317391: Mounted from library/python
7914c8f600f5: Mounted from library/python
latest: digest: sha256:5f561431ec37ed4774bad2b2ccdda19da2857ca12cfe1a69060a9d813cec6fdd size: 1990
```

Figure 8: Pushing Image To Docker Hub



mohamedemary Edit profile
Community User

Displaying 1 to 1 of 1 repositories

mohamedemary/myapp
By mohamedemary · Updated a few seconds ago
0 · 0

Figure 9: Screenshot of the image pushed to Docker Hub