

### Lab3:

- Create a dockerfile using vi editor  
*FROM debian:stable*  
*LABEL authors="amit"*  
*RUN apt-get update && apt-get install -y --force-yes apache2*  
*EXPOSE 80*
- Save the dockerfile
- `docker built -t mydeb .`

```
root@ubuntu20:~# cat dockerfile
FROM debian:stable
LABEL authors="amit"
RUN apt-get update && apt-get install -y --force-yes apache2
EXPOSE 80
root@ubuntu20:~# docker build -t mydeb .
Sending build context to Docker daemon 16.9kB
Step 1/4 : FROM debian:stable
stable: Pulling from library/debian
0b5aea898977: Pull complete
Digest: sha256:3d2aa501c4cefd4415895b1d877dfbba0739cab1d58cbe8f1baa3f01b6739690
Status: Downloaded newer image for debian:stable
---> f70ab914d71a
Step 2/4 : LABEL authors="amit"
---> Running in 1589a5adb5d7
Removing intermediate container 1589a5adb5d7
---> fd93cc136079
Step 3/4 : RUN apt-get update && apt-get install -y --force-yes apache2
---> Running in 5f23b0f14c23
```

### *docker images*

#### *docker inspect <image id>*

```
root@ubuntu20:~# docker inspect mydeb
[
  {
    "Id": "sha256:8d31f53b52b571cf8dcd89a3f8b4cb8277d51b65f8e1648fae780b88fef249d7",
    "RepoTags": [
      "mydeb:latest"
    ],
    "RepoDigests": [],
    "Parent": "sha256:4e7e1fb5d2fae5671b561af502fb29f786b06143f26b127a4fddb44f3427d0a5",
    "Comment": "",
    "Created": "2022-09-09T06:47:15.205660575Z",
    "Container": "aec752f4b684eef6af78fea2f93346dba843bc9f6771c9f22bba3b92f0b788",
    "ContainerConfig": {
      "Hostname": "aec752f4b684",
      "Domainname": "",
      "User": "",
      "AttachStdin": false,
      "AttachStdout": false,
      "AttachStderr": false
```

### CMD, RUN and ENTRYPOINT

To start, let's create a script, *log-event.sh*. It simply adds one line to a file and then prints it:

```
#!/bin/sh
echo `date` $@ >> log.txt;
cat log.txt;

mkdir test
cd test
vi date.sh
#copy and paste the above bash script
```

```
root@ubuntu20:~# vi date.sh
root@ubuntu20:~# chmod 777 date.sh
root@ubuntu20:~# ./date.sh
Fri Sep 9 07:11:29 UTC 2022
root@ubuntu20:~# cat date.sh
#!/bin/sh
echo `date` $@ >> log.txt;
cat log.txt;

root@ubuntu20:~#
```

---

Lets add the run command in dockerfile

```
FROM alpine
ADD date.sh /
RUN ["/date.sh", "image created"]
```

```
root@ubuntu20:~/test# docker build -t myimage .
Sending build context to Docker daemon 4.096kB
Step 1/3 : FROM alpine
latest: Pulling from library/alpine
213ec9aee27d: Pull complete
Digest: sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad
Status: Downloaded newer image for alpine:latest
--> 9c6f07244728
Step 2/3 : ADD date.sh /
--> 525a220915dc
Step 3/3 : RUN ["/date.sh", "image created"]
--> Running in 95f1c646bb91
```

*docker run myimage cat log.txt*

Check the time

```

root@ubuntu20:~/test# docker run myimage cat log.txt
Fri Sep 9 07:15:44 UTC 2022 image created
root@ubuntu20:~/test# date
Fri Sep 9 07:31:39 UTC 2022
root@ubuntu20:~/test# █

```

If we run the container several times, we'll see that the date in our log file doesn't change. This makes sense because the **run step executes at image build time**, not at the container runtime.

Modify the docker file and RUN command now

```

FROM alpine
ADD date.sh /
RUN ["/date.sh", "image created"]
CMD ["/date.sh", "container started"]

```

```
docker build -t myimage1 .
```

```

root@ubuntu20:~/test# docker run myimage1
Fri Sep 9 07:15:44 UTC 2022 image created
Fri Sep 9 07:34:37 UTC 2022 container started
root@ubuntu20:~/test# docker run myimage1
Fri Sep 9 07:15:44 UTC 2022 image created
Fri Sep 9 07:34:40 UTC 2022 container started
root@ubuntu20:~/test# cat dockerfile

```

If we run this multiple times, we'll see that the *image created* entry stays the same. But the *container started* entry updates with every run. This shows how *cmd* indeed executes every time the container starts.

If we run the older command

```
docker run myimage1 cat log.txt
```

```

root@ubuntu20:~/test# docker run myimage1 cat log.txt
Fri Sep 9 07:15:44 UTC 2022 image created
root@ubuntu20:~/test# █

```

This time the *cmd* specified in the Dockerfile is ignored. That's because we have specified arguments to the *docker run* command.

Lets add two commands:

```
root@ubuntu20:~/test# vi dockerfile
FROM alpine
ADD date.sh /
RUN ["/date.sh", "image created"]
CMD ["/date.sh", "container started"]
CMD ["/date.sh", "container running"]
```

docker build myimage2 .

```
root@ubuntu20:~/test# docker run myimage2
Fri Sep 9 07:15:44 UTC 2022 image created
Fri Sep 9 07:42:15 UTC 2022 container running
```

That's because **only the last cmd is invoked if more than one is specified.**

## ENTRYPOINT

```
FROM alpine
ADD date.sh /
RUN ["/date.sh", "image created"]
ENTRYPOINT ["/date.sh"]
```

docker build -t myimage3 .

```
root@ubuntu20:~/test# docker build -t myimage3 .
Sending build context to Docker daemon 4.096kB
Step 1/4 : FROM alpine
---> 9c6f07244728
Step 2/4 : ADD date.sh /
---> Using cache
---> 525a220915dc
Step 3/4 : RUN ["/date.sh", "image created"]
---> Using cache
---> 03858e7ae8a1
Step 4/4 : ENTRYPOINT ["/date.sh"]
---> Running in 9d53c1dd81cf
Removing intermediate container 9d53c1dd81cf
---> 0e8a06d7be41
Successfully built 0e8a06d7be41
Successfully tagged myimage3:latest
```

*docker run myimage3 welcome to container*

```
root@ubuntu20:~/test# docker run myimage3 welcome to container
Fri Sep 9 07:15:44 UTC 2022 image created
Fri Sep 9 07:49:15 UTC 2022 welcome to container
root@ubuntu20:~/test# █
```

Assignment: Try CMD and ENTRYPOINT together and let me know the observation

Tag and push

*docker login*

```
root@ubuntu20:~/test# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head to https://docs.docker.com/engine/reference/commandline/login/#credentials-store to create one.
Username: amitow
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

*docker image tag myimage amitow/myimage*

***docker image push amitow/myimage***

```
root@ubuntu20:~/test# docker image push amitow/myimage
Using default tag: latest
The push refers to repository [docker.io/amitow/myimage]
07d0f861a6d3: Pushed
8c8553f0e37a: Pushed
994393dc58e7: Mounted from library/alpine
```