

Dockerfile

- RUN, CMD and Entrypoint can be written in two forms
 - shell : here we directly write linux commands
 - exec: here we express commands in []

CMD vs ENTRYPOINT:

- CMD
- version 1

```
FROM alpine
CMD ["echo", "hello"]
```

```
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker container run --name trail1 cmd:demo
hello
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
5d2d62a68db2   cmd:demo   "echo hello"   10 seconds ago   Exited (0) 9 seconds ago   trail1
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker container run -d --name trail2 cmd:demo
1f1c8ce8feaa980efe415429c6a4145f1922f14608dffc7b29af5afb2c6db9e9
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker container run -d --name trail3 cmd:demo ping google.com
46205a52fe347996d42e7b4fe14e4edb44bd9b9f6fd5d213a52a20f83736bcd
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
46205a52fe34   cmd:demo   "ping google.com"   3 seconds ago   Up 2 seconds   trail3
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
46205a52fe34   cmd:demo   "ping google.com"   9 seconds ago   Up 8 seconds   trail3
1f1c8ce8feaa   cmd:demo   "echo hello"       49 seconds ago   Exited (0) 49 seconds ago   trail2
5d2d62a68db2   cmd:demo   "echo hello"       About a minute ago   Exited (0) About a minute ago   trail1
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ |
```

- version 2

```
FROM alpine
ENTRYPOINT ["echo"]
CMD ["hello"]
```

```

ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ vi Dockerfile
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker image build -t entryncmd:demo .
[+] Building 1.4s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 83B
=> [internal] load metadata for docker.io/library/alpine:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> CACHED [1/1] FROM docker.io/library/alpine:latest@sha256:77726ef6b57dd6f5bb551896826ec38bc3e53f75cddde31354fbff4f25238ebd
=> exporting to image
=> => exporting layers
=> => writing image sha256:065c5188b4af67598d32981c245d156d44fa75509c0793239c75b8f4d338b8c8
=> => naming to docker.io/library/entryncmd:demo
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker run --name trail1 entryncmd:demo
hello
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker run --name trail2 entryncmd:demo Devops
Devops
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
194343eb9821   entryncmd:demo "echo Devops"           20 seconds ago Exited (0) 18 seconds ago           trail2
0fc44d89adda   entryncmd:demo "echo hello"            26 seconds ago Exited (0) 25 seconds ago           trail1
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker run -d --name trail3 --entrypoint ping entryncmd:demo google.com
9f060c7c114456681e4971228a373c5d7fae0a2f5b51126b68992425d7fb6207
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
9f060c7c1144   entryncmd:demo "ping google.com"       3 seconds ago   Up 2 seconds              trail3
194343eb9821   entryncmd:demo "echo Devops"           About a minute ago Exited (0) About a minute ago           trail2
0fc44d89adda   entryncmd:demo "echo hello"            About a minute ago Exited (0) About a minute ago           trail1
ubuntu@ip-172-31-36-74:~/cmdnentrypoint$

```

Image: scratch

- Dockerfile

```

FROM scratch
CMD ["ls"]

```

- Scratch represents nothing. when we use scratch copying os into container should be done [Refer Here](#) and [Refer Here](#)

ENV

- ENV instruction is used to set environment variables [Refer Here](#)
- ENV can be passed while creating containers
- env example

```

FROM alpine
ENV APP="test"
CMD ["sleep", "1d"]

```

```

learningthoughts@docker-machine:~/test$ docker run -d --name envexp envdemo
0486321f71624da464470b69035c2fd9e32f32eae21e6f82572f21110aed2f89
learningthoughts@docker-machine:~/test$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS              NAMES
0486321f7162   envdemo  "sleep 1d"              3 seconds ago       Up 2 seconds              envexp
learningthoughts@docker-machine:~/test$ docker container exec envexp printenv
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=0486321f7162
APP=test
HOME=/root
learningthoughts@docker-machine:~/test$ docker run -d --name envexp1 -e "APP=nginx" envdemo
5555cea1a8543538e4f8dc8581c4bc5c201b2cd67ba5b00be68f75675441c1c5
learningthoughts@docker-machine:~/test$ docker container exec envexp1 printenv
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=5555cea1a854
APP=nginx
HOME=/root
learningthoughts@docker-machine:~/test$

```

Docker image layers

- [Refer Here](#) this article
- Impact: whenever a container is deleted the writable layer will be deleted (this may lead to data loss). Docker has a concept of volumes to avoid data loss.

database container

- Lets run mysql database in a container

```

docker container run -d -P -e 'MYSQL_ROOT_PASSWORD=admin' -e
'MYSQL_DATABASE=author' -e 'MYSQL_USER=ltdevops' -e 'MYSQL_PASSWORD=admin' --name
authordb mysql:8.0-debian

```

```

learningthoughts@docker-machine:~$ docker container run -d -P -e 'MYSQL_ROOT_PASSWORD=admin' -e 'MYSQL_DATABASE=a
uthor' -e 'MYSQL_USER=ltdevops' -e 'MYSQL_PASSWORD=admin' --name authordb mysql:8.0-debian
Unable to find image 'mysql:8.0-debian' locally
8.0-debian: Pulling from library/mysql
09f376ebb190: Pull complete
029cd480822b: Pull complete
225aa208e69a: Pull complete
d239b7cd9357: Pull complete
3a0ff6da3549: Pull complete
ae062e186090: Pull complete
040a0a6075c4: Pull complete
efe50e9a5499: Pull complete
b8e648b481d7: Pull complete
c8839099acf6: Pull complete
d5e3c8c90598: Pull complete
13925c5104b8: Pull complete
Digest: sha256:1579fe3a97a436cc10824fc771a07fcedc92213e7ab7604eb5d2976ca419abc8
Status: Downloaded newer image for mysql:8.0-debian
e52f7c606b42c48b36b115ecb89d24d3428f7910517b49d75ee4a409560226e0
learningthoughts@docker-machine:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS              NAMES
e52f7c606b42   mysql:8.0-debian  "docker-entrypoint.s..."  4 seconds ago       Up 2 seconds              0.0.0.0:32769->3306/t
cp, :::32769->3306/tcp, 0.0.0.0:32768->33060/tcp, :::32768->33060/tcp  authordb
5555cea1a854   envdemo  "sleep 1d"              10 minutes ago       Up 10 minutes              envexp1
0486321f7162   envdemo  "sleep 1d"              11 minutes ago       Up 11 minutes              envexp
learningthoughts@docker-machine:~$

```

- Now let's execute in an interactive mode using mysql client

```
learningthoughts@docker-machine:~$ docker container exec -it authordb mysql -u ltdevops -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.37 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

- [Refer Here](#) for create and insert commands

```
mysql>
mysql> use author;
Database changed
mysql> CREATE TABLE authors (id INT, name VARCHAR(20), email VARCHAR(20));
Query OK, 0 rows affected (0.14 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_author |
+-----+
| authors           |
+-----+
1 row in set (0.00 sec)

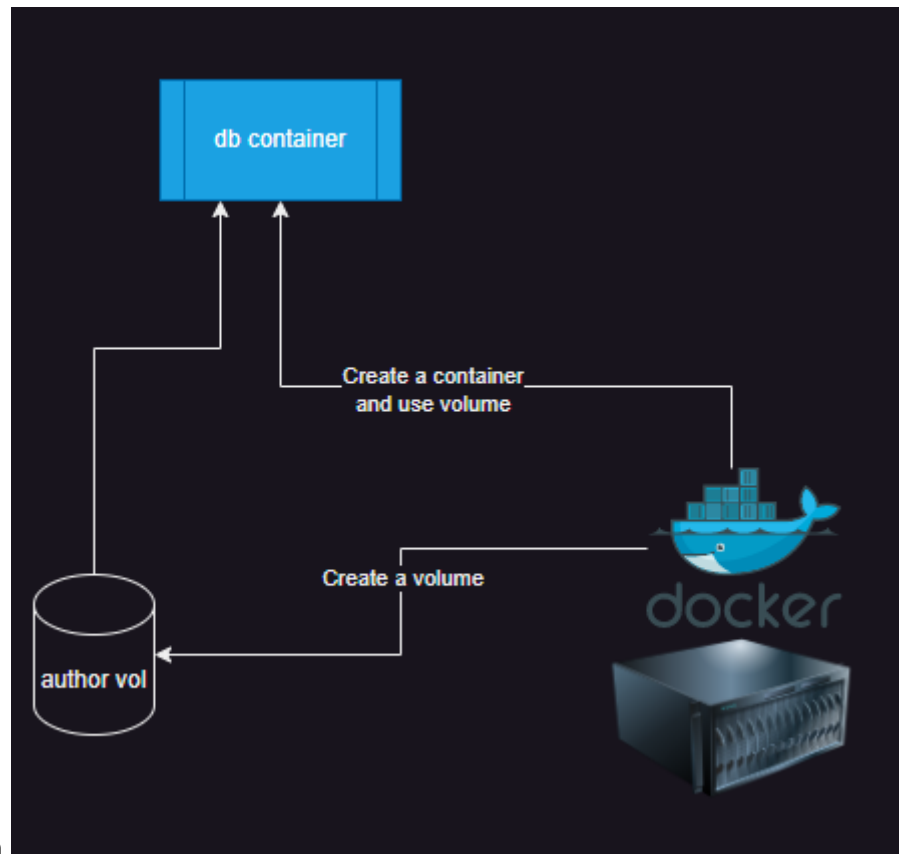
mysql> INSERT INTO authors (id,name,email) VALUES(1,"Vivek","xuz@abc.com");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO authors (id,name,email) VALUES(2,"Priya","p@gmail.com");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO authors (id,name,email) VALUES(3,"Tom","tom@yahoo.com");
Query OK, 1 row affected (0.05 sec)

mysql> select * from authors;
+----+-----+-----+
| id  | name  | email      |
+----+-----+-----+
| 1   | Vivek | xuz@abc.com |
| 2   | Priya | p@gmail.com |
| 3   | Tom   | tom@yahoo.com |
+----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```



- To be discussed in next session