Lab Exercise 5- Understanding CMD, RUN, and ENTRYPOINT in Dockerfile

Name:- Vansh Bhatt

Sap ID:- 500125395

Batch:- DevOps B1

To:- Hitesh Sharma Sir

Objective:

To learn the differences between CMD, RUN, and ENTRYPOINT instructions in Dockerfiles by creating and running Docker containers with different configurations.

Prerequisites:

- Docker installed on your machine
- Basic understanding of Docker and Dockerfile

Part 1: Overview of CMD, RUN, and ENTRYPOINT

- **RUN:** Executes commands at build time to install software, download dependencies, or configure the environment. The result is saved in the image.
- **CMD:** Specifies the default command to be executed when a container starts. It can be overridden when running a container.

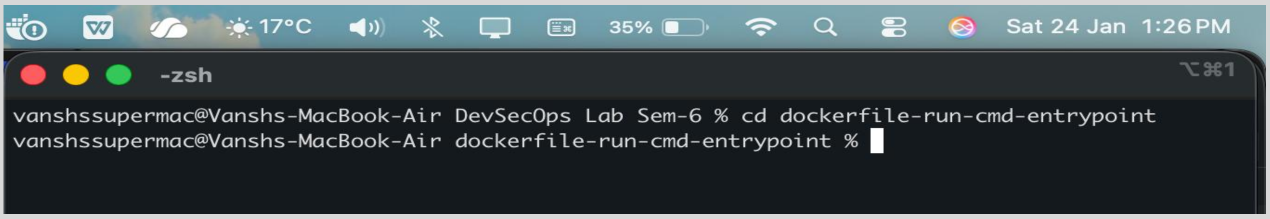
- **ENTRYPOINT:** Defines the main executable for the container, which can't be easily overridden. However, additional arguments can be passed when the container starts.

Part 2: Exploring RUN Command

1. **Create a Dockerfile with RUN:**

Create a directory called dockerfile-run-cmd-entrypoint and navigate to it:

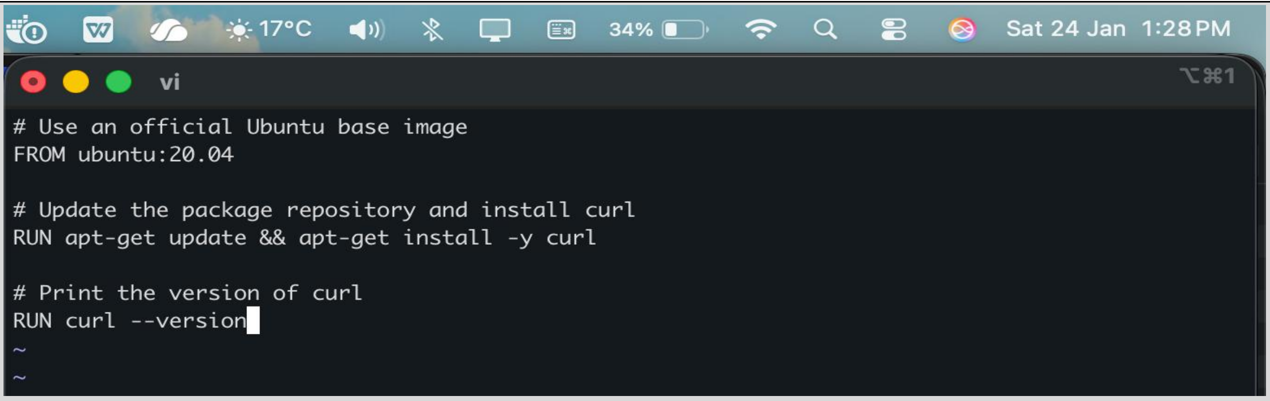
```
mkdir dockerfile-run-cmd-entrypoint && cd dockerfile-run-cmd-entrypoint
```



```
vanshssupermac@Vanshs-MacBook-Air DevSecOps Lab Sem-6 % cd dockerfile-run-cmd-entrypoint
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

Create a simple Dockerfile that uses the RUN instruction:

```
# Use an official Ubuntu base image

FROM ubuntu:20.04

# Update the package repository and install curl

RUN apt-get update && apt-get install -y curl

# Print the version of curl

RUN curl --version
```

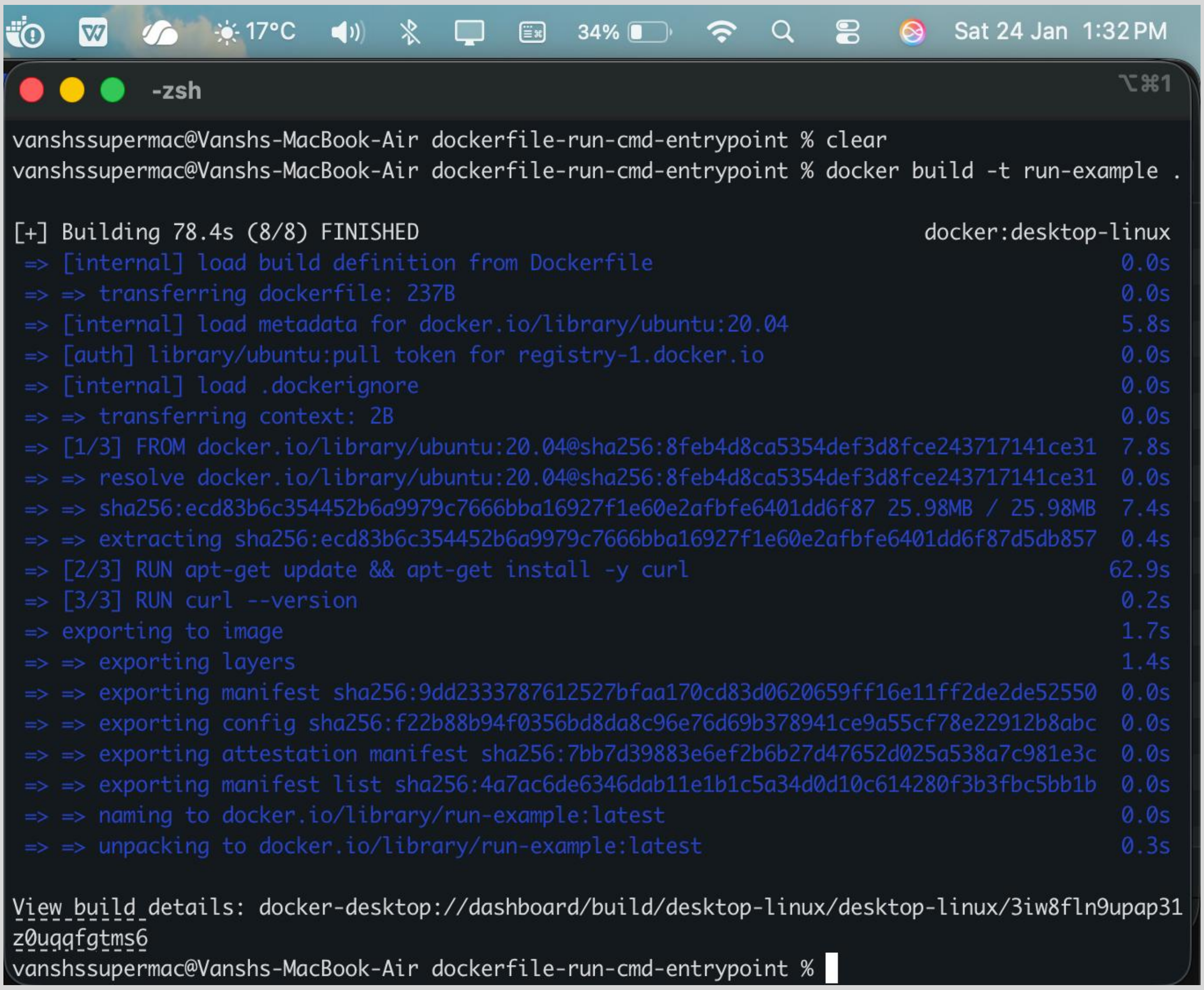


```
# Use an official Ubuntu base image
FROM ubuntu:20.04

# Update the package repository and install curl
RUN apt-get update && apt-get install -y curl

# Print the version of curl
RUN curl --version
~
~
```

2. **Build the Docker Image:**

Build the image using the Dockerfile:

```
docker build -t run-example .
```



```
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % clear
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker build -t run-example .

[+] Building 78.4s (8/8) FINISHED                                            docker:desktop-linux
 => [internal] load build definition from Dockerfile                                          0.0s
 => => transferring dockerfile: 237B                                                          0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                               5.8s
 => [auth] library/ubuntu:pull token for registry-1.docker.io                                 0.0s
 => [internal] load .dockerignore                                                             0.0s
 => => transferring context: 2B                                                               0.0s
 => [1/3] FROM docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31      7.8s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31      0.0s
 => => sha256:ecd83b6c354452b6a9979c7666bba16927f1e60e2afbfe6401dd6f87 25.98MB / 25.98MB      7.4s
 => => extracting sha256:ecd83b6c354452b6a9979c7666bba16927f1e60e2afbfe6401dd6f87d5db857      0.4s
 => [2/3] RUN apt-get update && apt-get install -y curl                                      62.9s
 => [3/3] RUN curl --version                                                                  0.2s
 => exporting to image                                                                        1.7s
 => => exporting layers                                                                       1.4s
 => => exporting manifest sha256:9dd2333787612527bfaa170cd83d0620659ff16e11ff2de2de52550      0.0s
 => => exporting config sha256:f22b88b94f0356bd8da8c96e76d69b378941ce9a55cf78e22912b8abc      0.0s
 => => exporting attestation manifest sha256:7bb7d39883e6ef2b6b27d47652d025a538a7c981e3c      0.0s
 => => exporting manifest list sha256:4a7ac6de6346dab11e1b1c5a34d0d10c614280f3b3fbc5bb1b      0.0s
 => => naming to docker.io/library/run-example:latest                                         0.0s
 => => unpacking to docker.io/library/run-example:latest                                      0.3s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/3iw8fln9upap31
z0uqqfgtms6
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

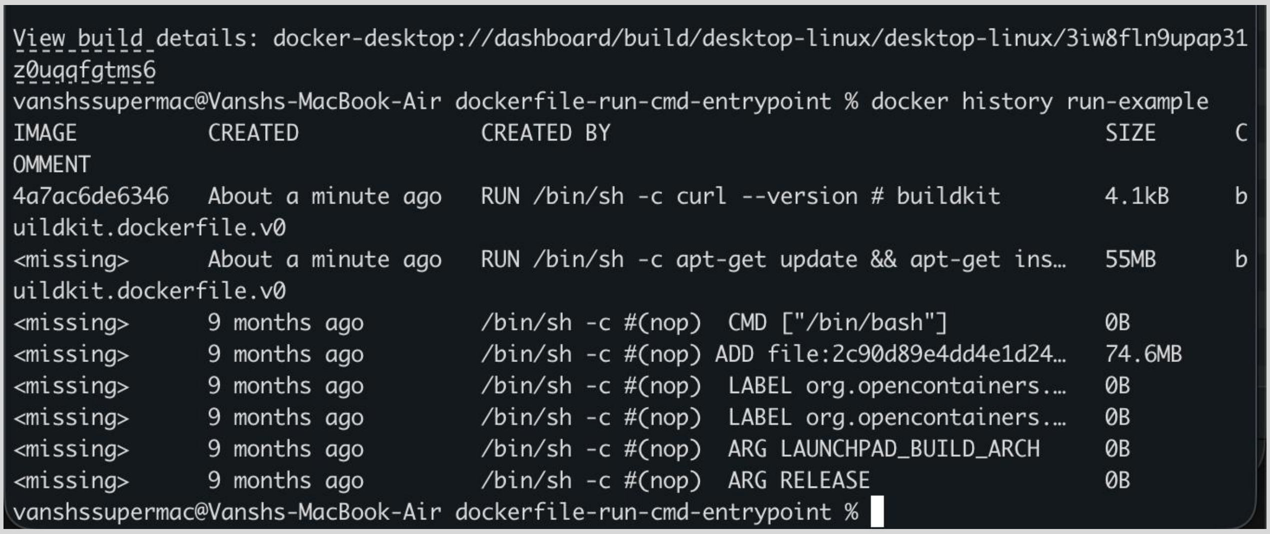
3. **Explanation:**

The RUN commands in this Dockerfile are executed during the image build process. The first RUN installs curl, and the second RUN command checks and prints the curl version. After the image is built, the commands executed by RUN are already baked into the image.

4. **Verify with Docker History:**

You can check the layers created by RUN using:

```
docker history run-example
```



```
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/3iw8fln9upap31
z0uqqfgtms6
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker history run-example
IMAGE          CREATED              CREATED BY                                      SIZE      C
OMMENT
4a7ac6de6346   About a minute ago   RUN /bin/sh -c curl --version # buildkit        4.1kB     b
uildkit.dockerfile.v0
<missing>      About a minute ago   RUN /bin/sh -c apt-get update && apt-get ins…   55MB      b
uildkit.dockerfile.v0
<missing>      9 months ago         /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>      9 months ago         /bin/sh -c #(nop) ADD file:2c90d89e4dd4e1d24…   74.6MB
<missing>      9 months ago         /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      9 months ago         /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      9 months ago         /bin/sh -c #(nop)  ARG LAUNCHPAD_BUILD_ARCH     0B
<missing>      9 months ago         /bin/sh -c #(nop)  ARG RELEASE                  0B
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

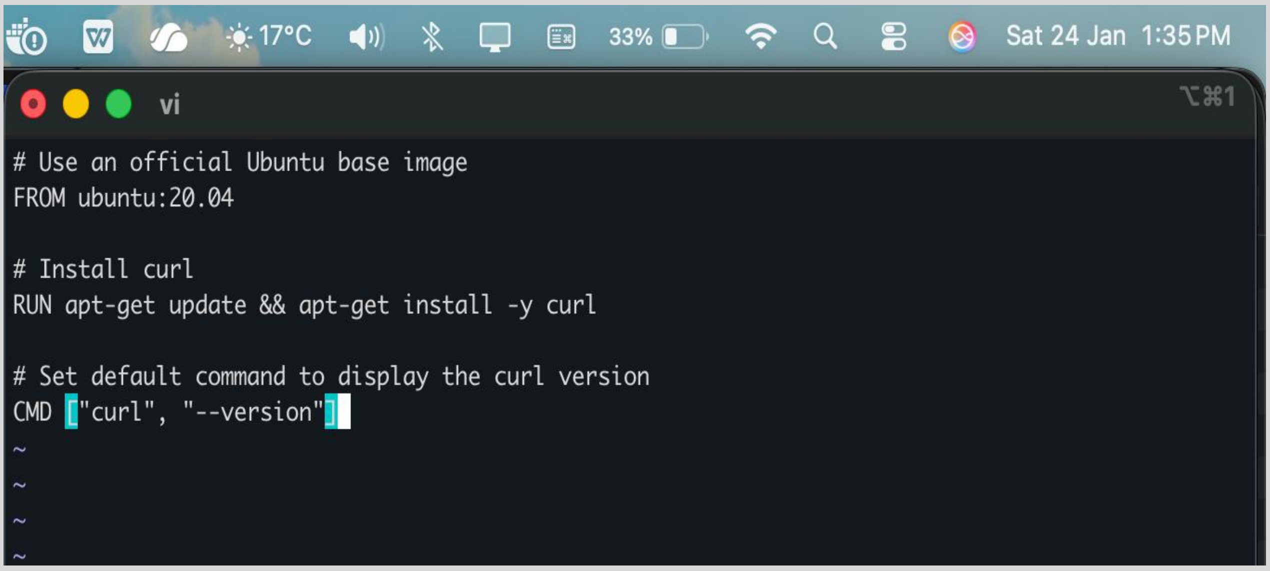
Each RUN command creates a new layer in the image.

Part 3: Exploring CMD Command

1. **Create a Dockerfile with CMD:**

Modify the Dockerfile to include the CMD instruction:

```
# Use an official Ubuntu base image
FROM ubuntu:20.04

# Install curl
RUN apt-get update && apt-get install -y curl

# Set default command to display the curl version
CMD ["curl", "--version"]
```

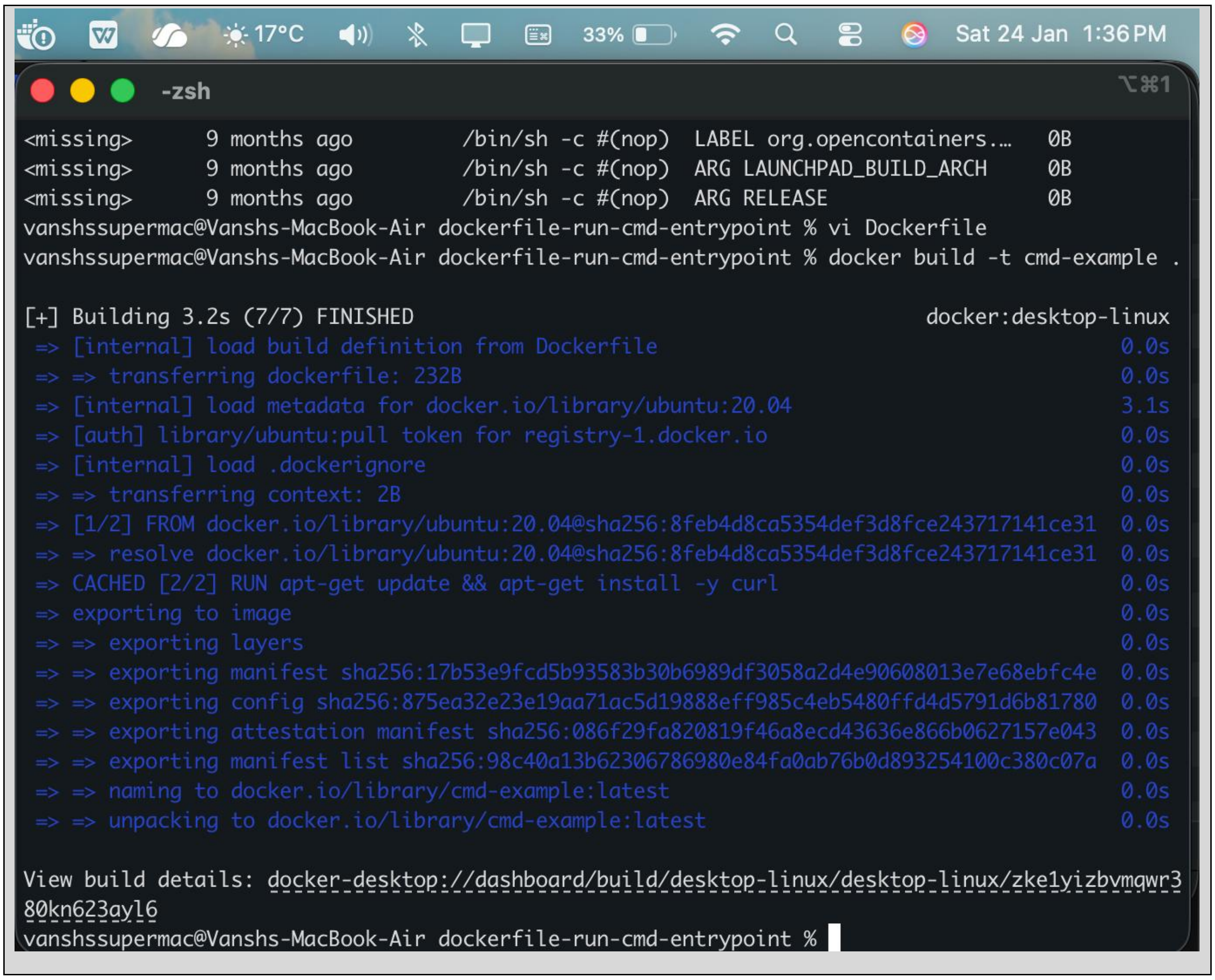


```
# Use an official Ubuntu base image
FROM ubuntu:20.04

# Install curl
RUN apt-get update && apt-get install -y curl

# Set default command to display the curl version
CMD ["curl", "--version"]
~
~
~
~
```

2. **Build the Docker Image:**

Build the Docker image again:

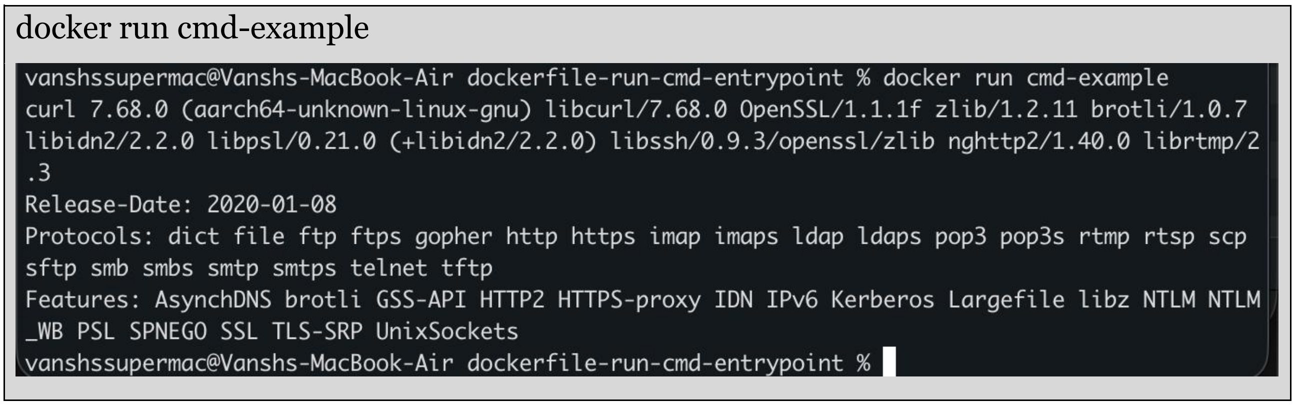
```
docker build -t cmd-example .
```



```
<missing>      9 months ago        /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      9 months ago        /bin/sh -c #(nop)  ARG LAUNCHPAD_BUILD_ARCH     0B
<missing>      9 months ago        /bin/sh -c #(nop)  ARG RELEASE                  0B
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % vi Dockerfile
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker build -t cmd-example .

[+] Building 3.2s (7/7) FINISHED                                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                                       0.0s
 => => transferring dockerfile: 232B                                                       0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                            3.1s
 => [auth] library/ubuntu:pull token for registry-1.docker.io                              0.0s
 => [internal] load .dockerignore                                                          0.0s
 => => transferring context: 2B                                                            0.0s
 => [1/2] FROM docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31  0.0s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31  0.0s
 => CACHED [2/2] RUN apt-get update && apt-get install -y curl                             0.0s
 => exporting to image                                                                     0.0s
 => => exporting layers                                                                    0.0s
 => => exporting manifest sha256:17b53e9fcd5b93583b30b6989df3058a2d4e90608013e7e68ebfc4e  0.0s
 => => exporting config sha256:875ea32e23e19aa71ac5d19888eff985c4eb5480ffd4d5791d6b81780  0.0s
 => => exporting attestation manifest sha256:086f29fa820819f46a8ecd43636e866b0627157e043  0.0s
 => => exporting manifest list sha256:98c40a13b62306786980e84fa0ab76b0d893254100c380c07a  0.0s
 => => naming to docker.io/library/cmd-example:latest                                      0.0s
 => => unpacking to docker.io/library/cmd-example:latest                                   0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/zke1yizbvmqwr3
80kn623ayl6
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

3. **Run the Container:**

Run the container and see the output:



```
docker run cmd-example
```

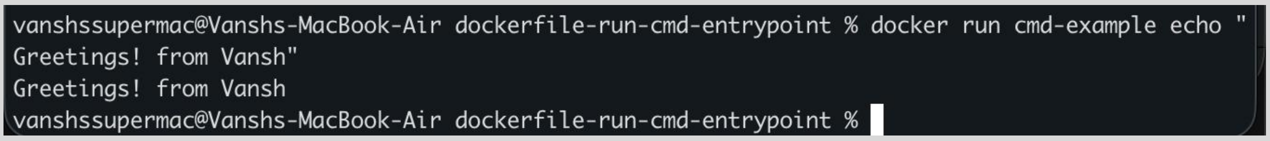
```
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker run cmd-example
curl 7.68.0 (aarch64-unknown-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 brotli/1.0.7
libidn2/2.2.0 libpsl/0.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2
.3
Release-Date: 2020-01-08
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtmp rtsp scp
sftp smb smbs smtp smtps telnet tftp
Features: AsynchDNS brotli GSS-API HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM NTLM
_WB PSL SPNEGO SSL TLS-SRP UnixSockets
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

The output will display the curl version as the default command defined by CMD is executed when the container starts.

4. **Override CMD:**

You can override the CMD by specifying a different command when you run the container:

```
docker run cmd-example echo "Hello from CMD!"
```



```
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker run cmd-example echo "
Greetings! from Vansh"
Greetings! from Vansh
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

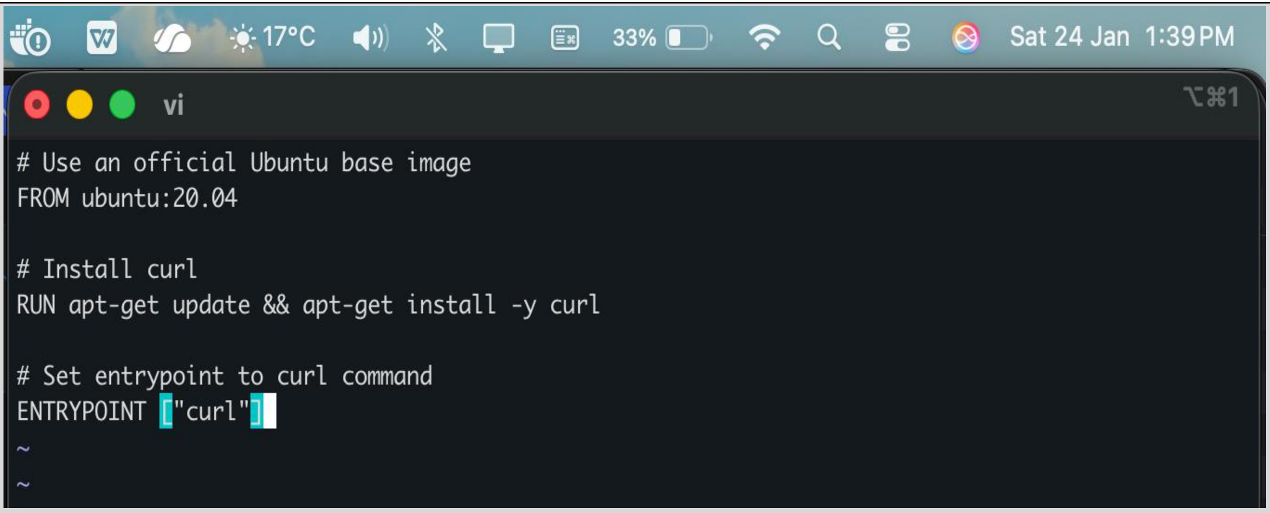
This will print Hello from CMD!, showing that the CMD can be overridden at runtime.

Part 4: Exploring ENTRYPOINT Command

1. **Create a Dockerfile with ENTRYPOINT:**

Modify the Dockerfile to use ENTRYPOINT instead of CMD:

```
# Use an official Ubuntu base image
FROM ubuntu:20.04

# Install curl
RUN apt-get update && apt-get install -y curl

# Set entrypoint to curl command
ENTRYPOINT ["curl"]
```

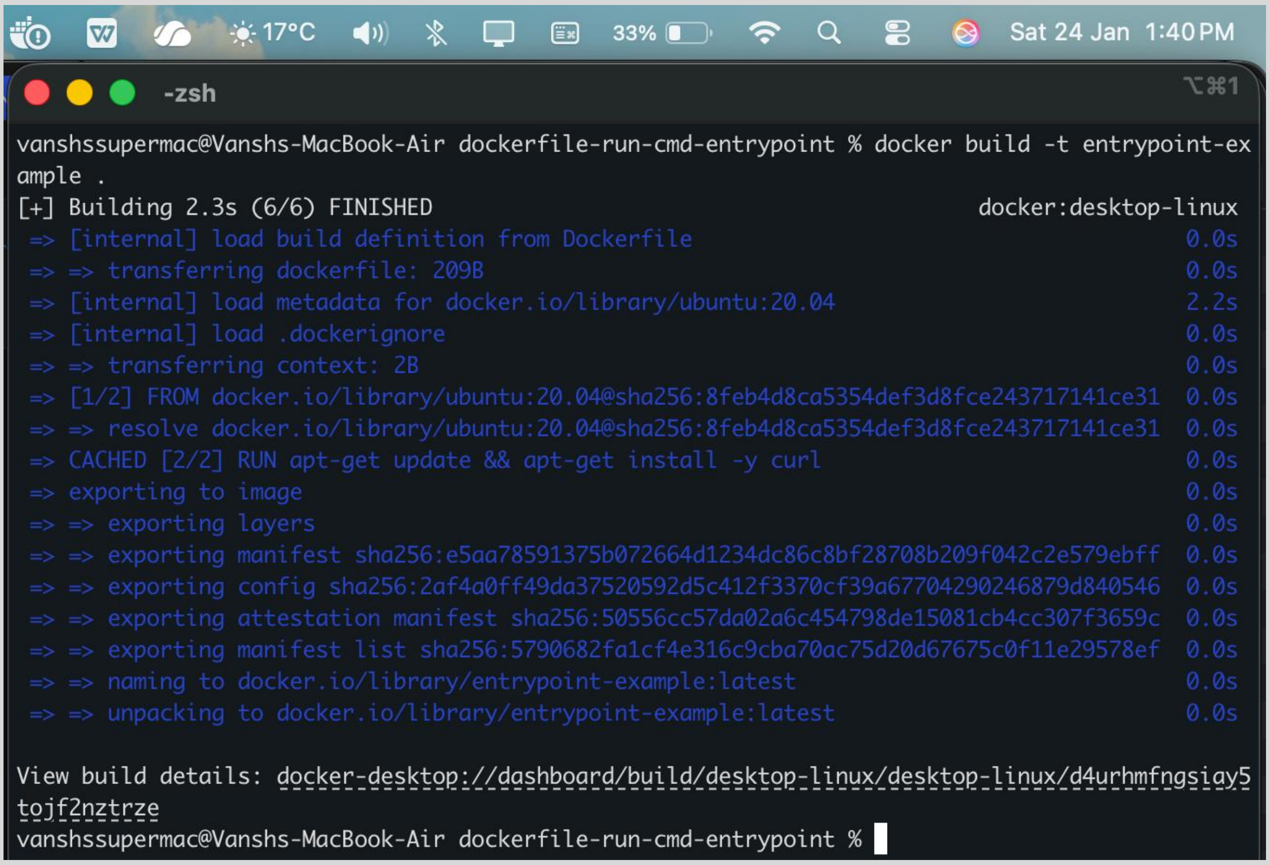


```
# Use an official Ubuntu base image
FROM ubuntu:20.04

# Install curl
RUN apt-get update && apt-get install -y curl

# Set entrypoint to curl command
ENTRYPOINT ["curl"]
~
~
```

2. **Build the Docker Image:**

Build the image with the ENTRYPOINT instruction:

```
docker build -t entrypoint-example .
```

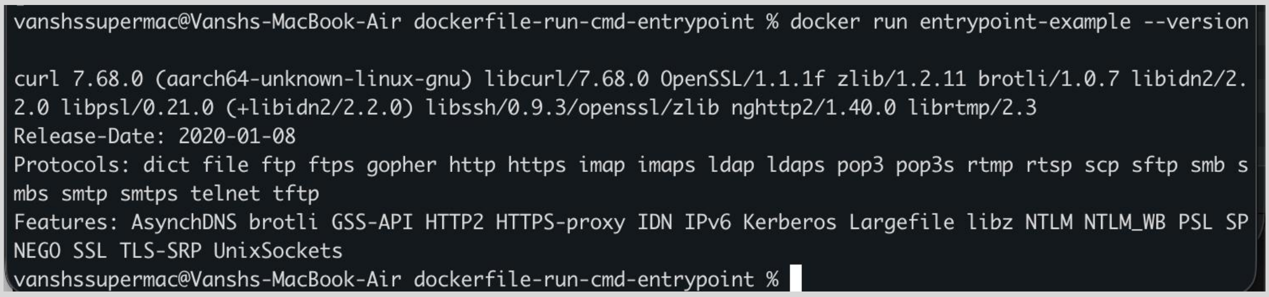


```
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker build -t entrypoint-ex
ample .
[+] Building 2.3s (6/6) FINISHED                                             docker:desktop-linux
 => [internal] load build definition from Dockerfile                                         0.0s
 => => transferring dockerfile: 209B                                                         0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                              2.2s
 => [internal] load .dockerignore                                                            0.0s
 => => transferring context: 2B                                                              0.0s
 => [1/2] FROM docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31    0.0s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31    0.0s
 => CACHED [2/2] RUN apt-get update && apt-get install -y curl                               0.0s
 => exporting to image                                                                       0.0s
 => => exporting layers                                                                      0.0s
 => => exporting manifest sha256:e5aa78591375b072664d1234dc86c8bf28708b209f042c2e579ebff    0.0s
 => => exporting config sha256:2af4a0ff49da37520592d5c412f3370cf39a67704290246879d840546    0.0s
 => => exporting attestation manifest sha256:50556cc57da02a6c454798de15081cb4cc307f3659c    0.0s
 => => exporting manifest list sha256:5790682fa1cf4e316c9cba70ac75d20d67675c0f11e29578ef    0.0s
 => => naming to docker.io/library/entrypoint-example:latest                                 0.0s
 => => unpacking to docker.io/library/entrypoint-example:latest                              0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/d4urhmfngsiay5
tojf2nztrze
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

3. **Run the Container:**

When you run the container, since ENTRYPOINT is set to curl, you need to provide arguments to the curl command:

```
docker run entrypoint-example --version
```



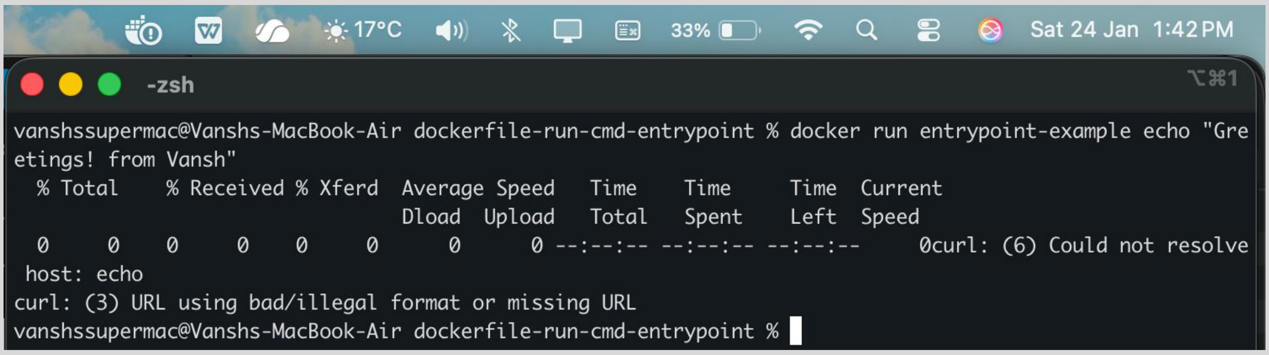
```
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker run entrypoint-example --version

curl 7.68.0 (aarch64-unknown-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 brotli/1.0.7 libidn2/2.
2.0 libpsl/0.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2.3
Release-Date: 2020-01-08
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtmp rtsp scp sftp smb s
mbs smtp smtps telnet tftp
Features: AsynchDNS brotli GSS-API HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM NTLM_WB PSL SP
NEGO SSL TLS-SRP UnixSockets
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

This will print the curl version because ENTRYPOINT defines the main executable (in this case, curl) and --version is passed as an argument to curl.

4. **Override ENTRYPOINT:**

Unlike CMD, the ENTRYPOINT is not easily overridden. If you try to override it using:

```
docker run entrypoint-example echo "Hello from ENTRYPOINT!"
```

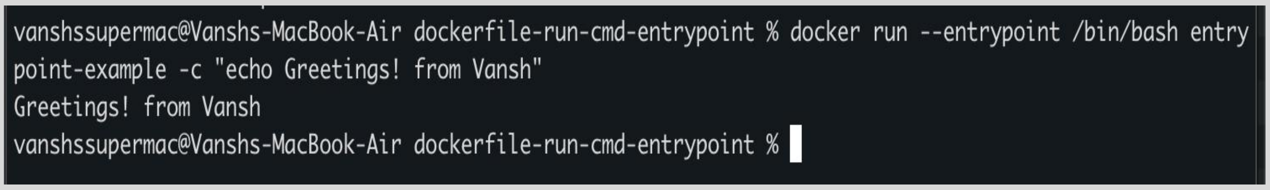


```
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker run entrypoint-example echo "Gre
etings! from Vansh"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0curl: (6) Could not resolve
 host: echo
curl: (3) URL using bad/illegal format or missing URL
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

It will result in an error because curl will interpret echo as an argument.

However, you can use the --entrypoint option to change the entrypoint:

```
docker run --entrypoint /bin/bash entrypoint-example -c "echo Hello from ENTRYPOINT!"
```



```
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker run --entrypoint /bin/bash entry
point-example -c "echo Greetings! from Vansh"
Greetings! from Vansh
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

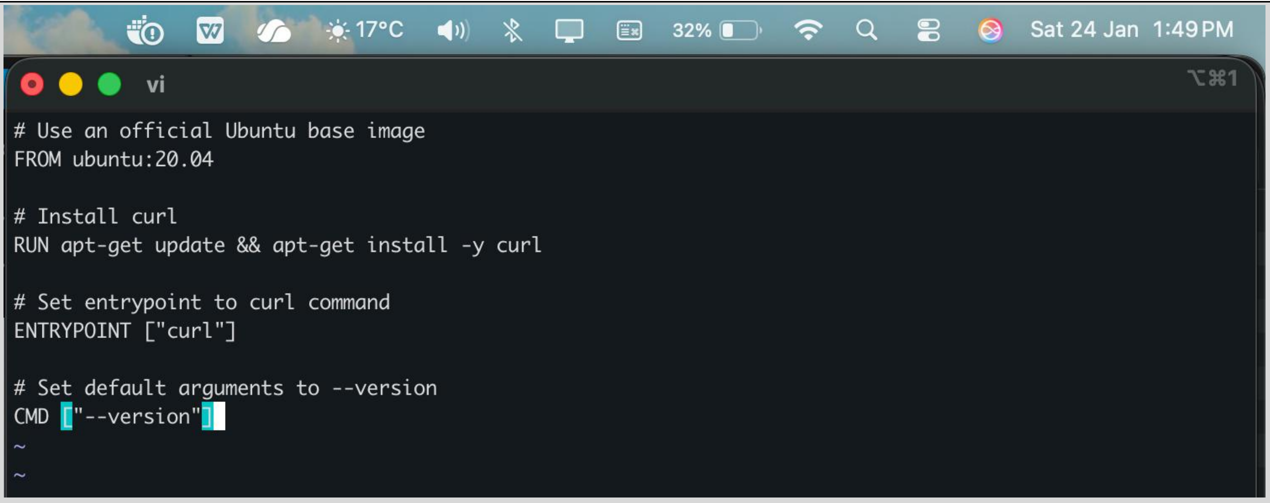
This runs the container with /bin/bash as the entrypoint, overriding the default ENTRYPOINT.

Part 5: Combining CMD and ENTRYPOINT

1. **Create a Dockerfile with Both CMD and ENTRYPOINT:**

Modify the Dockerfile to use both CMD and ENTRYPOINT:

```
# Use an official Ubuntu base image
FROM ubuntu:20.04

# Install curl
RUN apt-get update && apt-get install -y curl

# Set entrypoint to curl
ENTRYPOINT ["curl"]

# Set default arguments to --version
CMD ["--version"]
```

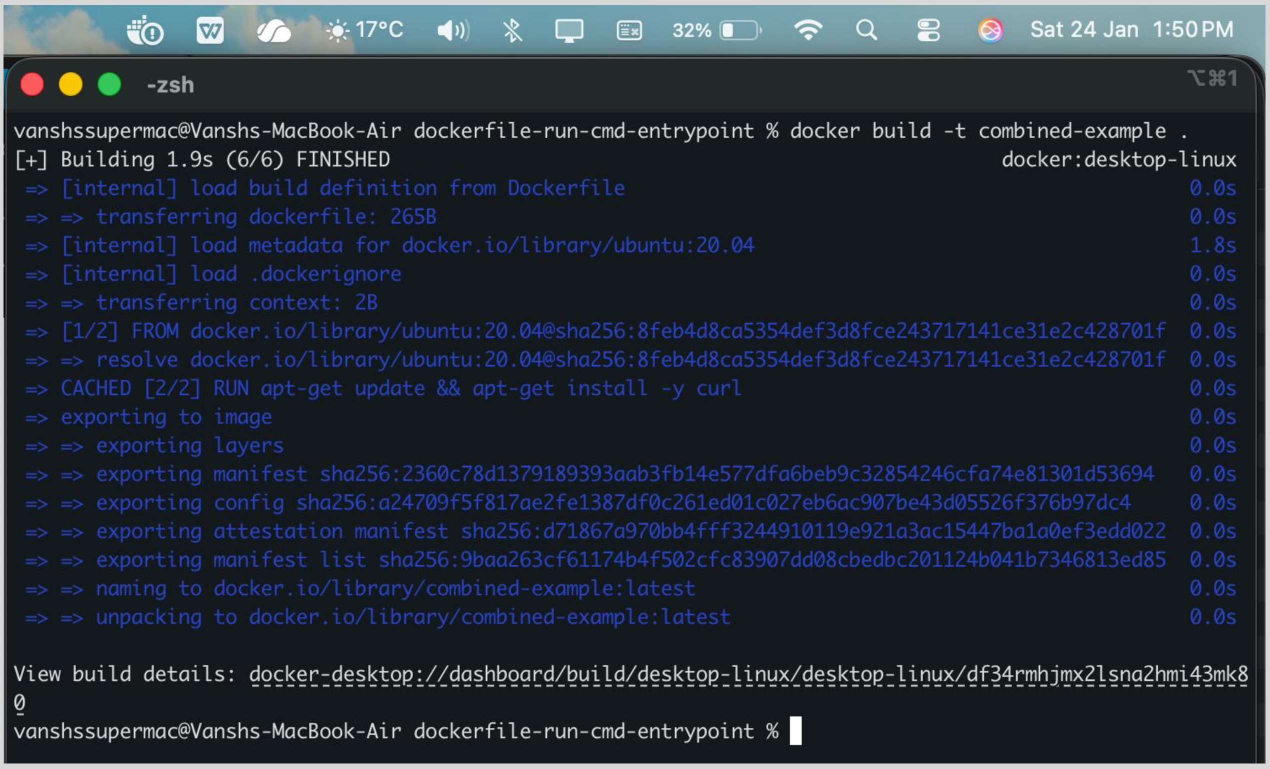


```
# Use an official Ubuntu base image
FROM ubuntu:20.04

# Install curl
RUN apt-get update && apt-get install -y curl

# Set entrypoint to curl command
ENTRYPOINT ["curl"]

# Set default arguments to --version
CMD ["--version"]
~
~
```

2. **Build the Image:**

Build the new image:

docker build -t combined-example .

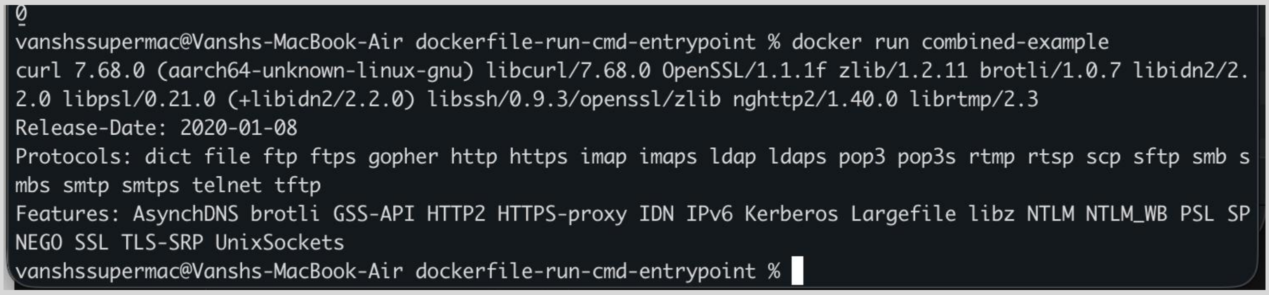


```
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker build -t combined-example .
[+] Building 1.9s (6/6) FINISHED                                                docker:desktop-linux
 => [internal] load build definition from Dockerfile                                            0.0s
 => => transferring dockerfile: 265B                                                            0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                                 1.8s
 => [internal] load .dockerignore                                                               0.0s
 => => transferring context: 2B                                                                 0.0s
 => [1/2] FROM docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f  0.0s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f  0.0s
 => CACHED [2/2] RUN apt-get update && apt-get install -y curl                                  0.0s
 => exporting to image                                                                          0.0s
 => => exporting layers                                                                         0.0s
 => => exporting manifest sha256:2360c78d1379189393aab3fb14e577dfa6beb9c32854246cfa74e81301d53694  0.0s
 => => exporting config sha256:a24709f5f817ae2fe1387df0c261ed01c027eb6ac907be43d05526f376b97dc4    0.0s
 => => exporting attestation manifest sha256:d71867a970bb4fff3244910119e921a3ac15447ba1a0ef3edd022  0.0s
 => => exporting manifest list sha256:9baa263cf61174b4f502cfc83907dd08cbedbc201124b041b7346813ed85  0.0s
 => => naming to docker.io/library/combined-example:latest                                      0.0s
 => => unpacking to docker.io/library/combined-example:latest                                   0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/df34rmhjmx2lsna2hmi43mk8
0
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint %
```

3. **Run the Container:**

When you run the container without specifying any arguments, it will use the CMD as arguments to ENTRYPOINT:

```
docker run combined-example
```



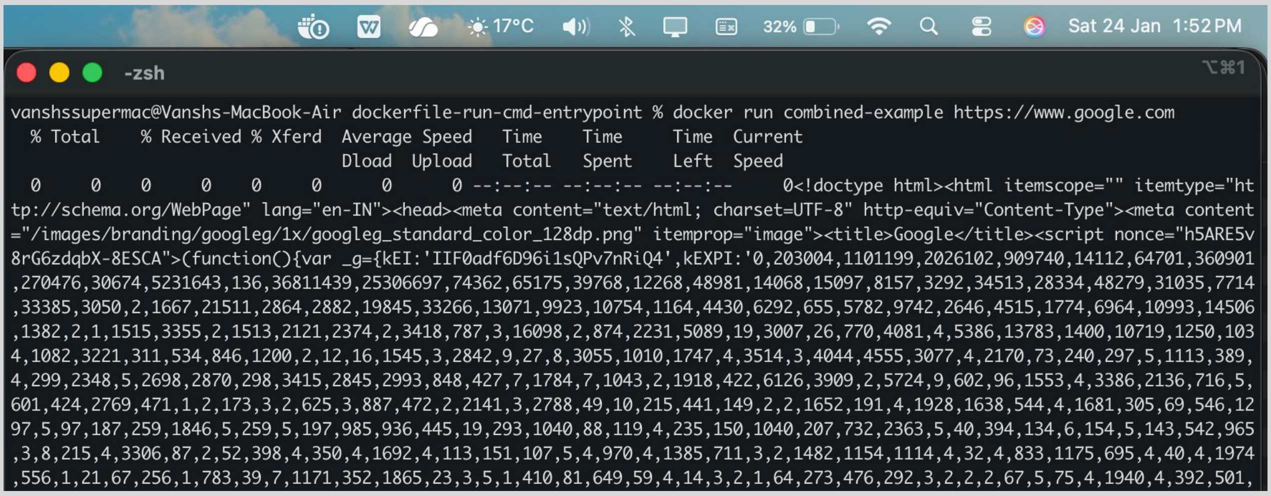
```
0
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker run combined-example
curl 7.68.0 (aarch64-unknown-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 brotli/1.0.7 libidn2/2.
2.0 libpsl/0.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2.3
Release-Date: 2020-01-08
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtmp rtsp scp sftp smb s
mbs smtp smtps telnet tftp
Features: AsynchDNS brotli GSS-API HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM NTLM_WB PSL SP
NEGO SSL TLS-SRP UnixSockets
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % 
```

The output will show the curl version, as ENTRYPOINT is curl and CMD provides --version as the argument.

4. **Override CMD Arguments:**

You can override the CMD arguments by specifying your own arguments:

```
docker run combined-example https://www.google.com
```



```
vanshssupermac@Vanshs-MacBook-Air dockerfile-run-cmd-entrypoint % docker run combined-example https://www.google.com
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0<!doctype html><html itemscope="" itemtype="ht
tp://schema.org/WebPage" lang="en-IN"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content
="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="h5ARE5v
8rG6zdqbX-8ESCA">(function(){var _g={kEI:'IIF0adf6D96i1sQPv7nRiQ4',kEXPI:'0,203004,1101199,2026102,909740,14112,64701,360901
,270476,30674,5231643,136,36811439,25306697,74362,65175,39768,12268,48981,14068,15097,8157,3292,34513,28334,48279,31035,7714
,33385,3050,2,1667,21511,2864,2882,19845,33266,13071,9923,10754,1164,4430,6292,655,5782,9742,2646,4515,1774,6964,10993,14506
,1382,2,1,1515,3355,2,1513,2121,2374,2,3418,787,3,16098,2,874,2231,5089,19,3007,26,770,4081,4,5386,13783,1400,10719,1250,103
4,1082,3221,311,534,846,1200,2,12,16,1545,3,2842,9,27,8,3055,1010,1747,4,3514,3,4044,4555,3077,4,2170,73,240,297,5,1113,389,
4,299,2348,5,2698,2870,298,3415,2845,2993,848,427,7,1784,7,1043,2,1918,422,6126,3909,2,5724,9,602,96,1553,4,3386,2136,716,5,
601,424,2769,471,1,2,173,3,2,625,3,887,472,2,2141,3,2788,49,10,215,441,149,2,2,1652,191,4,1928,1638,544,4,1681,305,69,546,12
97,5,97,187,259,1846,5,259,5,197,985,936,445,19,293,1040,88,119,4,235,150,1040,207,732,2363,5,40,394,134,6,154,5,143,542,965
,3,8,215,4,3306,87,2,52,398,4,350,4,1692,4,113,151,107,5,4,970,4,1385,711,3,2,1482,1154,1114,4,32,4,833,1175,695,4,40,4,1974
,556,1,21,67,256,1,783,39,7,1171,352,1865,23,3,5,1,410,81,649,59,4,14,3,2,1,64,273,476,292,3,2,2,2,67,5,75,4,1940,4,392,501,
```

This command will run curl https://www.google.com inside the container.

Summary of Differences:

- **RUN:** Executes commands during the image build process and creates layers. It is used to install packages and configure the environment.
- **CMD:** Specifies the default command to run when the container starts. It can be overridden by passing a different command when running the container.
- **ENTRYPOINT:** Specifies the main command for the container. It is harder to override but allows passing arguments from the command line. When combined with CMD, CMD provides the default arguments for ENTRYPOINT.

Conclusion:

This lab exercise demonstrates the fundamental differences between RUN, CMD, and ENTRYPOINT in Docker. Each command serves a different purpose, from image build-time configuration (RUN) to defining the container's behavior at runtime (CMD and ENTRYPOINT). Understanding these differences is crucial for building effective and flexible Docker images.

Thank You