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

Name Viraj Bhidola

Sap id 500121825

B2 Devops

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.

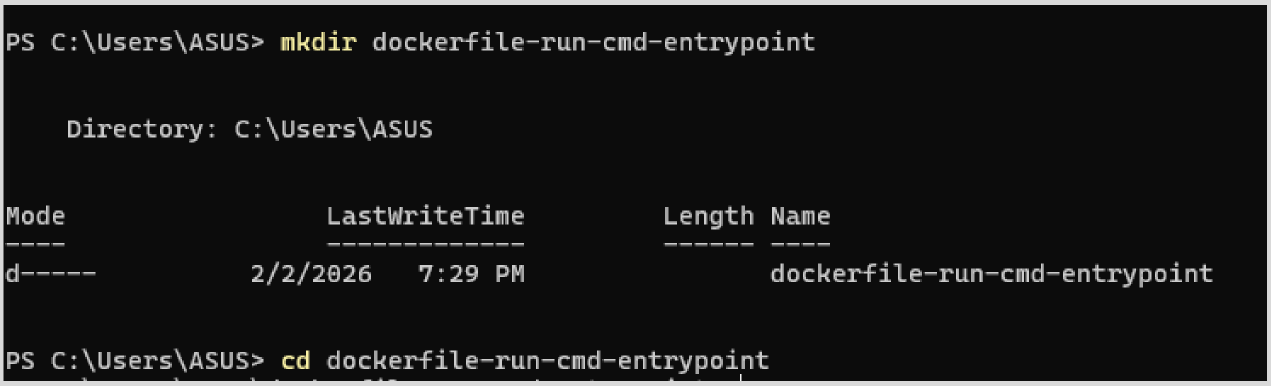
- **ENTRYPOINT:** Defines the main executable for the container, which can't be easily overridden. However, additional arguments can be passed when the container starts.

Part 2: Exploring RUN Command

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

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

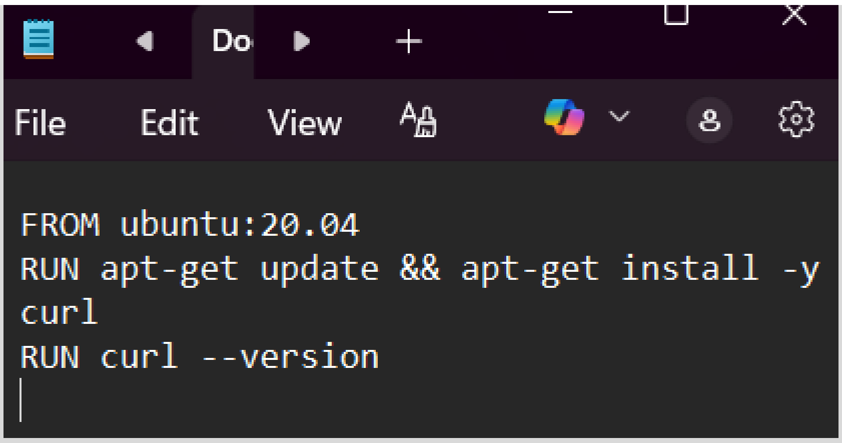
```
mkdir dockerfile-run-cmd-entrypoint && cd dockerfile-run-cmd-entrypoint
```



```
PS C:\Users\ASUS> mkdir dockerfile-run-cmd-entrypoint

    Directory: C:\Users\ASUS

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          2/2/2026   7:29 PM                dockerfile-run-cmd-entrypoint

PS C:\Users\ASUS> cd dockerfile-run-cmd-entrypoint
```

Create a simple Dockerfile that uses the RUN instruction:

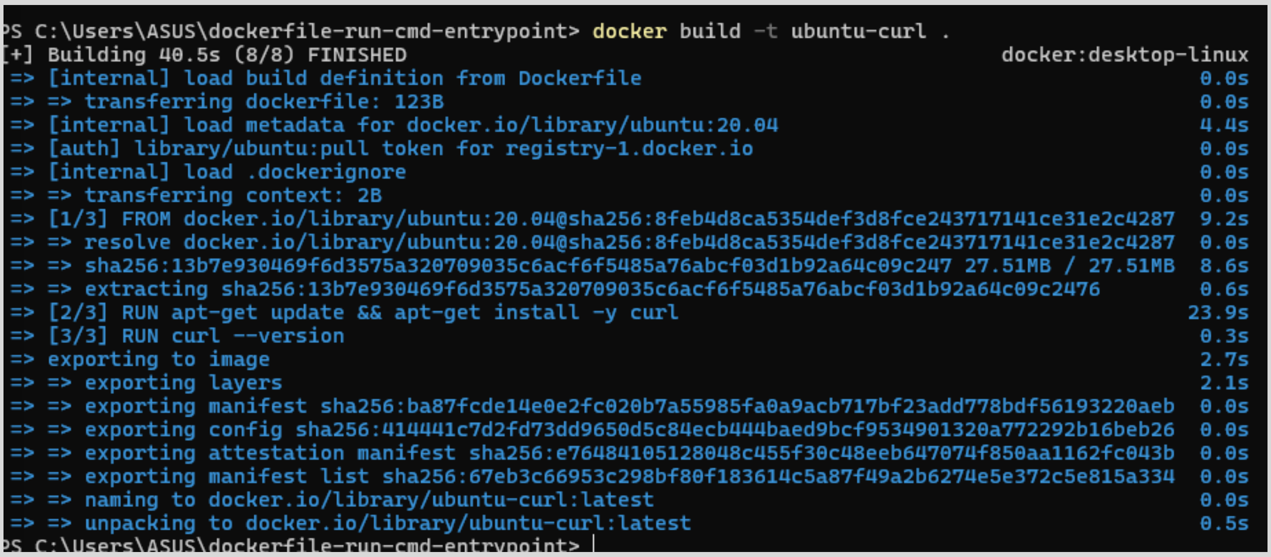
```
# Use an official Ubuntu base image

FROM ubuntu:20.04
```



```
FROM ubuntu:20.04
RUN apt-get update && apt-get install -y
curl
RUN curl --version
```

# Update the package repository and install curl

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



```
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker build -t ubuntu-curl .
[+] Building 40.5s (8/8) FINISHED                                                        docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                    0.0s
 => => transferring dockerfile: 123B                                                                    0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                                        4.4s
 => [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@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c4287  9.2s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c4287  0.0s
 => => sha256:13b7e930469f6d3575a320709035c6acf6f5485a76abcf03d1b92a64c09c247 27.51MB / 27.51MB  8.6s
 => => extracting sha256:13b7e930469f6d3575a320709035c6acf6f5485a76abcf03d1b92a64c09c2476        0.6s
 => [2/3] RUN apt-get update && apt-get install -y curl                                               23.9s
 => [3/3] RUN curl --version                                                                            0.3s
 => exporting to image                                                                                  2.7s
 => => exporting layers                                                                                 2.1s
 => => exporting manifest sha256:ba87fcde14e0e2fc020b7a55985fa0a9acb717bf23add778bdf56193220aeb  0.0s
 => => exporting config sha256:414441c7d2fd73dd9650d5c84ecb444baed9bcf9534901320a772292b16beb26  0.0s
 => => exporting attestation manifest sha256:e76484105128048c455f30c48eeb647074f850aa1162fc043b  0.0s
 => => exporting manifest list sha256:67eb3c66953c298bf80f183614c5a87f49a2b6274e5e372c5e815a334  0.0s
 => => naming to docker.io/library/ubuntu-curl:latest                                                   0.0s
 => => unpacking to docker.io/library/ubuntu-curl:latest                                                0.5s
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint>
```

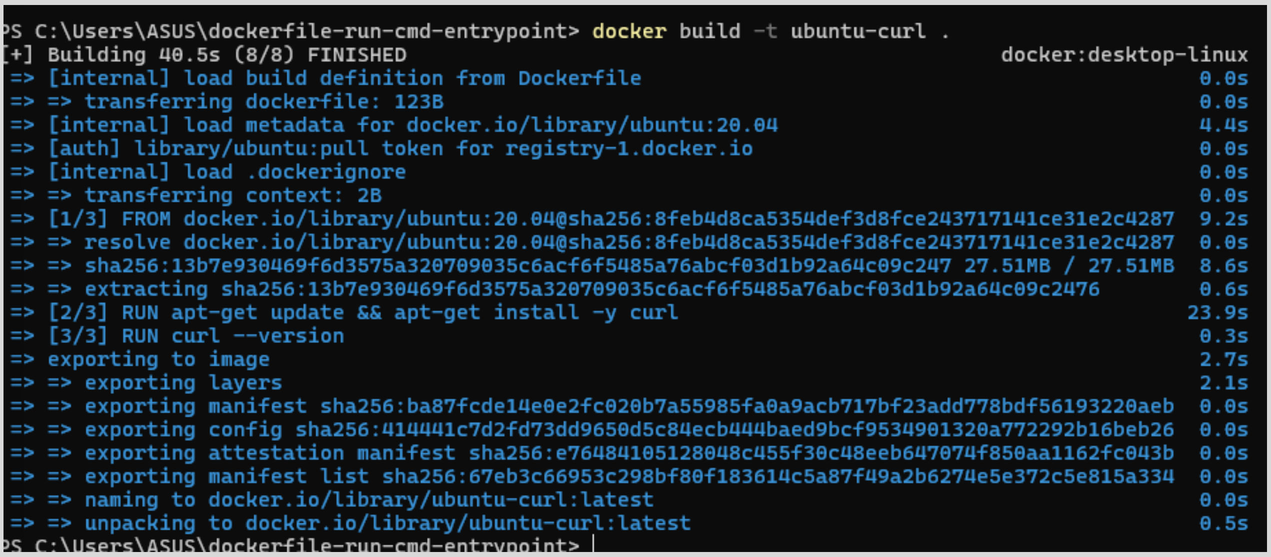
# Print the version of curl

RUN curl --version

2. **Build the Docker Image:**

Build the image using the Dockerfile:

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



```
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker build -t ubuntu-curl .
[+] Building 40.5s (8/8) FINISHED                                                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                       0.0s
 => => transferring dockerfile: 123B                                                                       0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                                            4.4s
 => [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@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c4287  9.2s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c4287  0.0s
 => => sha256:13b7e930469f6d3575a320709035c6acf6f5485a76abcf03d1b92a64c09c247 27.51MB / 27.51MB  8.6s
 => => extracting sha256:13b7e930469f6d3575a320709035c6acf6f5485a76abcf03d1b92a64c09c2476        0.6s
 => [2/3] RUN apt-get update && apt-get install -y curl                                                   23.9s
 => [3/3] RUN curl --version                                                                               0.3s
 => exporting to image                                                                                     2.7s
 => => exporting layers                                                                                    2.1s
 => => exporting manifest sha256:ba87fcde14e0e2fc020b7a55985fa0a9acb717bf23add778bdf56193220aeb  0.0s
 => => exporting config sha256:414441c7d2fd73dd9650d5c84ecb444baed9bcf9534901320a772292b16beb26  0.0s
 => => exporting attestation manifest sha256:e76484105128048c455f30c48eeb647074f850aa1162fc043b  0.0s
 => => exporting manifest list sha256:67eb3c66953c298bf80f183614c5a87f49a2b6274e5e372c5e815a334  0.0s
 => => naming to docker.io/library/ubuntu-curl:latest                                                      0.0s
 => => unpacking to docker.io/library/ubuntu-curl:latest                                                   0.5s
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint>
```

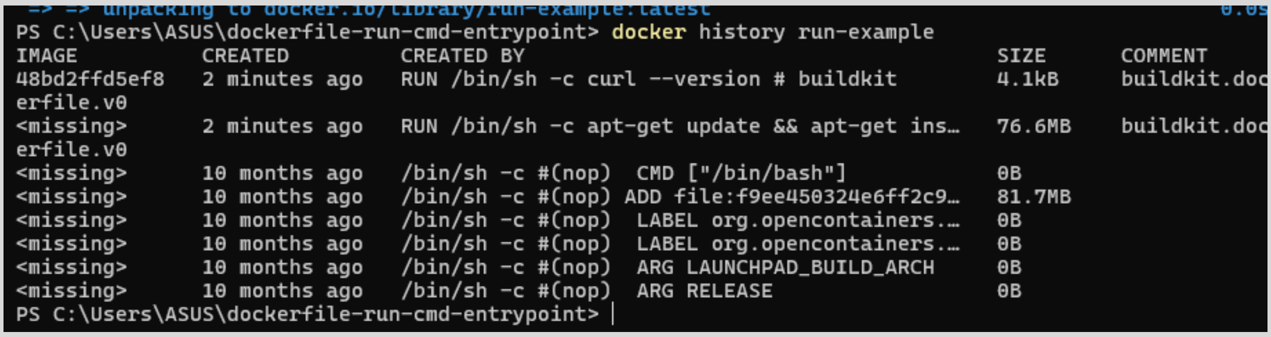
3. **Explanation:**

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
```



```
 => => unpacking to docker.io/library/run-example:latest                                  0.0s
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker history run-example
IMAGE          CREATED          CREATED BY                                      SIZE      COMMENT
48bd2ffd5ef8   2 minutes ago    RUN /bin/sh -c curl --version # buildkit        4.1kB     buildkit.doc
erfile.v0
<missing>      2 minutes ago    RUN /bin/sh -c apt-get update && apt-get ins…   76.6MB    buildkit.doc
erfile.v0
<missing>      10 months ago    /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>      10 months ago    /bin/sh -c #(nop) ADD file:f9ee450324e6ff2c9…   81.7MB
<missing>      10 months ago    /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      10 months ago    /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      10 months ago    /bin/sh -c #(nop)  ARG LAUNCHPAD_BUILD_ARCH     0B
<missing>      10 months ago    /bin/sh -c #(nop)  ARG RELEASE                  0B
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> |
```

Each RUN command creates a new layer in the image.

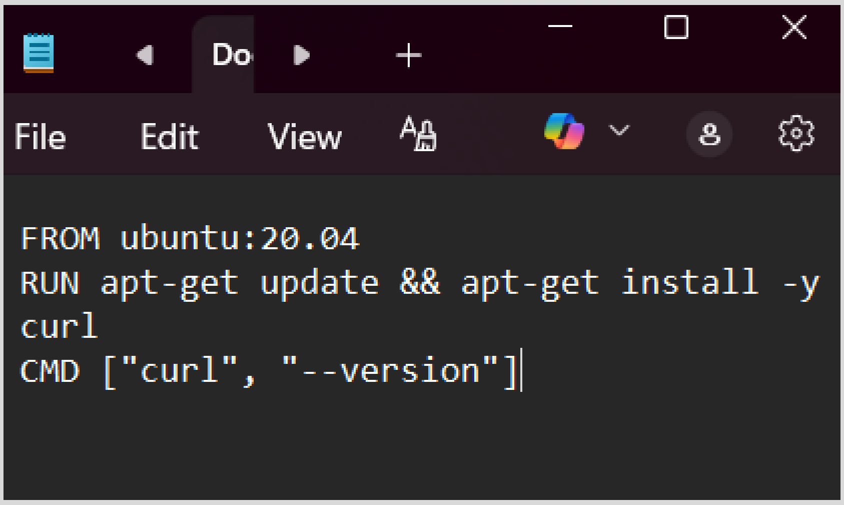
Part 3: Exploring CMD Command

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"]
```

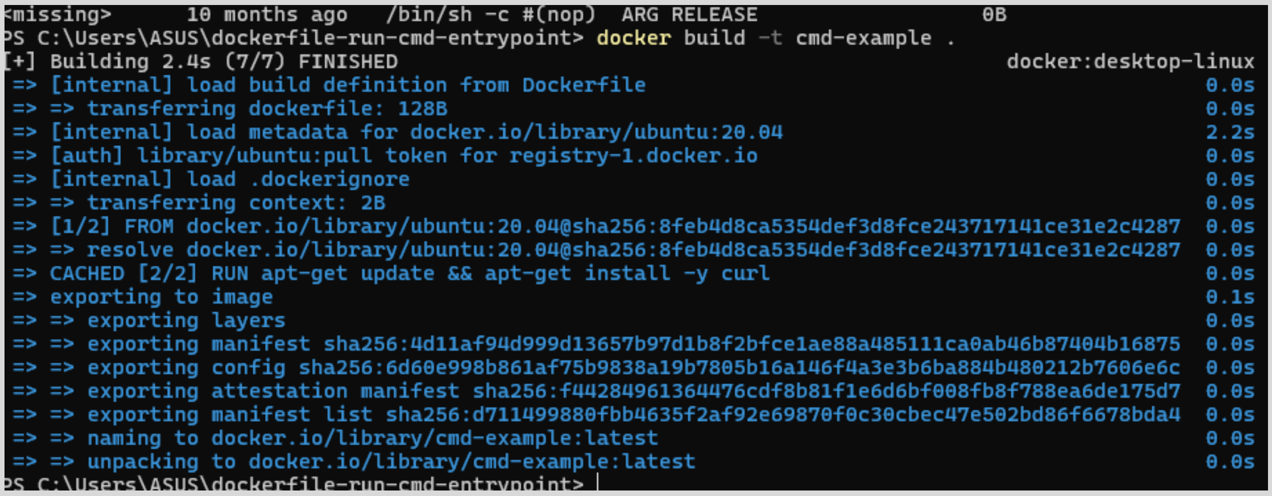


```
FROM ubuntu:20.04
RUN apt-get update && apt-get install -y
curl
CMD ["curl", "--version"]
```

2. **Build the Docker Image:**

Build the Docker image again:

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

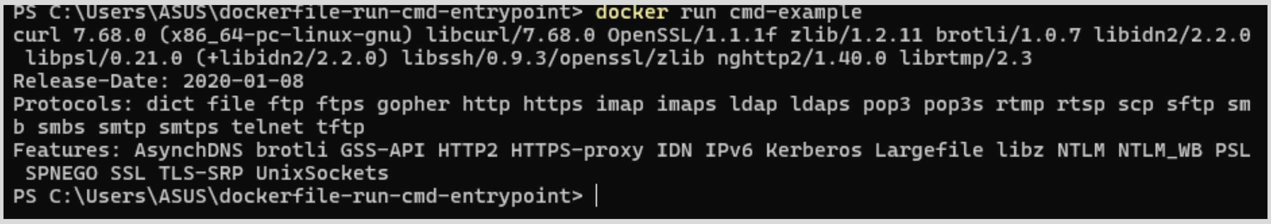


```
<missing>      10 months ago   /bin/sh -c #(nop)  ARG RELEASE                 0B
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker build -t cmd-example .
[+] Building 2.4s (7/7) FINISHED                                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                                       0.0s
 => => transferring dockerfile: 128B                                                       0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                            2.2s
 => [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@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c4287  0.0s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c4287  0.0s
 => CACHED [2/2] RUN apt-get update && apt-get install -y curl                             0.0s
 => exporting to image                                                                     0.1s
 => => exporting layers                                                                    0.0s
 => => exporting manifest sha256:4d11af94d999d13657b97d1b8f2bfce1ae88a485111ca0ab46b87404b16875  0.0s
 => => exporting config sha256:6d60e998b861af75b9838a19b7805b16a146f4a3e3b6ba884b480212b7606e6c  0.0s
 => => exporting attestation manifest sha256:f44284961364476cdf8b81f1e6d6bf008fb8f788ea6de175d7  0.0s
 => => exporting manifest list sha256:d711499880fbb4635f2af92e69870f0c30cbec47e502bd86f6678bda4  0.0s
 => => naming to docker.io/library/cmd-example:latest                                      0.0s
 => => unpacking to docker.io/library/cmd-example:latest                                   0.0s
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint>
```

3. **Run the Container:**

Run the container and see the output:

```
docker run cmd-example
```



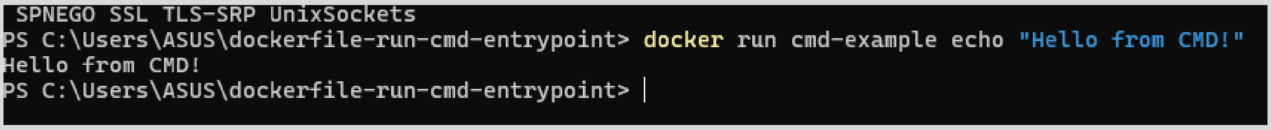
```
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker run cmd-example
curl 7.68.0 (x86_64-pc-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 brotli/1.0.7 libidn2/2.2.0
 libpsl/0.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2.3
Release-Date: 2020-01-08
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtmp rtsp scp sftp sm
b 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
PS C:\Users\ASUS\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!"
```



```
 SPNEGO SSL TLS-SRP UnixSockets
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker run cmd-example echo "Hello from CMD!"
Hello from CMD!
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> |
```

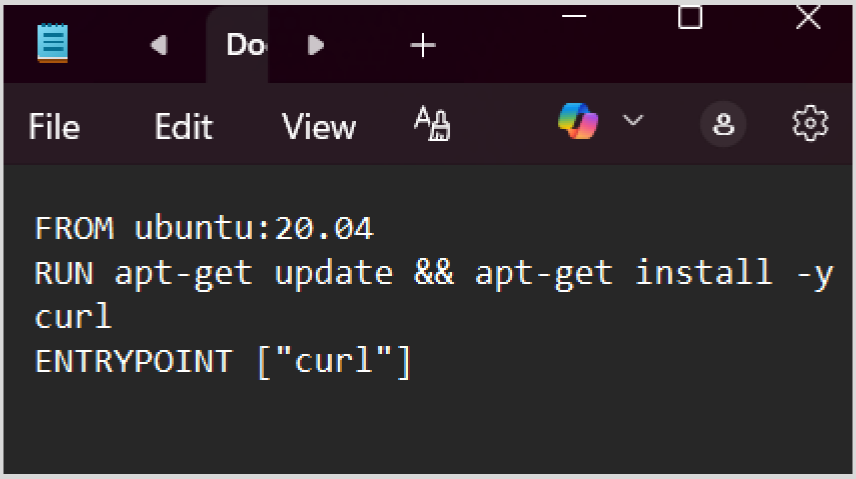
This will print Hello from CMD!, showing that the CMD can be overridden at runtime.

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"]
```

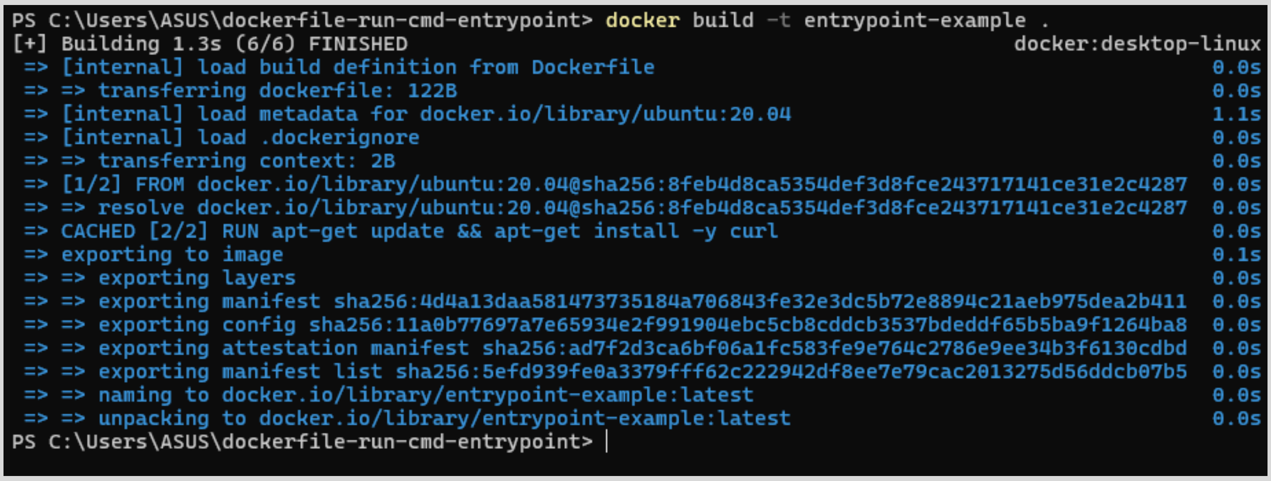


```
FROM ubuntu:20.04
RUN apt-get update && apt-get install -y
curl
ENTRYPOINT ["curl"]
```

2. **Build the Docker Image:**

Build the image with the ENTRYPOINT instruction:

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

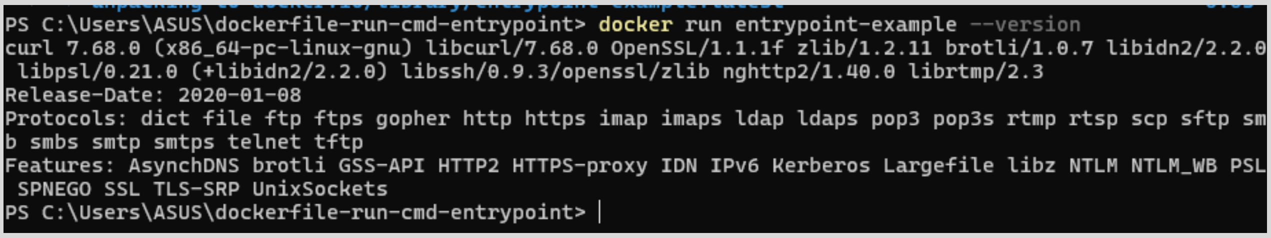


```
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker build -t entrypoint-example .
[+] Building 1.3s (6/6) FINISHED                                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile                                              0.0s
 => => transferring dockerfile: 122B                                                              0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                                   1.1s
 => [internal] load .dockerignore                                                                 0.0s
 => => transferring context: 2B                                                                   0.0s
 => [1/2] FROM docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c4287  0.0s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c4287  0.0s
 => CACHED [2/2] RUN apt-get update && apt-get install -y curl                                    0.0s
 => exporting to image                                                                            0.1s
 => => exporting layers                                                                           0.0s
 => => exporting manifest sha256:4d4a13daa581473735184a706843fe32e3dc5b72e8894c21aeb975dea2b411  0.0s
 => => exporting config sha256:11a0b77697a7e65934e2f991904ebc5cb8cddcb3537bdeddf65b5ba9f1264ba8  0.0s
 => => exporting attestation manifest sha256:ad7f2d3ca6bf06a1fc583fe9e764c2786e9ee34b3f6130cdbd  0.0s
 => => exporting manifest list sha256:5efd939fe0a3379fff62c222942df8ee7e79cac2013275d56ddcb07b5  0.0s
 => => naming to docker.io/library/entrypoint-example:latest                                      0.0s
 => => unpacking to docker.io/library/entrypoint-example:latest                                   0.0s
PS C:\Users\ASUS\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
```



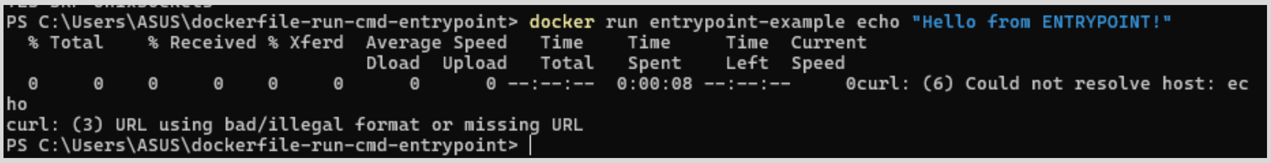
```
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker run entrypoint-example --version
curl 7.68.0 (x86_64-pc-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 brotli/1.0.7 libidn2/2.2.0
 libpsl/0.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2.3
Release-Date: 2020-01-08
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop3s rtmp rtsp scp sftp sm
b 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
PS C:\Users\ASUS\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!"
```

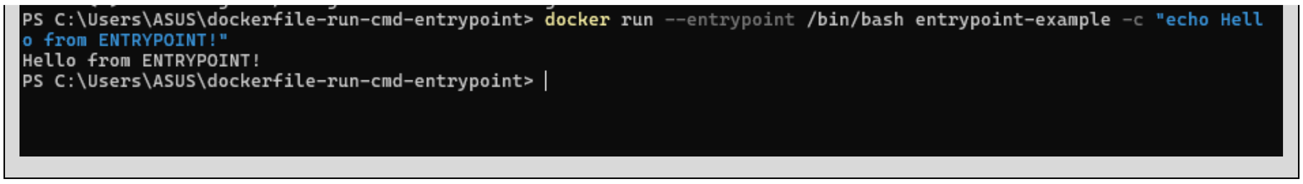


```
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker run entrypoint-example echo "Hello from ENTRYPOINT!"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:08 --:--:--     0curl: (6) Could not resolve host: ec
ho
curl: (3) URL using bad/illegal format or missing URL
PS C:\Users\ASUS\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!"
```



```
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker run --entrypoint /bin/bash entrypoint-example -c "echo Hell
o from ENTRYPOINT!"
Hello from ENTRYPOINT!
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> |
```

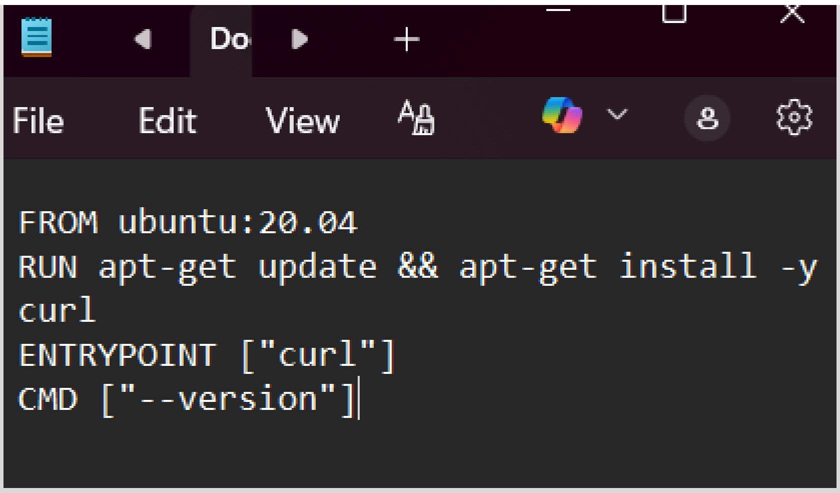
This runs the container with /bin/bash as the entrypoint, overriding the default ENTRYPOINT.

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"]
```

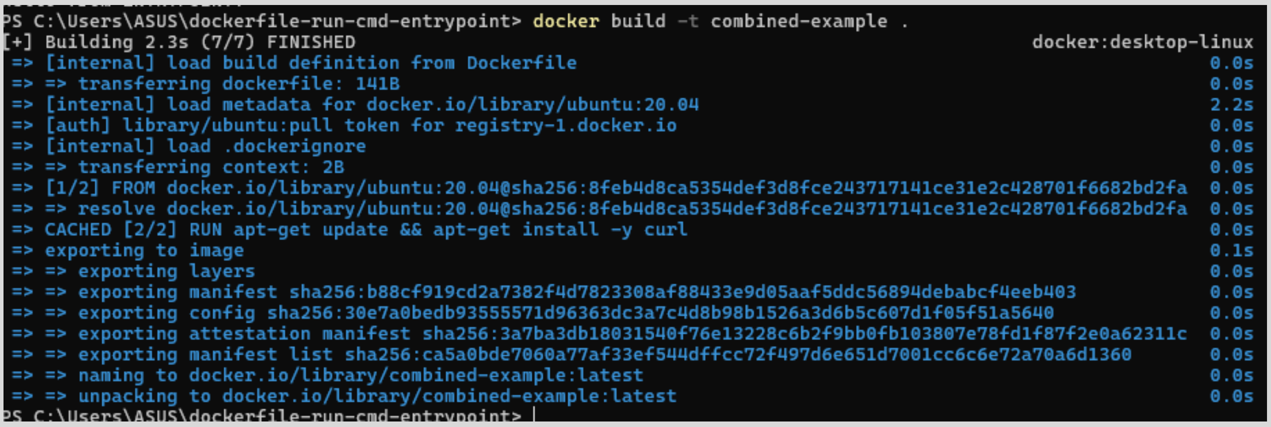


```
FROM ubuntu:20.04
RUN apt-get update && apt-get install -y
curl
ENTRYPOINT ["curl"]
CMD ["--version"]
```

2. **Build the Image:**

Build the new image:

```
docker build -t combined-example .
```

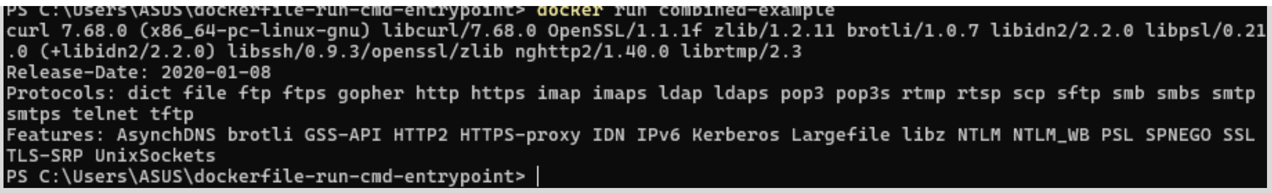


```
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker build -t combined-example .
[+] Building 2.3s (7/7) FINISHED                                                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                              0.0s
 => => transferring dockerfile: 141B                                                                              0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                                                   2.2s
 => [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@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f6682bd2fa  0.0s
 => => resolve docker.io/library/ubuntu:20.04@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f6682bd2fa  0.0s
 => CACHED [2/2] RUN apt-get update && apt-get install -y curl                                                    0.0s
 => exporting to image                                                                                            0.1s
 => => exporting layers                                                                                           0.0s
 => => exporting manifest sha256:b88cf919cd2a7382f4d7823308af88433e9d05aaf5ddc56894debabcf4eeb403                 0.0s
 => => exporting config sha256:30e7a0bedb93555571d96363dc3a7c4d8b98b1526a3d6b5c607d1f05f51a5640                   0.0s
 => => exporting attestation manifest sha256:3a7ba3db18031540f76e13228c6b2f9bb0fb103807e78fd1f87f2e0a62311c  0.0s
 => => exporting manifest list sha256:ca5a0bde7060a77af33ef544dffcc72f497d6e651d7001cc6c6e72a70a6d1360        0.0s
 => => naming to docker.io/library/combined-example:latest                                                        0.0s
 => => unpacking to docker.io/library/combined-example:latest                                                     0.0s
PS C:\Users\ASUS\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
```



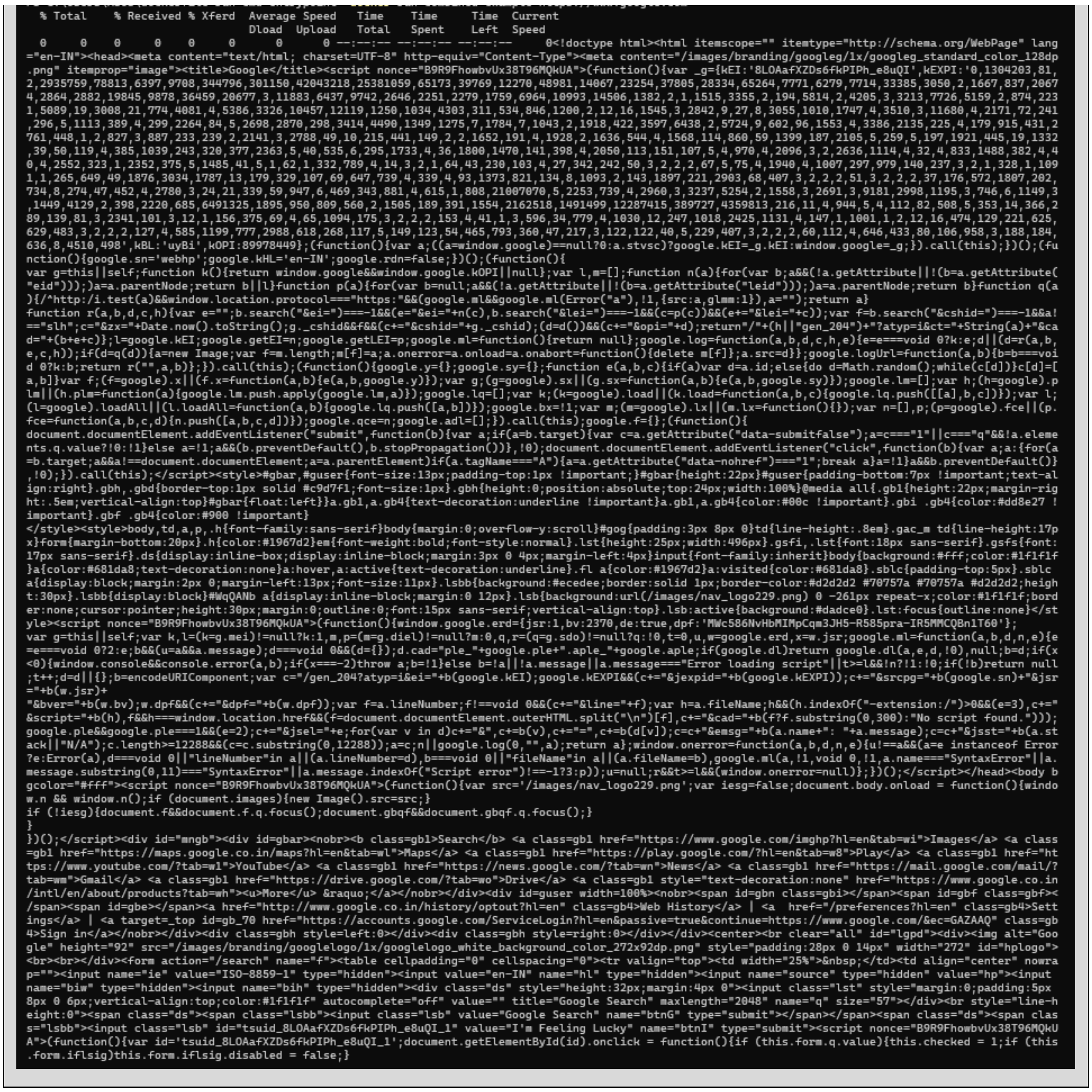
```
PS C:\Users\ASUS\dockerfile-run-cmd-entrypoint> docker run combined-example
curl 7.68.0 (x86_64-pc-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 brotli/1.0.7 libidn2/2.2.0 libpsl/0.21
.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2.3
Release-Date: 2020-01-08
Protocols: 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
PS C:\Users\ASUS\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
```



```
  % 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="http://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="B9R9FhowbvUx38T96MQkUA">(function(){var _g={kEI:'8LOAafXZDs6fkPIPh_e8uQI',kEXPI:'0,1304203,81,
2,2935759,78813,6397,9708,344796,301150,42043218,25381059,65173,39769,12270,48981,14067,23254,37805,28334,65264,7771,6279,7714,33385,3050,2,1667,837,2067
4,2864,2882,19845,9878,36459,20677,3,11883,6437,9742,2646,2251,2279,1759,6964,10993,14506,1382,2,1,1515,3355,2,194,5814,2,4205,3,3213,7726,5159,2,874,223
1,5089,19,3008,21,774,4081,4,5386,3326,10457,12119,1250,1034,4303,311,534,846,1200,2,12,16,1545,3,2842,9,27,8,3055,1010,1747,4,3510,3,11680,4,2171,72,241
,296,5,1113,389,4,299,2264,84,5,2698,2870,298,3414,4490,1349,1275,7,1784,7,1043,2,1918,422,3597,6438,2,5724,9,602,96,1553,4,3386,2135,225,4,179,915,431,2
761,448,1,2,827,3,887,233,239,2,2141,3,2788,49,10,215,441,149,2,2,1652,191,4,1928,2,1636,544,4,1568,114,860,59,1399,187,2105,5,259,5,197,1921,445,19,1332
,39,50,119,4,385,1039,243,320,377,2363,5,40,535,6,295,1733,4,36,1800,1470,141,398,4,2050,113,151,107,5,4,970,4,2096,3,2,2636,1114,4,32,4,833,1488,382,4,4
0,4,2552,323,1,2352,375,5,1485,41,5,1,62,1,332,789,4,14,3,2,1,64,43,230,103,4,27,342,242,50,3,2,2,2,67,5,75,4,1940,4,1007,297,979,140,237,3,2,1,328,1,109
1,1,265,649,49,1876,3034,1787,13,179,329,107,69,647,739,4,339,4,93,1373,821,134,8,1093,2,143,1897,221,2903,68,407,3,2,2,2,51,3,2,2,2,37,176,572,1807,202,
734,8,274,47,452,4,2780,3,24,21,339,59,947,6,469,343,881,4,615,1,808,21007070,5,2253,739,4,2960,3,3237,5254,2,1558,3,2691,3,9181,2998,1195,3,746,6,1149,3
,1449,4129,2,398,2220,685,6491325,1895,950,809,560,2,1505,189,391,1554,2162518,1491499,12287415,389727,4359813,216,11,4,944,5,4,112,82,508,5,353,14,366,2
89,139,81,3,2341,101,3,12,1,156,375,69,4,65,1094,175,3,2,2,2,153,4,41,1,3,596,34,779,4,1030,12,247,1018,2425,1131,4,147,1,1001,1,2,12,16,474,129,221,625,
629,483,3,2,2,2,127,4,585,1199,777,2988,618,268,117,5,149,123,54,465,793,360,47,217,3,122,122,40,5,229,407,3,2,2,2,60,112,4,646,433,80,106,958,3,188,184,
636,8,4510,498',kBL:'uyBi',kOPI:89978449};(function(){var a;((a=window.google)==null?0:a.stvsc)?google.kEI=_g.kEI:window.google=_g;}).call(this);})();(fu
nction(){google.sn='webhp';google.kHL='en-IN';google.rdn=false;})();(function(){
var g=this||self;function k(){return window.google&&window.google.kOPI||null};var l,m=[];function n(a){for(var b;a&&(!a.getAttribute||!(b=a.getAttribute(
"eid")));)a=a.parentNode;return b||l}function p(a){for(var b=null;a&&(!a.getAttribute||!(b=a.getAttribute("leid")));)a=a.parentNode;return b}function q(a
){/^http:/i.test(a)&&window.location.protocol==="https:"&&(google.ml&&google.ml(Error("a"),!1,{src:a,glmm:1}),a="");return a}
function r(a,b,d,c,h){var e="";b.search("&ei=")===-1&&(e="&ei="+n(c),b.search("&lei=")===-1&&(c=p(c))&&(e+="&lei="+c));var f=b.search("&cshid=")===-1&&a!
=="slh";c="&zx="+Date.now().toString();g._cshid&&f&&(c+="&cshid="+g._cshid);(d=d())&&(c+="&opi="+d);return"/"+(h||"gen_204")+"?atyp=i&ct="+String(a)+"&ca
d="+(b+e+c)};l=google.kEI;google.getEI=n;google.getLEI=p;google.ml=function(){return null};google.log=function(a,b,d,c,h,e){e=e===void 0?k:e;d||(d=r(a,b,
e,c,h));if(d=q(d)){a=new Image;var f=m.length;m[f]=a;a.onerror=a.onload=a.onabort=function(){delete m[f]};a.src=d}};google.logUrl=function(a,b){b=b===voi
d 0?k:b;return r("",a,b)};}).call(this);(function(){google.y={};google.sy={};function e(a,b,c){if(a)var d=a.id;else{do d=Math.random();while(c[d])}c[d]=[
a,b]}var f;(f=google).x||(f.x=function(a,b){e(a,b,google.y)});var g;(g=google).sx||(g.sx=function(a,b){e(a,b,google.sy)});google.lm=[];var h;(h=google).p
lm||(h.plm=function(a){google.lm.push.apply(google.lm,a)});google.lq=[];var k;(k=google).load||(k.load=function(a,b,c){google.lq.push([[a],b,c])});var l;
(l=google).loadAll||(l.loadAll=function(a,b){google.lq.push([a,b])});google.bx=!1;var m;(m=google).lx||(m.lx=function(){});var n=[],p;(p=google).fce||(p.
fce=function(a,b,c,d){n.push([a,b,c,d])});google.qce=n;google.adl=[];}).call(this);google.f={};(function(){
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");a=c==="1"||c==="q"&&!a.eleme
nts.q.value?!0:!1}else a=!1;a&&(b.preventDefault(),b.stopPropagation())},!0);document.documentElement.addEventListener("click",function(b){var a;a:{for(a
=b.target;a&&a!==document.documentElement;a=a.parentElement)if(a.tagName==="A"){a=a.getAttribute("data-nohref")==="1";break a}a=!1}a&&b.preventDefault()}
,!0);}).call(this);</script><style>#gbar,#guser{font-size:13px;padding-top:1px !important;}#gbar{height:22px}#guser{padding-bottom:7px !important;text-al
ign:right}.gbh,.gbd{border-top:1px solid #c9d7f1;font-size:1px}.gbh{height:0;position:absolute;top:24px;width:100%}@media all{.gb1{height:22px;margin-rig
ht:.5em;vertical-align:top}#gbar{float:left}}a.gb1,a.gb4{text-decoration:underline !important}a.gb1,a.gb4{color:#00c !important}.gbi .gb4{color:#dd8e27 !
important}.gbf .gb4{color:#900 !important}
</style><style>body,td,a,p,.h{font-family:sans-serif}body{margin:0;overflow-y:scroll}#gog{padding:3px 8px 0}td{line-height:.8em}.gac_m td{line-height:17p
x}form{margin-bottom:20px}.h{color:#1967d2}em{font-weight:bold;font-style:normal}.lst{height:25px;width:496px}.gsfi,.lst{font:18px sans-serif}.gsfs{font:
17px sans-serif}.ds{display:inline-box;display:inline-block;margin:3px 0 4px;margin-left:4px}input{font-family:inherit}body{background:#fff;color:#1f1f1f
}a{color:#681da8;text-decoration:none}a:hover,a:active{text-decoration:underline}.fl a{color:#1967d2}a:visited{color:#681da8}.sblc{padding-top:5px}.sblc
a{display:block;margin:2px 0;margin-left:13px;font-size:11px}.lsbb{background:#ecedee;border:solid 1px;border-color:#d2d2d2 #70757a #70757a #d2d2d2;heigh
t:30px}.lsbb{display:block}#WqQANb a{display:inline-block;margin:0 12px}.lsb{background:url(/images/nav_logo229.png) 0 -261px repeat-x;color:#1f1f1f;bord
er:none;cursor:pointer;height:30px;margin:0;outline:0;font:15px sans-serif;vertical-align:top}.lsb:active{background:#dadce0}.lst:focus{outline:none}</st
yle><script nonce="B9R9FhowbvUx38T96MQkUA">(function(){window.google.erd={jsr:1,bv:2370,de:true,dpf:'MWc586NvHbMIMpCqm3JH5-R585pra-IR5MMCQBn1T60'};
var g=this||self;var k,l=(k=g.mei)!=null?k:1,m,p=(m=g.diel)!=null?m:0,q,r=(q=g.sdo)!=null?q:!0,t=0,u,w=google.erd,x=w.jsr;google.ml=function(a,b,d,n,e){e
=e===void 0?2:e;b&&(u=a&&a.message);d===void 0&&(d={});d.cad="ple_"+google.ple+".aple_"+google.aple;if(google.dl)return google.dl(a,e,d,!0),null;b=d;if(x
<0){window.console&&console.error(a,b);if(x===-2)throw a;b=!1}else b=!a||!a.message||a.message==="Error loading script"||t>=l&&!n?!1:!0;if(!b)return null
;t++;d=d||{};b=encodeURIComponent;var c="/gen_204?atyp=i&ei="+b(google.kEI);google.kEXPI&&(c+="&jexpid="+b(google.kEXPI));c+="&srcpg="+b(google.sn)+"&jsr
="+b(w.jsr)+
"&bver="+b(w.bv);w.dpf&&(c+="&dpf="+b(w.dpf));var f=a.lineNumber;f!==void 0&&(c+="&line="+f);var h=a.fileName;h&&(h.indexOf("-extension:/")>0&&(e=3),c+="
&script="+b(h),f&&h===window.location.href&&(f=document.documentElement.outerHTML.split("\n")[f],c+="&cad="+b(f?f.substring(0,300):"No script found.")));
google.ple&&google.ple===1&&(e=2);c+="&jsel="+e;for(var v in d)c+="&",c+=b(v),c+="=",c+=b(d[v]);c=c+"&emsg="+b(a.name+": "+a.message);c=c+"&jsst="+b(a.st
ack||"N/A");c.length>=12288&&(c=c.substring(0,12288));a=c;n||google.log(0,"",a);return a};window.onerror=function(a,b,d,n,e){u!==a&&(a=e instanceof Error
?e:Error(a),d===void 0||"lineNumber"in a||(a.lineNumber=d),b===void 0||"fileName"in a||(a.fileName=b),google.ml(a,!1,void 0,!1,a.name==="SyntaxError"||a.
message.substring(0,11)==="SyntaxError"||a.message.indexOf("Script error")!==-1?3:p));u=null;r&&t>=l&&(window.onerror=null)};})();</script></head><body b
gcolor="#fff"><script nonce="B9R9FhowbvUx38T96MQkUA">(function(){var src='/images/nav_logo229.png';var iesg=false;document.body.onload = function(){windo
w.n && window.n();if (document.images){new Image().src=src;}
if (!iesg){document.f&&document.f.q.focus();document.gbqf&&document.gbqf.q.focus();}
}
})();</script><div id="mngb"><div id=gbar><nobr><b class=gb1>Search</b> <a class=gb1 href="https://www.google.com/imghp?hl=en&tab=wi">Images</a> <a class
=gb1 href="https://maps.google.co.in/maps?hl=en&tab=wl">Maps</a> <a class=gb1 href="https://play.google.com/?hl=en&tab=w8">Play</a> <a class=gb1 href="ht
tps://www.youtube.com/?tab=w1">YouTube</a> <a class=gb1 href="https://news.google.com/?tab=wn">News</a> <a class=gb1 href="https://mail.google.com/mail/?
tab=wm">Gmail</a> <a class=gb1 href="https://drive.google.com/?tab=wo">Drive</a> <a class=gb1 style="text-decoration:none" href="https://www.google.co.in
/intl/en/about/products?tab=wh"><u>More</u> &raquo;</a></nobr></div><div id=guser width=100%><nobr><span id=gbn class=gbi></span><span id=gbf class=gbf><
/span><span id=gbe></span><a href="http://www.google.co.in/history/optout?hl=en" class=gb4>Web History</a> | <a  href="/preferences?hl=en" class=gb4>Sett
ings</a> | <a target=_top id=gb_70 href="https://accounts.google.com/ServiceLogin?hl=en&passive=true&continue=https://www.google.com/&ec=GAZAAQ" class=gb
4>Sign in</a></nobr></div><div class=gbh style=left:0></div><div class=gbh style=right:0></div></div><center><br clear="all" id="lgpd"><div><img alt="Goo
gle" height="92" src="/images/branding/googlelogo/1x/googlelogo_white_background_color_272x92dp.png" style="padding:28px 0 14px" width="272" id="hplogo">
<br><br></div><form action="/search" name="f"><table cellpadding="0" cellspacing="0"><tr valign="top"><td width="25%">&nbsp;</td><td align="center" nowra
p=""><input name="ie" value="ISO-8859-1" type="hidden"><input value="en-IN" name="hl" type="hidden"><input name="source" type="hidden" value="hp"><input
name="biw" type="hidden"><input name="bih" type="hidden"><div class="ds" style="height:32px;margin:4px 0"><input class="lst" style="margin:0;padding:5px
8px 0 6px;vertical-align:top;color:#1f1f1f" autocomplete="off" value="" title="Google Search" maxlength="2048" name="q" size="57"></div><br style="line-h
eight:0"><span class="ds"><span class="lsbb"><input class="lsb" value="Google Search" name="btnG" type="submit"></span></span><span class="ds"><span clas
s="lsbb"><input class="lsb" id="tsuid_8LOAafXZDs6fkPIPh_e8uQI_1" value="I'm Feeling Lucky" name="btnI" type="submit"><script nonce="B9R9FhowbvUx38T96MQkU
A">(function(){var id='tsuid_8LOAafXZDs6fkPIPh_e8uQI_1';document.getElementById(id).onclick = function(){if (this.form.q.value){this.checked = 1;if (this
.form.iflsig)this.form.iflsig.disabled = false;}
```

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.