RUN vs CMD vs ENTRYPOINT in Docker

EBOOK

Understanding CMD, RUN, and ENTRYPOINT in Dockerfile

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

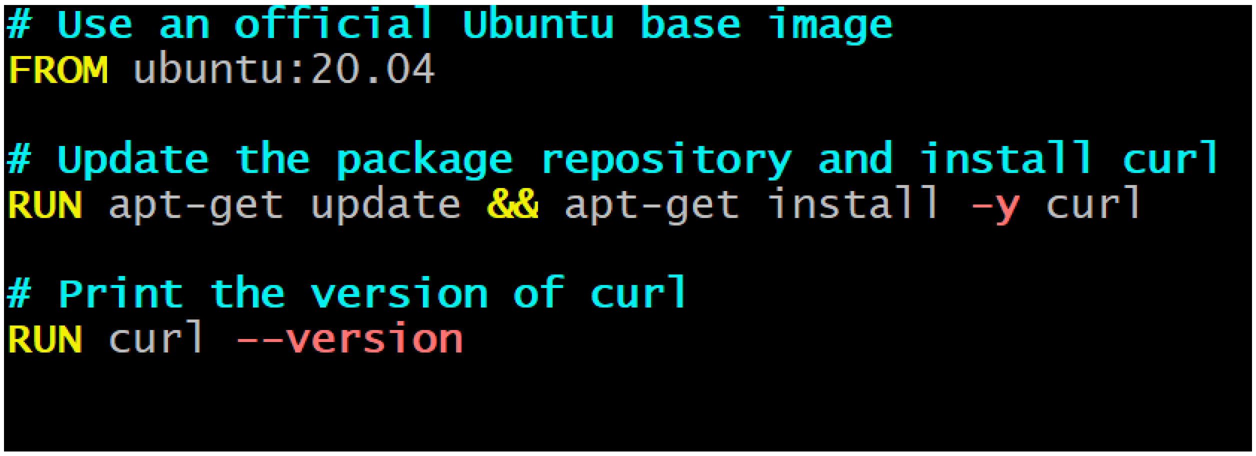
1. **Create a Dockerfile with RUN:**

Create a directory called dockerfilecommands and navigate to it:

```
mkdir dockerfilecommands && cd dockerfilecommands
```

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
```

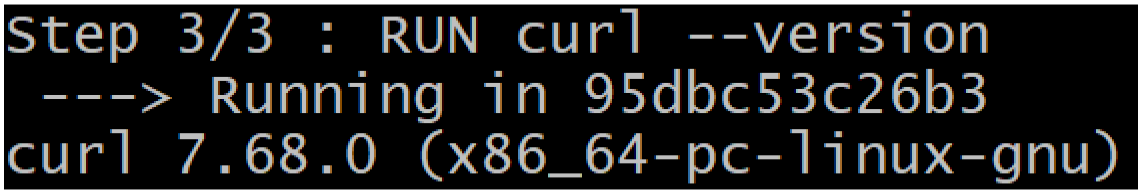


```
# Use an official Ubuntu base image
FROM ubuntu:20.04

# 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 runcommand .
```



```
Step 3/3 : RUN curl --version
 ---> Running in 95dbc53c26b3
curl 7.68.0 (x86_64-pc-linux-gnu)
```

3. **Explanation:**

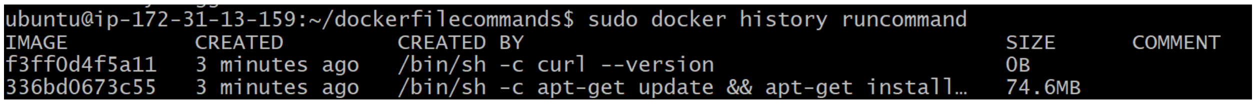
The RUN commands in this Dockerfile are executed during the image build process. The first RUN installs curl, and the second RUN command checks and prints the curl version. After the image is built, the commands executed by RUN are already baked into the image.

4. **Verify with Docker History:**

You can check the layers created by RUN using:

```
docker history runcommand
```

Each RUN command creates a new layer in the image.



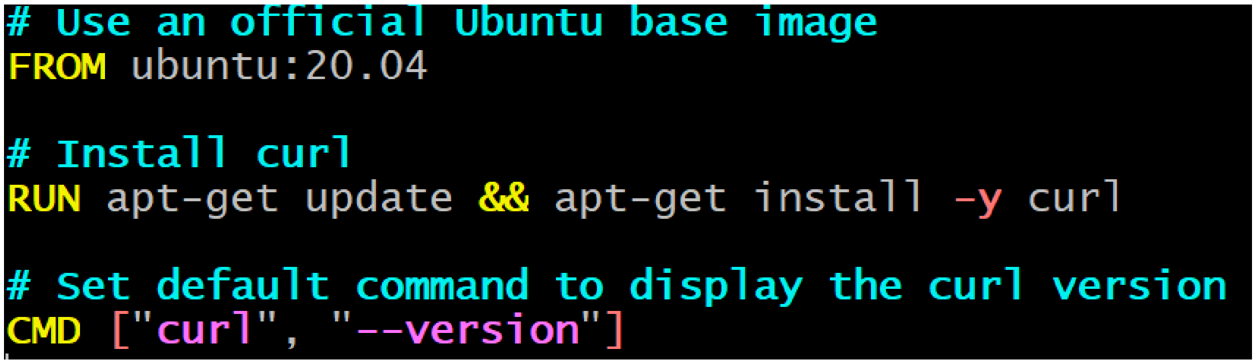
```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker history runcommand
IMAGE          CREATED         CREATED BY                                      SIZE      COMMENT
f3ff0d4f5a11   3 minutes ago   /bin/sh -c curl --version                       0B
336bd0673c55   3 minutes ago   /bin/sh -c apt-get update && apt-get install…   74.6MB
```

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"]
```

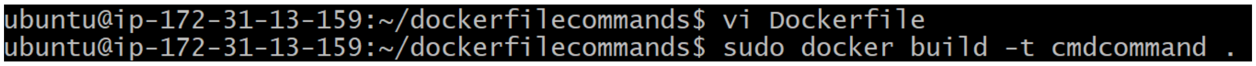


```
# Use an official Ubuntu base image
FROM ubuntu:20.04

# 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 cmdcommand .
```



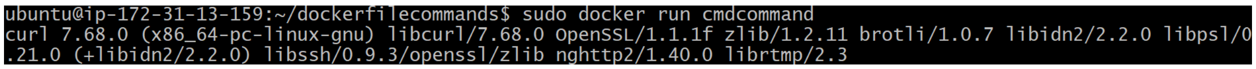
```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ vi Dockerfile
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker build -t cmdcommand .
```

3. **Run the Container:**

Run the container and see the output:

```
docker run cmdcommand
```

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



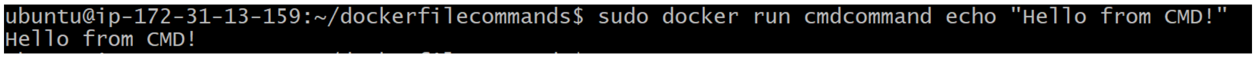
```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker run cmdcommand
curl 7.68.0 (x86_64-pc-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 brotli/1.0.7 libidn2/2.2.0 libpsl/0
.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2.3
```

4. **Override CMD:**

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

```
docker run cmdcommand echo "Hello from CMD!"
```

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



```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker run cmdcommand echo "Hello from CMD!"
Hello from CMD!
```

Part 4: Exploring ENTRYPOINT Command

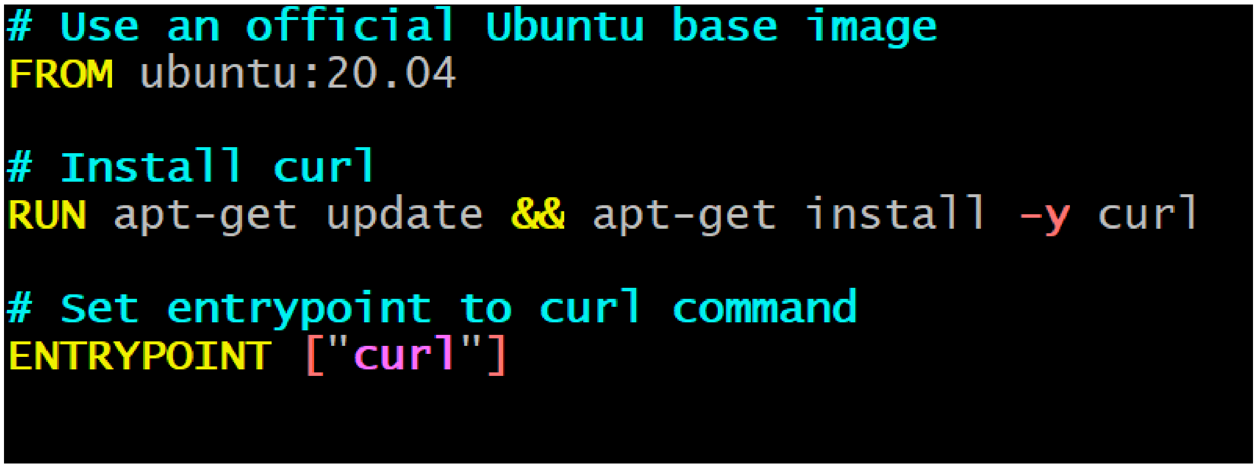
1. **Create a Dockerfile with ENTRYPOINT:**

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"]
```

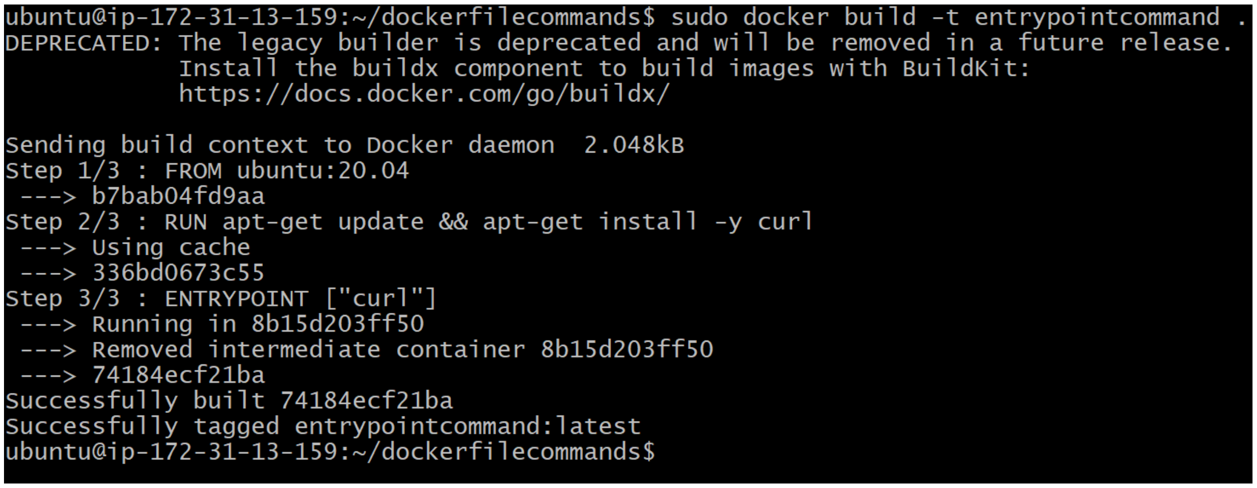


```
# Use an official Ubuntu base image
FROM ubuntu:20.04

# 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 entrypointcommand .
```



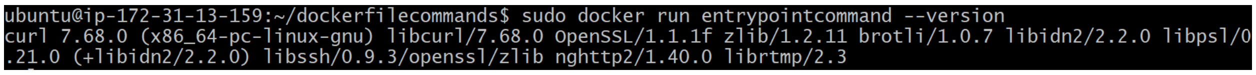
```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker build -t entrypointcommand .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  2.048kB
Step 1/3 : FROM ubuntu:20.04
 ---> b7bab04fd9aa
Step 2/3 : RUN apt-get update && apt-get install -y curl
 ---> Using cache
 ---> 336bd0673c55
Step 3/3 : ENTRYPOINT ["curl"]
 ---> Running in 8b15d203ff50
 ---> Removed intermediate container 8b15d203ff50
 ---> 74184ecf21ba
Successfully built 74184ecf21ba
Successfully tagged entrypointcommand:latest
ubuntu@ip-172-31-13-159:~/dockerfilecommands$
```

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 entrypointcommand --version
```

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.



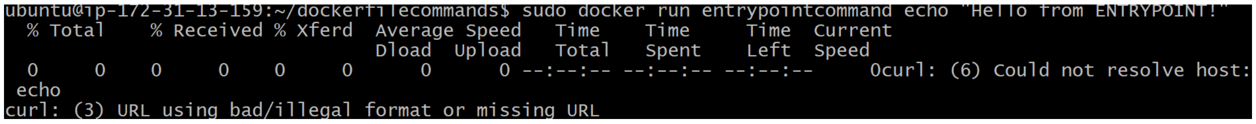
```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker run entrypointcommand --version
curl 7.68.0 (x86_64-pc-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 brotli/1.0.7 libidn2/2.2.0 libpsl/0
.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2.3
```

4. **Override ENTRYPOINT:**

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

```
docker run entrypointcommand echo "Hello from ENTRYPOINT!"
```

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

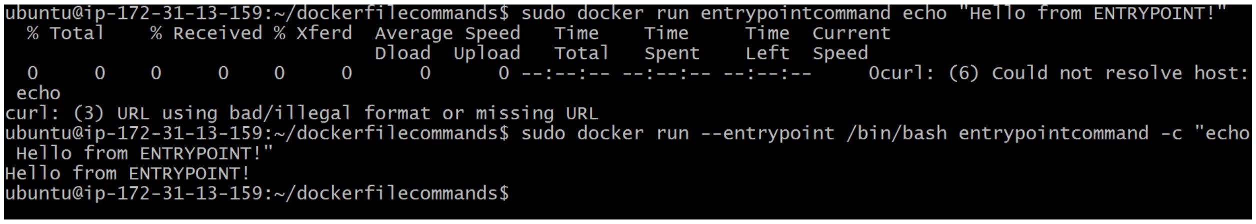


```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker run entrypointcommand echo "Hello from ENTRYPOINT!"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0curl: (6) Could not resolve host:
 echo
curl: (3) URL using bad/illegal format or missing URL
```

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

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

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



```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker run entrypointcommand 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 --:--:-- --:--:-- --:--:--     0curl: (6) Could not resolve host:
 echo
curl: (3) URL using bad/illegal format or missing URL
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker run --entrypoint /bin/bash entrypointcommand -c "echo
 Hello from ENTRYPOINT!"
Hello from ENTRYPOINT!
ubuntu@ip-172-31-13-159:~/dockerfilecommands$
```

Part 5: Combining CMD and ENTRYPOINT

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

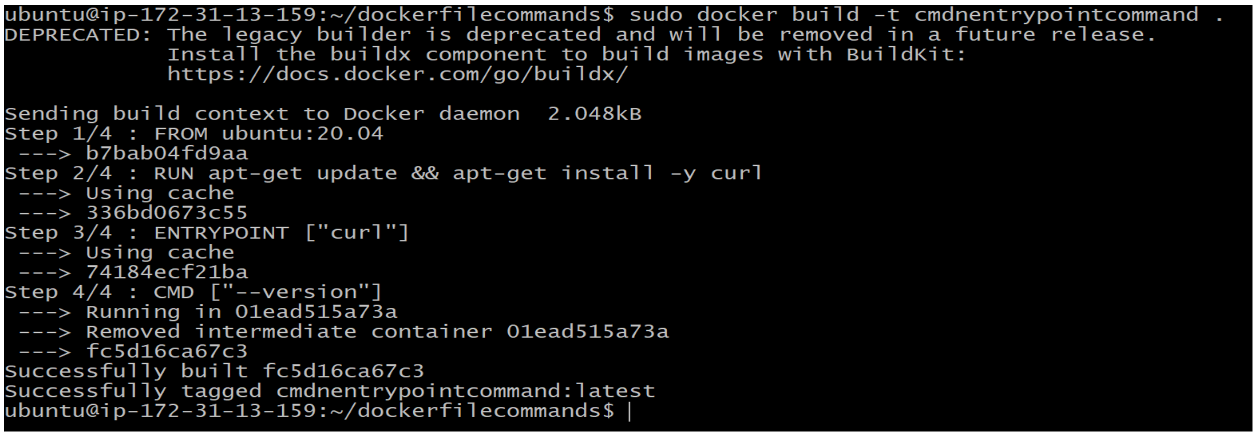
Modify the Dockerfile to use both CMD and ENTRYPOINT:

```
# 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"]
```

2. **Build the Image:**

Build the new image:

```
docker build -t cmdnentrypointcommand .
```



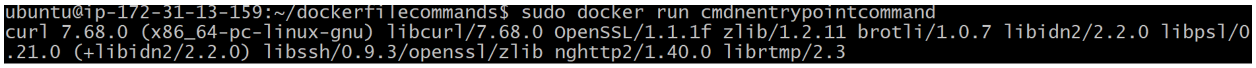
```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker build -t cmdnentrypointcommand .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  2.048kB
Step 1/4 : FROM ubuntu:20.04
 ---> b7bab04fd9aa
Step 2/4 : RUN apt-get update && apt-get install -y curl
 ---> Using cache
 ---> 336bd0673c55
Step 3/4 : ENTRYPOINT ["curl"]
 ---> Using cache
 ---> 74184ecf21ba
Step 4/4 : CMD ["--version"]
 ---> Running in 01ead515a73a
 ---> Removed intermediate container 01ead515a73a
 ---> fc5d16ca67c3
Successfully built fc5d16ca67c3
Successfully tagged cmdnentrypointcommand:latest
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ |
```

3. **Run the Container:**

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

```
docker run cmdnentrypointcommand
```

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



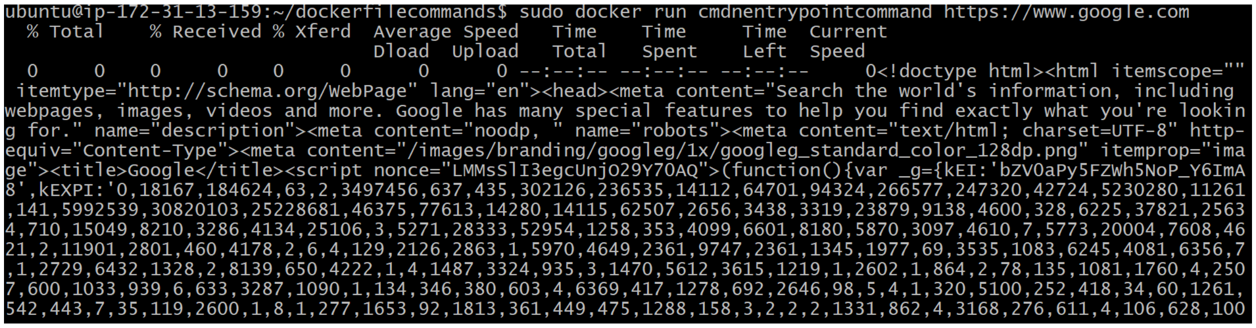
```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker run cmdnentrypointcommand
curl 7.68.0 (x86_64-pc-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 brotli/1.0.7 libidn2/2.2.0 libpsl/0
.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/zlib nghttp2/1.40.0 librtmp/2.3
```

4. **Override CMD Arguments:**

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

```
docker run cmdnentrypointcommand https://www.google.com
```

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



```
ubuntu@ip-172-31-13-159:~/dockerfilecommands$ sudo docker run cmdnentrypointcommand 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"><head><meta content="Search the world's information, including
webpages, images, videos and more. Google has many special features to help you find exactly what you're lookin
g for." name="description"><meta content="noodp, " name="robots"><meta content="text/html; charset=UTF-8" http-
equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="ima
ge"><title>Google</title><script nonce="LMMsSlI3egcUnjO29Y70AQ">(function(){var _g={kEI:'bZVOaPy5FZWh5NoP_Y6ImA
8',kEXPI:'0,18167,184624,63,2,3497456,637,435,302126,236535,14112,64701,94324,266577,247320,42724,5230280,11261
,141,5992539,30820103,25228681,46375,77613,14280,14115,62507,2656,3438,3319,23879,9138,4600,328,6225,37821,2563
4,710,15049,8210,3286,4134,25106,3,5271,28333,52954,1258,353,4099,6601,8180,5870,3097,4610,7,5773,20004,7608,46
21,2,11901,2801,460,4178,2,6,4,129,2126,2863,1,5970,4649,2361,9747,2361,1345,1977,69,3535,1083,6245,4081,6356,7
,1,2729,6432,1328,2,8139,650,4222,1,4,1487,3324,935,3,1470,5612,3615,1219,1,2602,1,864,2,78,135,1081,1760,4,250
7,600,1033,939,6,633,3287,1090,1,134,346,380,603,4,6369,417,1278,692,2646,98,5,4,1,320,5100,252,418,34,60,1261,
542,443,7,35,119,2600,1,8,1,277,1653,92,1813,361,449,475,1288,158,3,2,2,2,1331,862,4,3168,276,611,4,106,628,100
```

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.
