Lesson 02 Demo 04

Displaying Layers of a Docker Image

Objective: To display the layered structure of Docker images for providing insight into the hierarchical arrangement of image layers within Docker's architecture

Tools required: Ubuntu

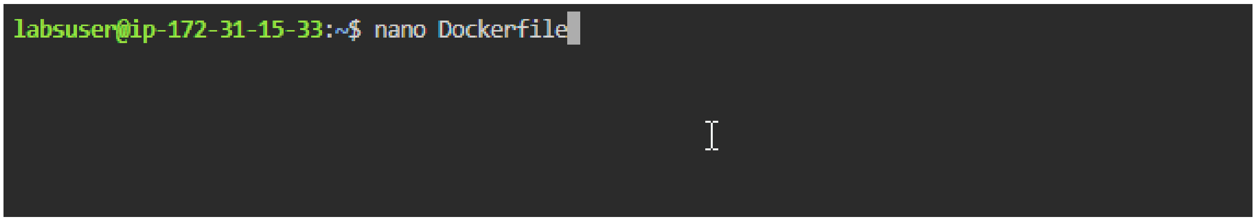
Prerequisites: None

Steps to be followed:

1. Create and display the layers of a Docker image

Step 1: Create and display the layers of a Docker image

1.1 Create a Docker file with multiple instructions using the following command:
nano Dockerfile



```
labsuser@ip-172-31-15-33:~$ nano Dockerfile
```

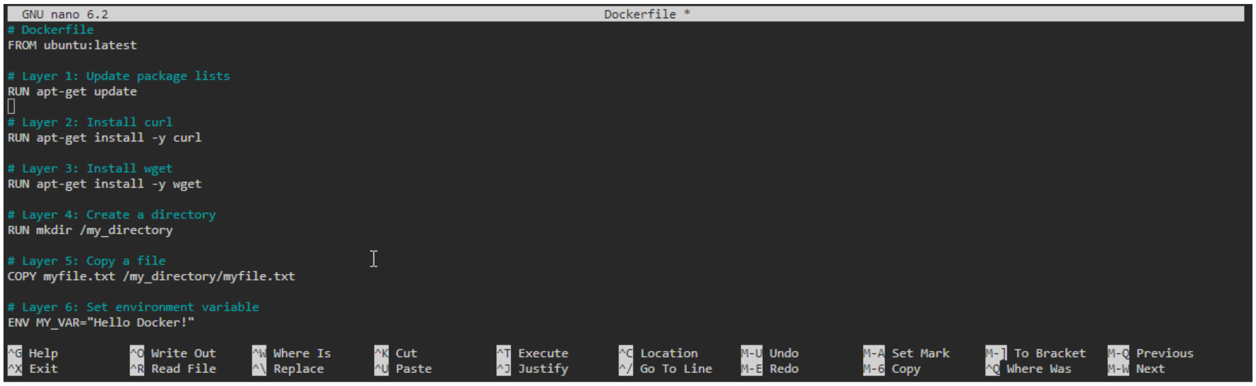
1.2 Add the following configurations in the **Dockerfile** as shown in the screenshot below:

```
# Dockerfile
FROM ubuntu:latest

# Layer 1: Update package lists
RUN apt-get update

# Layer 2: Install curl
RUN apt-get install -y curl

# Layer 3: Install wget
RUN apt-get install -y wget
```

# Layer 4: Create a directory
RUN mkdir /my_directory

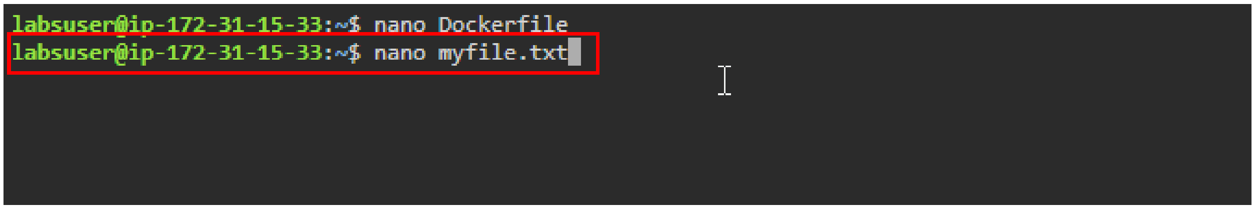
# Layer 5: Copy a file
COPY myfile.txt /my_directory/myfile.txt

# Layer 6: Set environment variable
ENV MY_VAR="Hello Docker!"



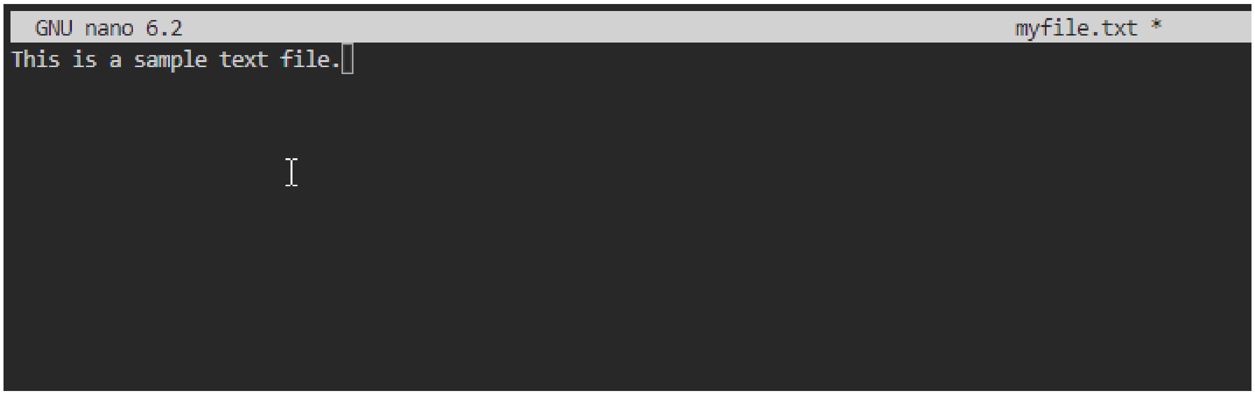
```
# Dockerfile
FROM ubuntu:latest

# Layer 1: Update package lists
RUN apt-get update

# Layer 2: Install curl
RUN apt-get install -y curl

# Layer 3: Install wget
RUN apt-get install -y wget

# Layer 4: Create a directory
RUN mkdir /my_directory

# Layer 5: Copy a file
COPY myfile.txt /my_directory/myfile.txt

# Layer 6: Set environment variable
ENV MY_VAR="Hello Docker!"
```

1.3 Create a text file using the following command:
nano myfile.txt



```
labsuser@ip-172-31-15-33:~$ nano Dockerfile
labsuser@ip-172-31-15-33:~$ nano myfile.txt
```

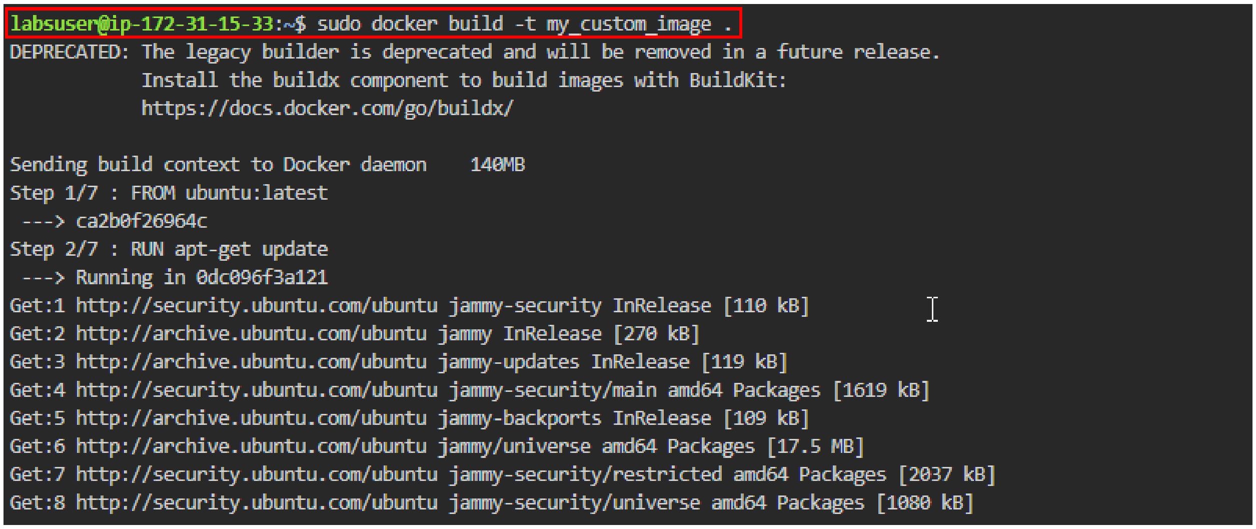
1.4 Add the following sentence in the **myfile.txt**:
This is a sample text file.



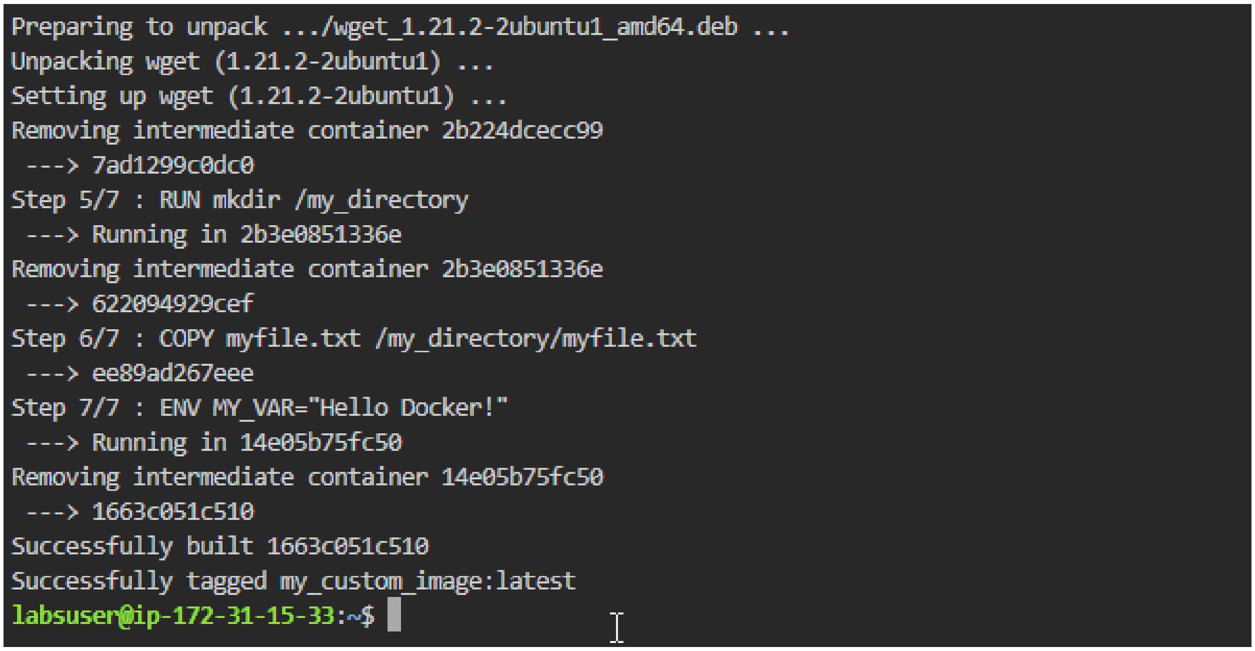
```
This is a sample text file.
```

1.5 Execute the following command to build the Docker image using the Dockerfile:

sudo docker build -t my_custom_image .

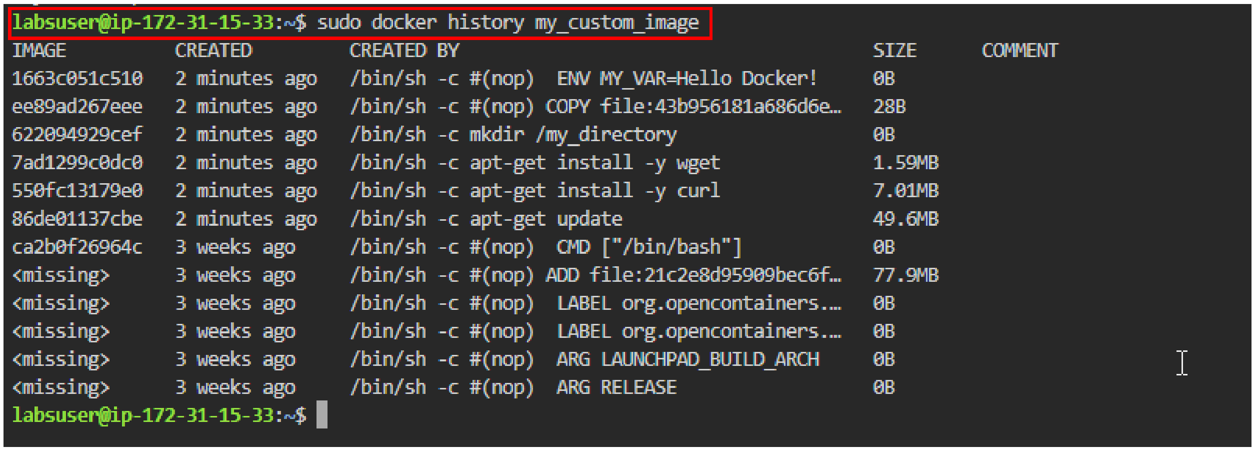


```
labsuser@ip-172-31-15-33:~$ sudo docker build -t my_custom_image .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon    140MB
Step 1/7 : FROM ubuntu:latest
 ---> ca2b0f26964c
Step 2/7 : RUN apt-get update
 ---> Running in 0dc096f3a121
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1619 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [2037 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1080 kB]
```



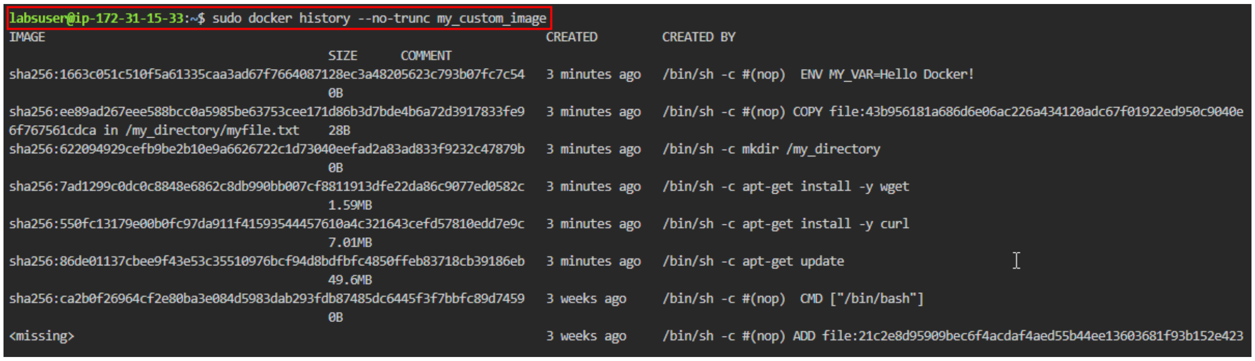
```
Preparing to unpack .../wget_1.21.2-2ubuntu1_amd64.deb ...
Unpacking wget (1.21.2-2ubuntu1) ...
Setting up wget (1.21.2-2ubuntu1) ...
Removing intermediate container 2b224dcecc99
 ---> 7ad1299c0dc0
Step 5/7 : RUN mkdir /my_directory
 ---> Running in 2b3e0851336e
Removing intermediate container 2b3e0851336e
 ---> 622094929cef
Step 6/7 : COPY myfile.txt /my_directory/myfile.txt
 ---> ee89ad267eee
Step 7/7 : ENV MY_VAR="Hello Docker!"
 ---> Running in 14e05b75fc50
Removing intermediate container 14e05b75fc50
 ---> 1663c051c510
Successfully built 1663c051c510
Successfully tagged my_custom_image:latest
labsuser@ip-172-31-15-33:~$
```

1.6 Execute the following command to view the history of the image, including all the layers:
sudo docker history my_custom_image

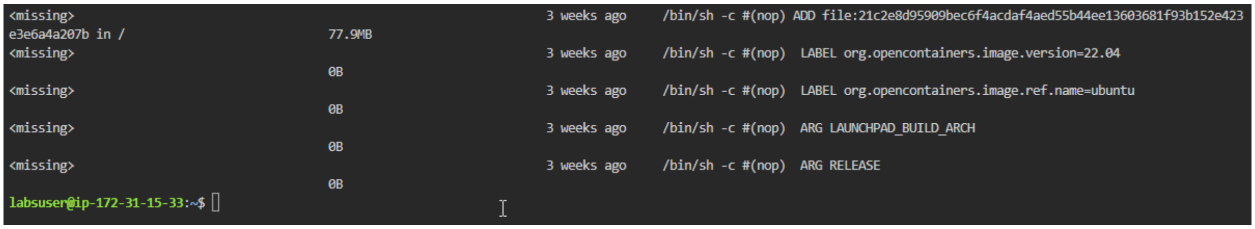


```
labsuser@ip-172-31-15-33:~$ sudo docker history my_custom_image
IMAGE          CREATED         CREATED BY                                      SIZE      COMMENT
1663c051c510   2 minutes ago   /bin/sh -c #(nop)  ENV MY_VAR=Hello Docker!     0B
ee89ad267eee   2 minutes ago   /bin/sh -c #(nop) COPY file:43b956181a686d6e…   28B
622094929cef   2 minutes ago   /bin/sh -c mkdir /my_directory                  0B
7ad1299c0dc0   2 minutes ago   /bin/sh -c apt-get install -y wget              1.59MB
550fc13179e0   2 minutes ago   /bin/sh -c apt-get install -y curl              7.01MB
86de01137cbe   2 minutes ago   /bin/sh -c apt-get update                       49.6MB
ca2b0f26964c   3 weeks ago     /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>      3 weeks ago     /bin/sh -c #(nop) ADD file:21c2e8d95909bec6f…   77.9MB
<missing>      3 weeks ago     /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      3 weeks ago     /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      3 weeks ago     /bin/sh -c #(nop)  ARG LAUNCHPAD_BUILD_ARCH     0B
<missing>      3 weeks ago     /bin/sh -c #(nop)  ARG RELEASE                  0B
labsuser@ip-172-31-15-33:~$
```

1.7 Run the following command to inspect individual layers of the image using their respective layer IDs:
sudo docker history --no-trunc my_custom_image



```
labsuser@ip-172-31-15-33:~$ sudo docker history --no-trunc my_custom_image
IMAGE                                                                     CREATED         CREATED BY
                                            SIZE      COMMENT
sha256:1663c051c510f5a61335caa3ad67f7664087128ec3a48205623c793b07fc7c54   3 minutes ago   /bin/sh -c #(nop)  ENV MY_VAR=Hello Docker!
                                            0B
sha256:ee89ad267eee588bcc0a5985be63753cee171d86b3d7bde4b6a72d3917833fe9   3 minutes ago   /bin/sh -c #(nop) COPY file:43b956181a686d6e06ac226a434120adc67f01922ed950c9040e
6f767561cdca in /my_directory/myfile.txt    28B
sha256:622094929cefb9be2b10e9a6626722c1d73040eefad2a83ad833f9232c47879b   3 minutes ago   /bin/sh -c mkdir /my_directory
                                            0B
sha256:7ad1299c0dc0c8848e6862c8db990bb007cf8811913dfe22da86c9077ed0582c   3 minutes ago   /bin/sh -c apt-get install -y wget
                                            1.59MB
sha256:550fc13179e00b0fc97da911f41593544457610a4c321643cefd57810edd7e9c   3 minutes ago   /bin/sh -c apt-get install -y curl
                                            7.01MB
sha256:86de01137cbee9f43e53c35510976bcf94d8bdfbfc4850ffeb83718cb39186eb   3 minutes ago   /bin/sh -c apt-get update
                                            49.6MB
sha256:ca2b0f26964cf2e80ba3e084d5983dab293fdb87485dc6445f3f7bbfc89d7459   3 weeks ago     /bin/sh -c #(nop)  CMD ["/bin/bash"]
                                            0B
<missing>                                                                 3 weeks ago     /bin/sh -c #(nop) ADD file:21c2e8d95909bec6f4acdaf4aed55b44ee13603681f93b152e423
```



```
<missing>                                                                 3 weeks ago     /bin/sh -c #(nop) ADD file:21c2e8d95909bec6f4acdaf4aed55b44ee13603681f93b152e423
e3e6a4a207b in /                            77.9MB
<missing>                                                                 3 weeks ago     /bin/sh -c #(nop)  LABEL org.opencontainers.image.version=22.04
                                            0B
<missing>                                                                 3 weeks ago     /bin/sh -c #(nop)  LABEL org.opencontainers.image.ref.name=ubuntu
                                            0B
<missing>                                                                 3 weeks ago     /bin/sh -c #(nop)  ARG LAUNCHPAD_BUILD_ARCH
                                            0B
<missing>                                                                 3 weeks ago     /bin/sh -c #(nop)  ARG RELEASE
                                            0B
labsuser@ip-172-31-15-33:~$
```

By following these steps, you have successfully displayed the layered structure of Docker images to provide insight into the hierarchical arrangement of image layers within Docker's architecture.