Docker Creation

This document shows how you can create your own docker and use it as per your required tool and operating system.

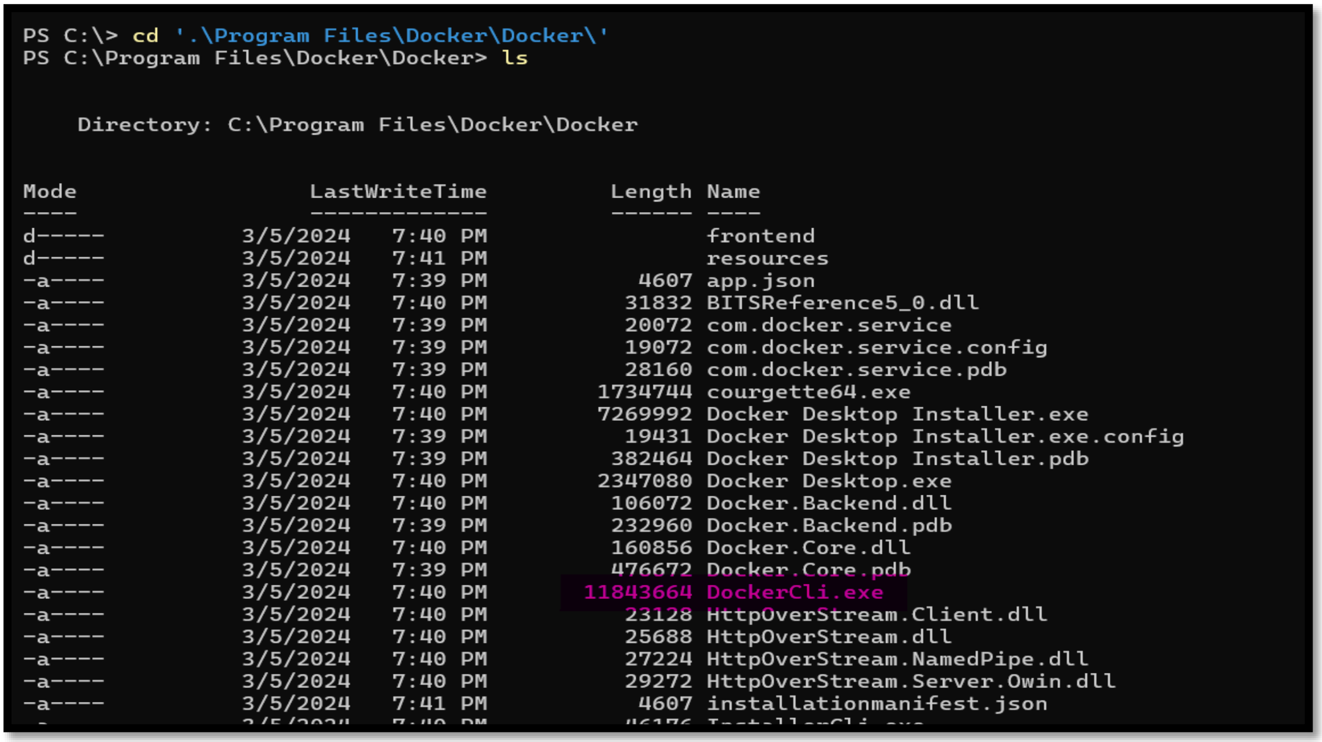
1. Windows as Host Machine

Step 1: Install the Docker Desktop from: https://www.docker.com/products/docker-desktop/\n\n\n

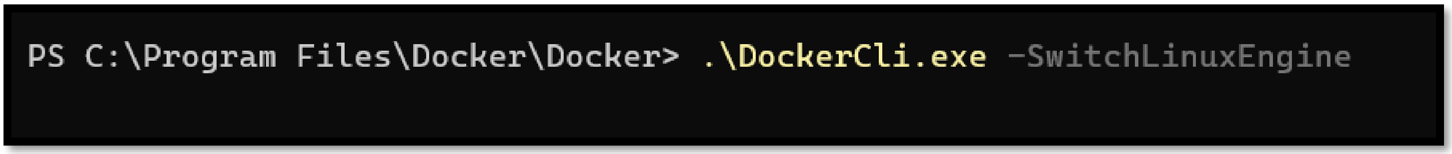
Step2: Please run the installer after the download is completed.

Step 3: Open your terminal and dive into this path C:\Program Files\Docker\Docker and run this command.

C: >> **.\DockerCli.exe -SwitchLinuxEngine**



```
PS C:\> cd '.\Program Files\Docker\Docker\'
PS C:\Program Files\Docker\Docker> ls


    Directory: C:\Program Files\Docker\Docker


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          3/5/2024   7:40 PM                frontend
d-----          3/5/2024   7:41 PM                resources
-a----          3/5/2024   7:39 PM           4607 app.json
-a----          3/5/2024   7:40 PM          31832 BITSReference5_0.dll
-a----          3/5/2024   7:39 PM          20072 com.docker.service
-a----          3/5/2024   7:39 PM          19072 com.docker.service.config
-a----          3/5/2024   7:39 PM          28160 com.docker.service.pdb
-a----          3/5/2024   7:40 PM        1734744 courgette64.exe
-a----          3/5/2024   7:40 PM        7269992 Docker Desktop Installer.exe
-a----          3/5/2024   7:39 PM          19431 Docker Desktop Installer.exe.config
-a----          3/5/2024   7:39 PM         382464 Docker Desktop Installer.pdb
-a----          3/5/2024   7:40 PM        2347080 Docker Desktop.exe
-a----          3/5/2024   7:40 PM         106072 Docker.Backend.dll
-a----          3/5/2024   7:39 PM         232960 Docker.Backend.pdb
-a----          3/5/2024   7:40 PM         160856 Docker.Core.dll
-a----          3/5/2024   7:39 PM         476672 Docker.Core.pdb
-a----          3/5/2024   7:40 PM       11843664 DockerCli.exe
-a----          3/5/2024   7:40 PM          23128 HttpOverStream.Client.dll
-a----          3/5/2024   7:40 PM          25688 HttpOverStream.dll
-a----          3/5/2024   7:40 PM          27224 HttpOverStream.NamedPipe.dll
-a----          3/5/2024   7:40 PM          29272 HttpOverStream.Server.Owin.dll
-a----          3/5/2024   7:41 PM           4607 installationmanifest.json
-a----          3/5/2024   7:40 PM         461176 InstallerCli.exe
```

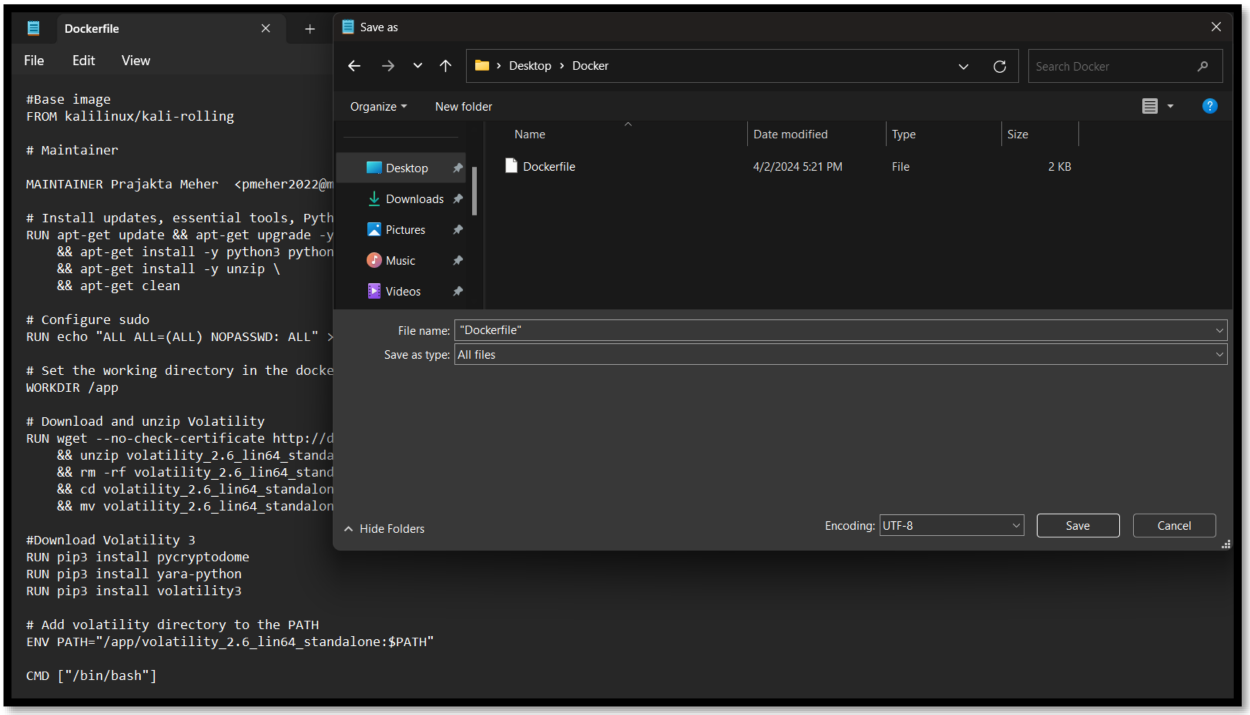


```
PS C:\Program Files\Docker\Docker> .\DockerCli.exe -SwitchLinuxEngine
```

Step 4: Creating the Docker file:

Note: The following code is designed using kali Linux as image file and installs tools such as Volatility 2 and 3. However, it can be adapted to install other tools and run on different operating systems such as Windows, Linux, or macOS.

- Step1: Begin by opening Notepad and writing your code to meet your specific requirements.
- Step 2: Save the file without any file extension to ensure Docker accepts it during the build process. When saving the file, enclose the filename in double quotes, such as ("Dockerfile"), and select "All Files" as the file type before saving it.



```
#Base image
FROM kalilinux/kali-rolling

# Maintainer

MAINTAINER Prajakta Meher  <pmeher2022@m

# Install updates, essential tools, Pyth
RUN apt-get update && apt-get upgrade -y
    && apt-get install -y python3 python
    && apt-get install -y unzip \
    && apt-get clean

# Configure sudo
RUN echo "ALL ALL=(ALL) NOPASSWD: ALL" >

# Set the working directory in the docke
WORKDIR /app

# Download and unzip Volatility
RUN wget --no-check-certificate http://d
    && unzip volatility_2.6_lin64_standa
    && rm -rf volatility_2.6_lin64_stand
    && cd volatility_2.6_lin64_standalon
    && mv volatility_2.6_lin64_standalon

#Download Volatility 3
RUN pip3 install pycryptodome
RUN pip3 install yara-python
RUN pip3 install volatility3

# Add volatility directory to the PATH
ENV PATH="/app/volatility_2.6_lin64_standalone:$PATH"

CMD ["/bin/bash"]
```

Name	Date modified	Type	Size
Dockerfile	4/2/2024 5:21 PM	File	2 KB

File name: "Dockerfile"

Save as type: All files

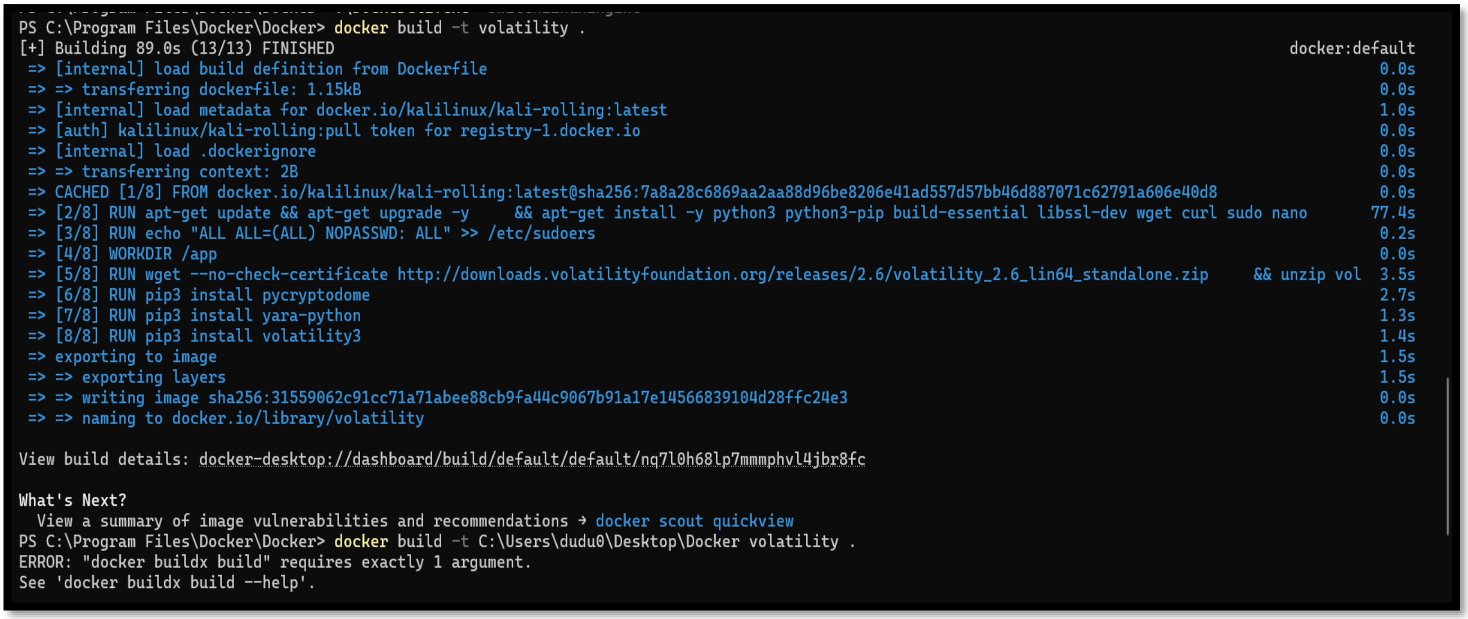
Encoding: UTF-8

Script:

```
#Base image
FROM kalilinux/kali-rolling



# Install updates, essential tools, Python3, and pip
RUN apt-get update && apt-get upgrade -y \
    && apt-get install -y python3 python3-pip build-essential libssl-dev wget curl sudo nano \
    && apt-get install -y unzip \
    && apt-get clean

# Configure sudo
RUN echo "ALL ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

# Set the working directory in the docker image
WORKDIR /app

# Download and unzip Volatility
RUN wget --no-check-certificate http://downloads.volatilityfoundation.org/releases/2.6/volatility_2.6_lin64_standalone.zip \
    && unzip volatility_2.6_lin64_standalone.zip \
    && rm -rf volatility_2.6_lin64_standalone.zip \
    && cd volatility_2.6_lin64_standalone/ \
    && mv volatility_2.6_lin64_standalone volatility

#Download Volatility 3
RUN pip3 install pycryptodome
RUN pip3 install yara-python
RUN pip3 install volatility3

# Add volatility directory to the PATH
ENV PATH="/app/volatility_2.6_lin64_standalone:$PATH"

CMD ["/bin/bash"]
```

Step 5: Building the docker image:

1. Once you save your file open Command Prompt or PowerShell.
2. Navigate to the directory containing your Dockerfile.
3. Run the following command to build the Docker image:
    a. C: >> docker build -t <image_name> .
4. Replace <image_name> with the desired name for your Docker image. Don't forget the period at the end, which indicates the current directory as the build context.

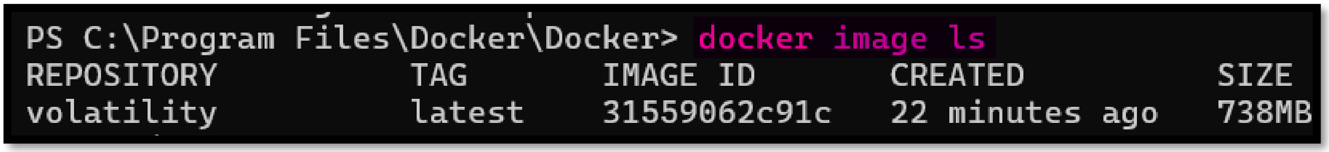


```
PS C:\Program Files\Docker\Docker> docker build -t volatility .
[+] Building 89.0s (13/13) FINISHED                                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                                             0.0s
 => => transferring dockerfile: 1.15kB                                                                                           0.0s
 => [internal] load metadata for docker.io/kalilinux/kali-rolling:latest                                                         1.0s
 => [auth] kalilinux/kali-rolling:pull token for registry-1.docker.io                                                            0.0s
 => [internal] load .dockerignore                                                                                                0.0s
 => => transferring context: 2B                                                                                                  0.0s
 => CACHED [1/8] FROM docker.io/kalilinux/kali-rolling:latest@sha256:7a8a28c6869aa2aa88d96be8206e41ad557d57bb46d887071c62791a606e40d8  0.0s
 => [2/8] RUN apt-get update && apt-get upgrade -y     && apt-get install -y python3 python3-pip build-essential libssl-dev wget curl sudo nano      77.4s
 => [3/8] RUN echo "ALL ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers                                                                 0.2s
 => [4/8] WORKDIR /app                                                                                                           0.0s
 => [5/8] RUN wget --no-check-certificate http://downloads.volatilityfoundation.org/releases/2.6/volatility_2.6_lin64_standalone.zip     && unzip vol  3.5s
 => [6/8] RUN pip3 install pycryptodome                                                                                          2.7s
 => [7/8] RUN pip3 install yara-python                                                                                           1.3s
 => [8/8] RUN pip3 install volatility3                                                                                           1.4s
 => exporting to image                                                                                                           1.5s
 => => exporting layers                                                                                                          1.5s
 => => writing image sha256:31559062c91cc71a71abee88cb9fa44c9067b91a17e14566839104d28ffc24e3                                     0.0s
 => => naming to docker.io/library/volatility                                                                                    0.0s

View build details: docker-desktop://dashboard/build/default/default/nq7l0h68lp7mmmphvl4jbr8fc

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Program Files\Docker\Docker> docker build -t C:\Users\dudu0\Desktop\Docker volatility .
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.
```

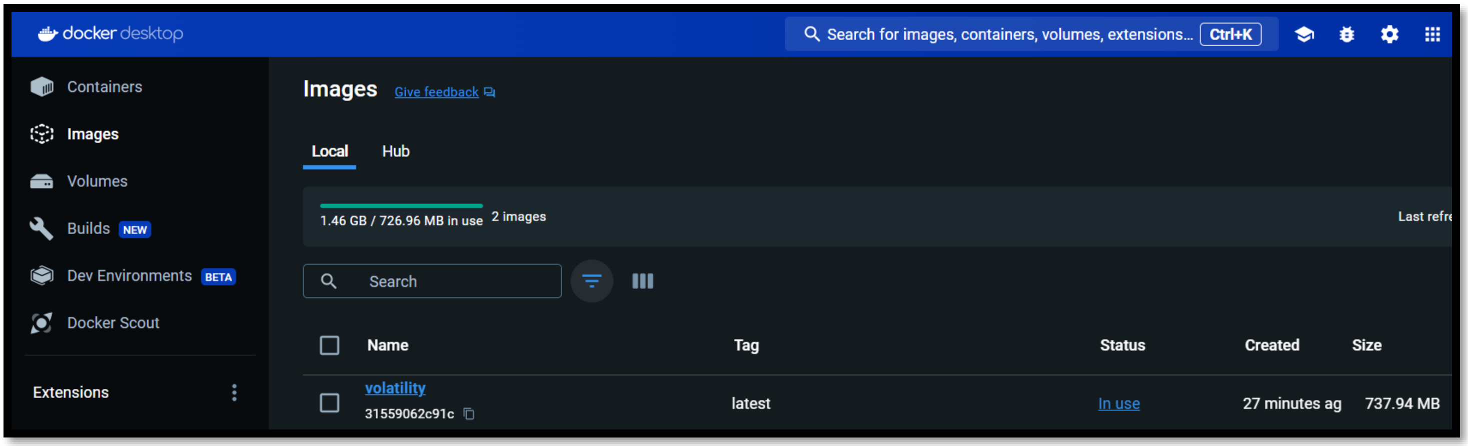
Step 6: Checking the docker images:

Use the below command to check if your image is built successfully or not.

C: >> docker image ls



```
PS C:\Program Files\Docker\Docker> docker image ls
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
volatility          latest    31559062c91c   22 minutes ago   738MB
```

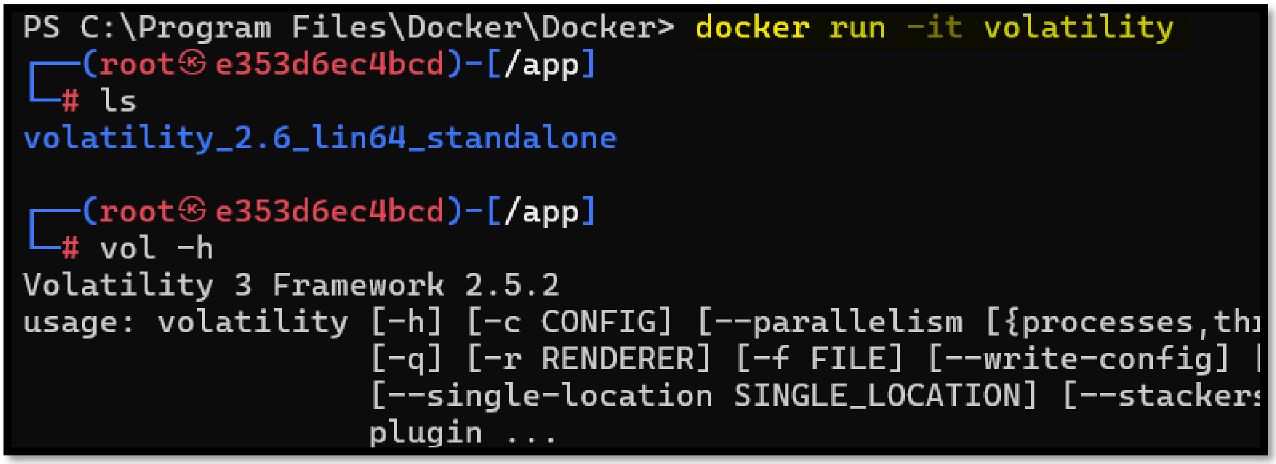


1.46 GB / 726.96 MB in use 2 images

□	Name	Tag	Status	Created	Size
□	volatility 31559062c91c	latest	In use	27 minutes ag	737.94 MB

Step 7: Running the docker:

C: >> docker run -it volatility



```
PS C:\Program Files\Docker\Docker> docker run -it volatility
┌──(root㉿e353d6ec4bcd)-[/app]
└─# ls
volatility_2.6_lin64_standalone

┌──(root㉿e353d6ec4bcd)-[/app]
└─# vol -h
Volatility 3 Framework 2.5.2
usage: volatility [-h] [-c CONFIG] [--parallelism [{processes,thı
                  [-q] [-r RENDERER] [-f FILE] [--write-config] |
                  [--single-location SINGLE_LOCATION] [--stackers
                  plugin ...
```

Here I have set volatility to the environment path in the docker script. Therefore, you can execute volatility anywhere from the terminal.

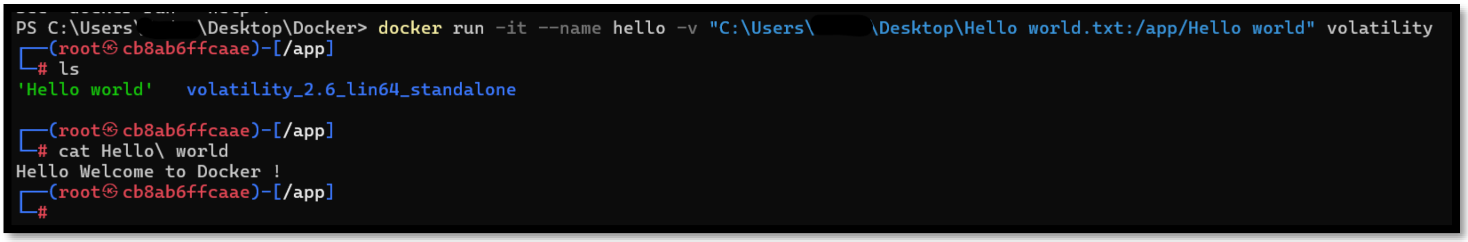
Step 8: Accessing and sharing files from main Host machine.

If you want to share files from your host machine to docker file, you can use the steps below.

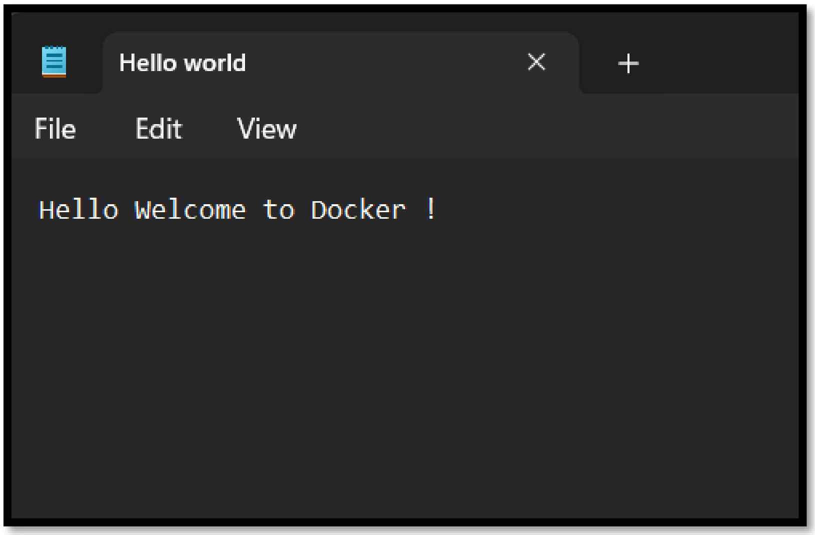
1. Open Command Prompt or PowerShell.
2. And run.

C:>> docker run -it --name any_random_name -v C:\\Users\\path_of_your_file :/app/file name your_image name

Example: C:>> docker run -it --name user -v C:\\Users\\admin\Desktop\Hello world :/app/ Hello world volatility

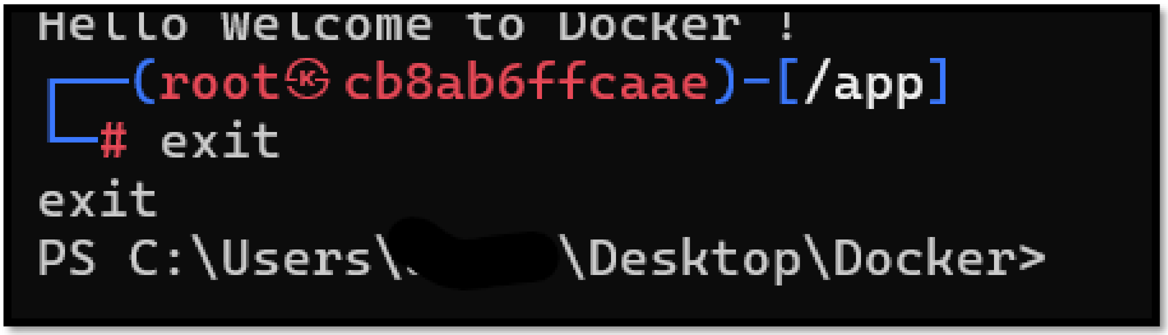


```
PS C:\Users\[redacted]\Desktop\Docker> docker run -it --name hello -v "C:\Users\[redacted]\Desktop\Hello world.txt:/app/Hello world" volatility
┌──(root㉿cb8ab6ffcaae)-[/app]
└─# ls
'Hello world'   volatility_2.6_lin64_standalone

┌──(root㉿cb8ab6ffcaae)-[/app]
└─# cat Hello\ world
Hello Welcome to Docker !
┌──(root㉿cb8ab6ffcaae)-[/app]
└─#
```



```
Hello Welcome to Docker !
```

Step 9: How to exit the running container:



```
Hello Welcome to Docker !
┌──(root㉿cb8ab6ffcaae)-[/app]
└─# exit
exit
PS C:\Users\[redacted]\Desktop\Docker>
```

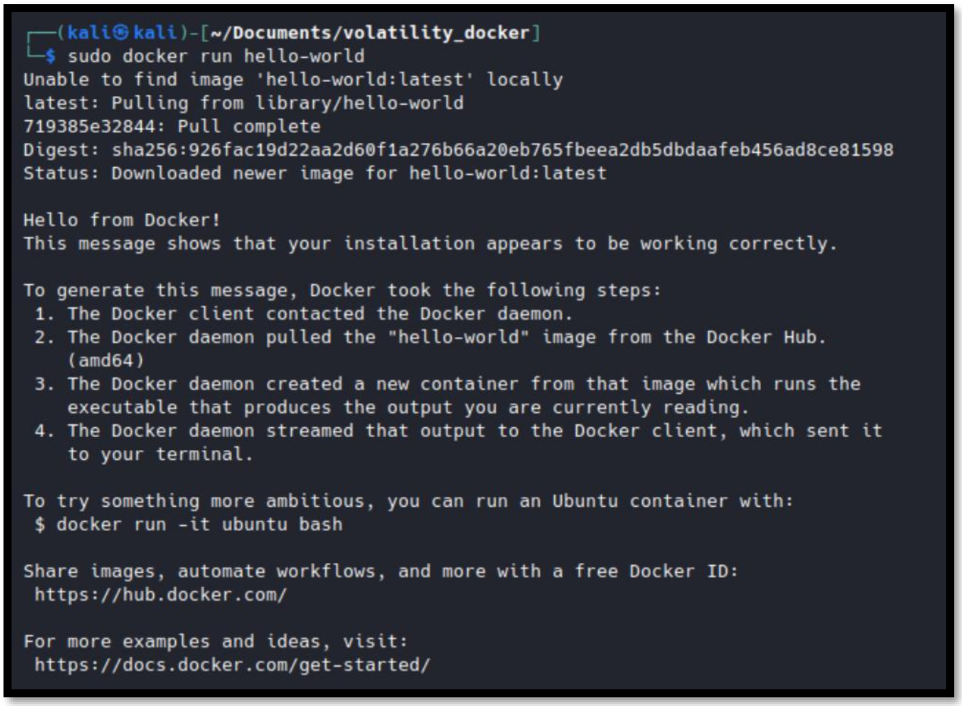
2.Linux as Host Machine

Step 1 : Installing docker in a linux machine:

$ sudo apt install docker.io

Once the docker is installed, we can verify that by running a sample docker image:

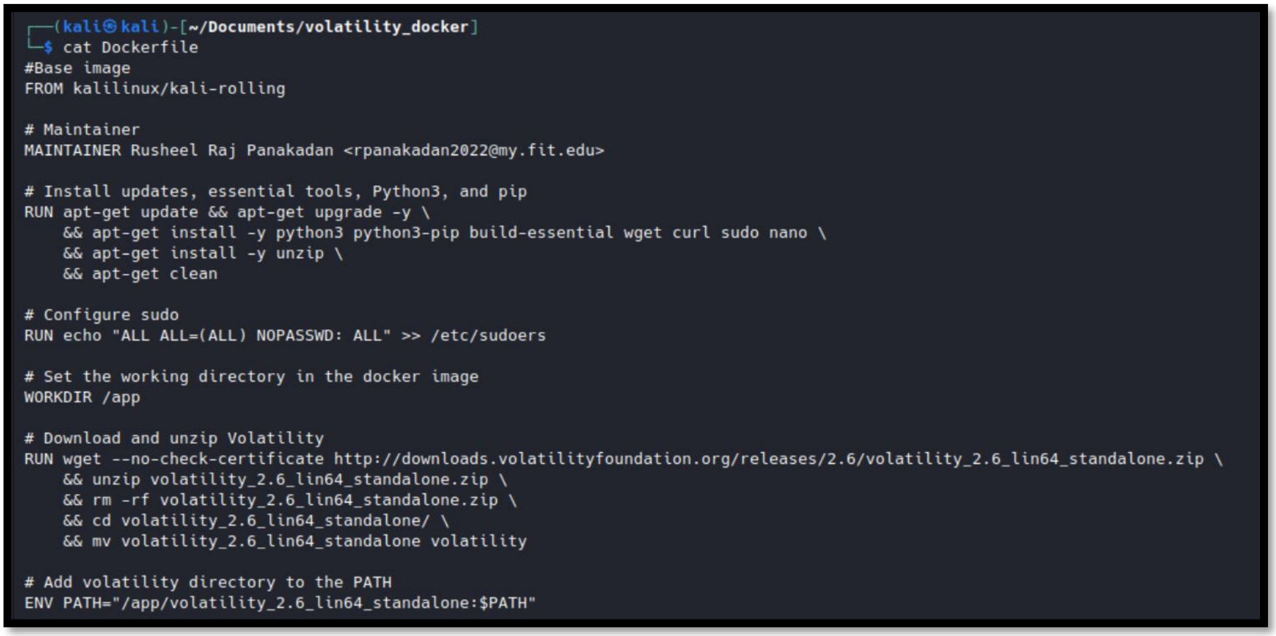
$ sudo docker run hello-world



```
┌──(kali㉿kali)-[~/Documents/volatility_docker]
└─$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:926fac19d22aa2d60f1a276b66a20eb765fbeea2db5dbdaafeb456ad8ce81598
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Step 2: Creating the Dockerfile:

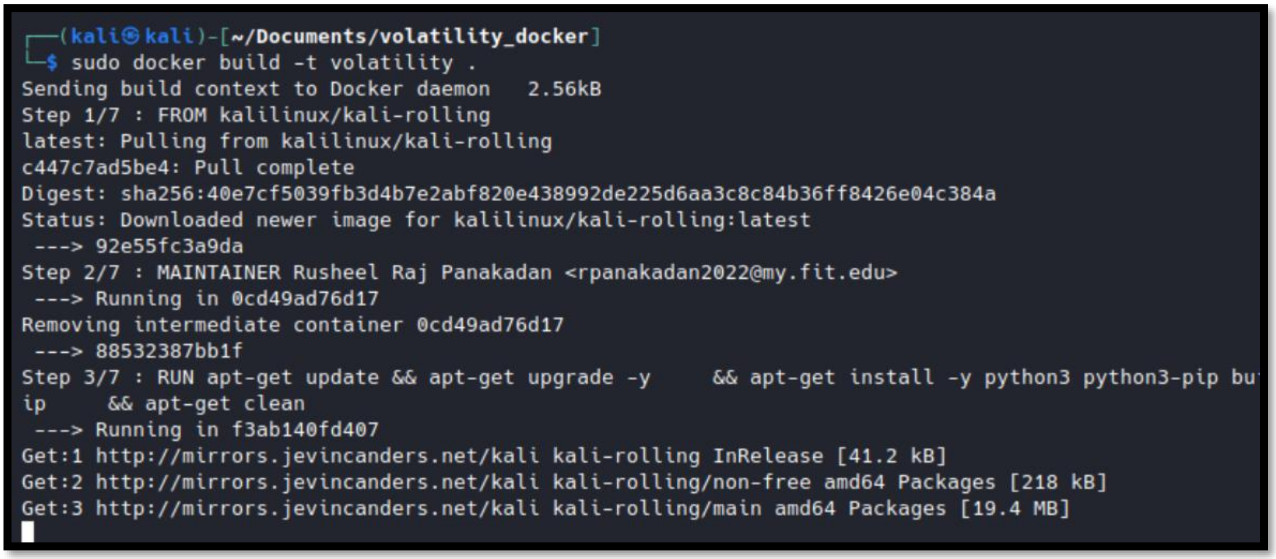
Start by creating a text file named Dockerfile (without any file extension) in a directory. This file will contain instructions for building your Docker image.



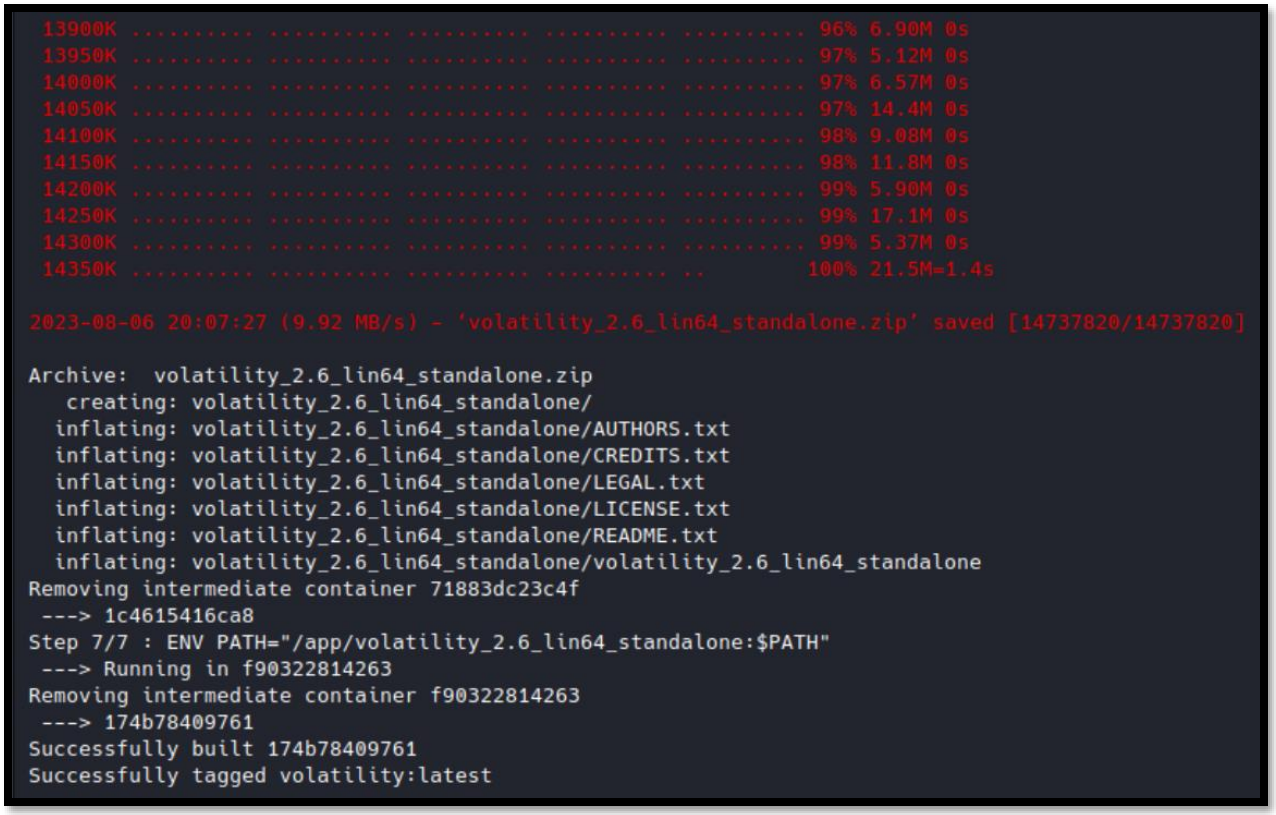
```
┌──(kali㉿kali)-[~/Documents/volatility_docker]
└─$ cat Dockerfile
#Base image
FROM kalilinux/kali-rolling

# Maintainer
MAINTAINER Rusheel Raj Panakadan <rpanakadan2022@my.fit.edu>

# Install updates, essential tools, Python3, and pip
RUN apt-get update && apt-get upgrade -y \
    && apt-get install -y python3 python3-pip build-essential wget curl sudo nano \
    && apt-get install -y unzip \
    && apt-get clean

# Configure sudo
RUN echo "ALL ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

# Set the working directory in the docker image
WORKDIR /app

# Download and unzip Volatility
RUN wget --no-check-certificate http://downloads.volatilityfoundation.org/releases/2.6/volatility_2.6_lin64_standalone.zip \
    && unzip volatility_2.6_lin64_standalone.zip \
    && rm -rf volatility_2.6_lin64_standalone.zip \
    && cd volatility_2.6_lin64_standalone/ \
    && mv volatility_2.6_lin64_standalone volatility

# Add volatility directory to the PATH
ENV PATH="/app/volatility_2.6_lin64_standalone:$PATH"
```

Step 3: Building the docker image:

$ sudo docker build -t volatility .



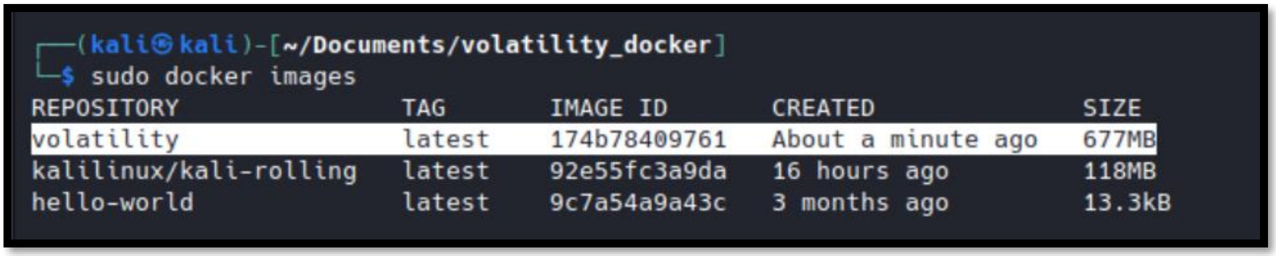
```
┌──(kali㉿kali)-[~/Documents/volatility_docker]
└─$ sudo docker build -t volatility .
Sending build context to Docker daemon   2.56kB
Step 1/7 : FROM kalilinux/kali-rolling
latest: Pulling from kalilinux/kali-rolling
c447c7ad5be4: Pull complete
Digest: sha256:40e7cf5039fb3d4b7e2abf820e438992de225d6aa3c8c84b36ff8426e04c384a
Status: Downloaded newer image for kalilinux/kali-rolling:latest
 ---> 92e55fc3a9da
Step 2/7 : MAINTAINER Rusheel Raj Panakadan <rpanakadan2022@my.fit.edu>
 ---> Running in 0cd49ad76d17
Removing intermediate container 0cd49ad76d17
 ---> 88532387bb1f
Step 3/7 : RUN apt-get update && apt-get upgrade -y     && apt-get install -y python3 python3-pip bu
ip     && apt-get clean
 ---> Running in f3ab140fd407
Get:1 http://mirrors.jevincanders.net/kali kali-rolling InRelease [41.2 kB]
Get:2 http://mirrors.jevincanders.net/kali kali-rolling/non-free amd64 Packages [218 kB]
Get:3 http://mirrors.jevincanders.net/kali kali-rolling/main amd64 Packages [19.4 MB]
█
```



```
 13900K .......... .......... .......... .......... .......... 96% 6.90M 0s
 13950K .......... .......... .......... .......... .......... 97% 5.12M 0s
 14000K .......... .......... .......... .......... .......... 97% 6.57M 0s
 14050K .......... .......... .......... .......... .......... 97% 14.4M 0s
 14100K .......... .......... .......... .......... .......... 98% 9.08M 0s
 14150K .......... .......... .......... .......... .......... 98% 11.8M 0s
 14200K .......... .......... .......... .......... .......... 99% 5.90M 0s
 14250K .......... .......... .......... .......... .......... 99% 17.1M 0s
 14300K .......... .......... .......... .......... .......... 99% 5.37M 0s
 14350K .......... .......... .......... .......... ..        100% 21.5M=1.4s

2023-08-06 20:07:27 (9.92 MB/s) - 'volatility_2.6_lin64_standalone.zip' saved [14737820/14737820]

Archive:  volatility_2.6_lin64_standalone.zip
   creating: volatility_2.6_lin64_standalone/
  inflating: volatility_2.6_lin64_standalone/AUTHORS.txt
  inflating: volatility_2.6_lin64_standalone/CREDITS.txt
  inflating: volatility_2.6_lin64_standalone/LEGAL.txt
  inflating: volatility_2.6_lin64_standalone/LICENSE.txt
  inflating: volatility_2.6_lin64_standalone/README.txt
  inflating: volatility_2.6_lin64_standalone/volatility_2.6_lin64_standalone
Removing intermediate container 71883dc23c4f
 ---> 1c4615416ca8
Step 7/7 : ENV PATH="/app/volatility_2.6_lin64_standalone:$PATH"
 ---> Running in f90322814263
Removing intermediate container f90322814263
 ---> 174b78409761
Successfully built 174b78409761
Successfully tagged volatility:latest
```

Step 4: Checking the docker images:

$ sudo docker images

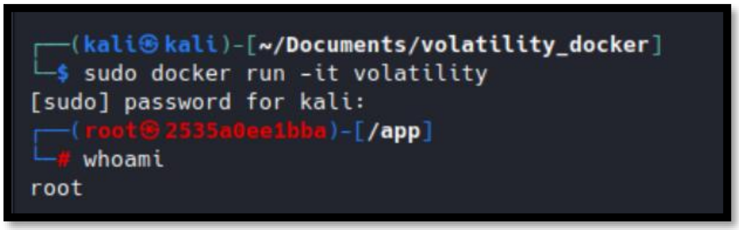


```
┌──(kali㉿kali)-[~/Documents/volatility_docker]
└─$ sudo docker images
REPOSITORY               TAG       IMAGE ID       CREATED              SIZE
volatility               latest    174b78409761   About a minute ago   677MB
kalilinux/kali-rolling   latest    92e55fc3a9da   16 hours ago         118MB
hello-world              latest    9c7a54a9a43c   3 months ago         13.3kB
```

Now, we can confirm that our volatility image has been built successfully.

Step 5: Running the docker:

$ sudo docker run -it volatility

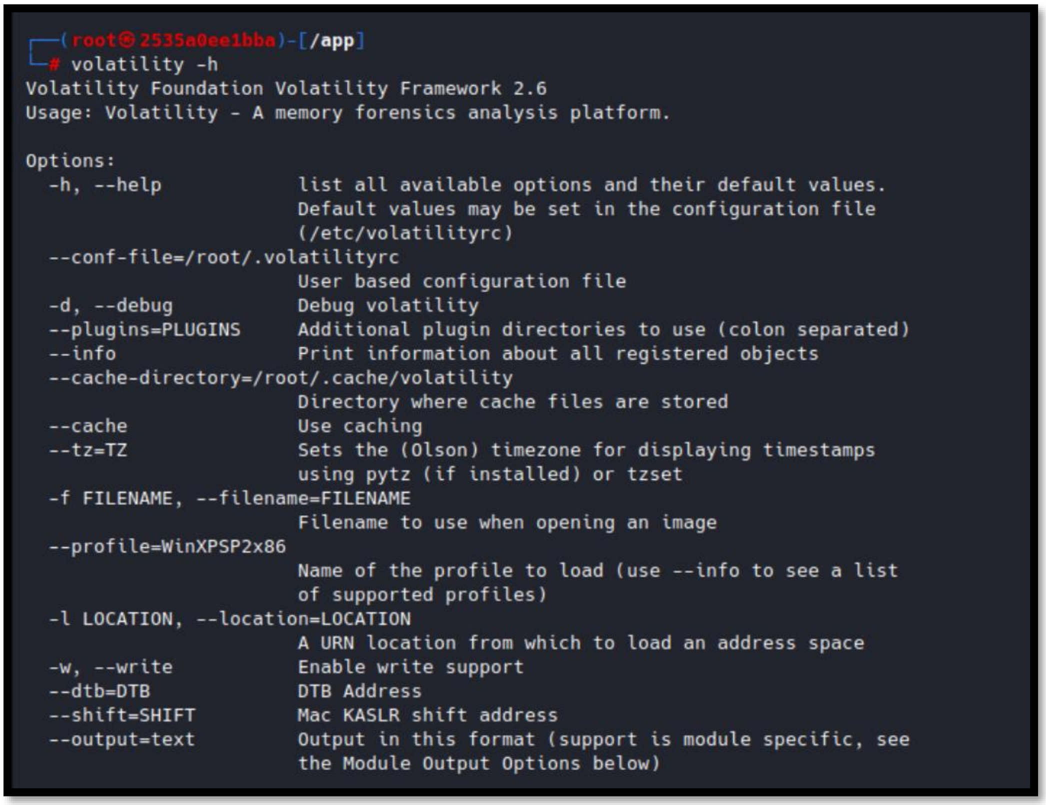


```
┌──(kali㉿kali)-[~/Documents/volatility_docker]
└─$ sudo docker run -it volatility
[sudo] password for kali:
┌──(root㉿2535a0ee1bba)-[/app]
└─# whoami
root
```

Step 6: Running Volatility inside the docker container:

I have set volatility to the environment path in the docker script. Therefore, you can execute volatility anywhere from the terminal.

All you need to type is just: volatility!



```
┌──(root㉿2535a0ee1bba)-[/app]
└─# volatility -h
Volatility Foundation Volatility Framework 2.6
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=/root/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (colon separated)
  --info                Print information about all registered objects
  --cache-directory=/root/.cache/volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the (Olson) timezone for displaying timestamps
                        using pytz (if installed) or tzset
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load (use --info to see a list
                        of supported profiles)
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --shift=SHIFT         Mac KASLR shift address
  --output=text         Output in this format (support is module specific, see
                        the Module Output Options below)
```

Step 8: Attaching USB device with the docker:

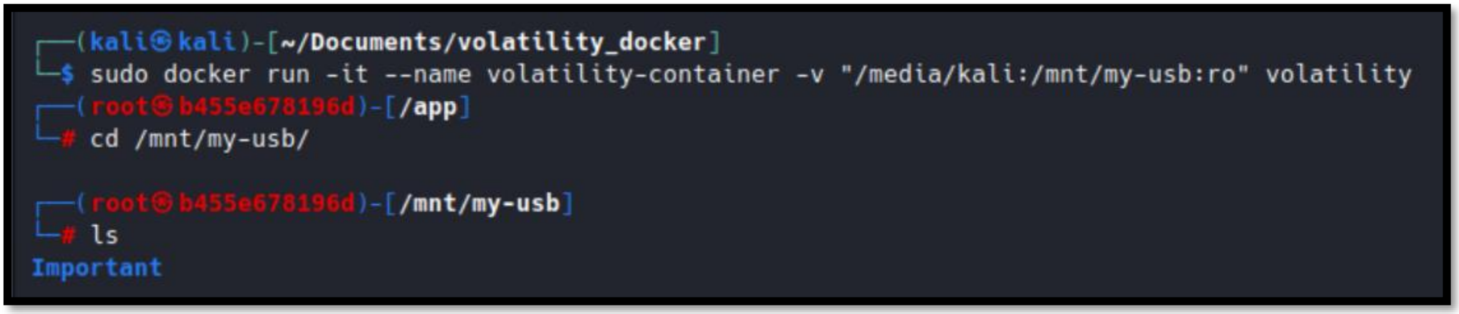
To make sure that the USB device is perfectly attached/accessible to the Virtual machine running the docker, check out these pre-requisites below.

- Install USB drivers. For VirtualBox, you can find the USB driver at “C:\Program Files\Oracle\VirtualBox\drivers\USB\filter\VBoxUSBMon.inf”.
- Install Extension Pack. VirtualBox users’ needs to install extension pack for their drivers.
- Make sure the correct USB controller and USB device filters are selected for the detection of the USB device to your virtual machine hosting the docker.

Once the USB is detected by the virtual machine, we should attach it to the docker container during the runtime, so that once we are in docker container, we can access the USB with ease.

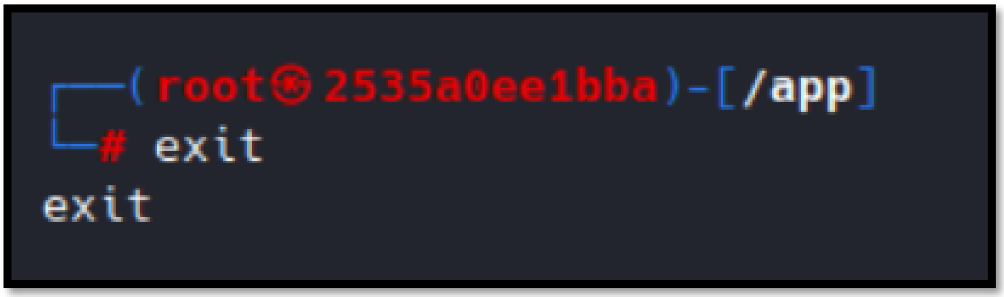
```
$ sudo docker run -it --name volatility-container -v "/media/kali:/mnt/my-usb:ro" volatility
```

Where, volatility-container is the name of container, /media/kali is the location of the USB connected to my virtual machine, /mnt/my-usb:ro is the location where I can find the attached USB in the docker container, and :ro stating giving them read-only access.



```
┌──(kali㉿kali)-[~/Documents/volatility_docker]
└─$ sudo docker run -it --name volatility-container -v "/media/kali:/mnt/my-usb:ro" volatility
┌──(root㉿b455e678196d)-[/app]
└─# cd /mnt/my-usb/

┌──(root㉿b455e678196d)-[/mnt/my-usb]
└─# ls
Important
```

Step 8: How to exit the running container:



```
┌──(root㉿2535a0ee1bba)-[/app]
└─# exit
exit
```