

Assignment-3

A] Create a docker image from a Dockerfile against which when container comes up it should:

Display the date in form of mm-dd-yy [ENTRYPOINT use case]

Display the date in form of pf mm-dd-yy [ENTRYPOINT use case]

Display the pwd to be ``/var/log`` [WORKDIR use case]

Display the pwd to be ``/var/log`` [CMD and WORKDIR use case]

Display output of `echo $NAME` to be CODEMIND [ENV use case]

Map a `source_code.txt` inside the container in a way it should sync the content inside the container at container run time. [VOLUME use case]

docker image name should be 'codemind_docker_assignment_image:v1'

docker container name should be 'codemind_docker_assignment_container'

In this Dockerfile, we start with the `alpine:latest` base image. We then use the `WORKDIR` command to set the current working directory to `/var/log`. Finally, we use the `ENTRYPOINT` command to specify the command to be executed when the container starts. In this case, we use a shell command to display the current date in the format `mm-dd-yy` using `$(date +%m-%d-%y)`, and then display the current working directory using `pwd`. When the container is started, it will display the date followed by the working directory in the specified format.

```
bbhushan@ubuntu:/var/log$ ls
alternatives.log  btcp          dmesg.1.gz  dpkg.log    hp          lastlog      syslog       ubuntu-advantage.log  vmware-network.2.log  vmware-vmsvc-root.2.log
apt              cups          dmesg.2.gz  faillog     installer   openvpn      syslog.1     ubuntu-advantage-timer.log  vmware-network.3.log  vmware-vmsvc-root.3.log
auth.log         dist-upgrade  dmesg.3.gz  fontconfig.log  journal    private      syslog.2.gz  unattended-upgrades     vmware-network.4.log  vmware-vmsvc-root.log
auth.log.1       dmesg         dmesg.4.gz  gdm3        kern.log    source_code.txt  syslog.3.gz  vmware          vmware-network.log     vmware-vntoolsd-root.log
bootstrap.log    dmesg.0       Dockerfile  gpu-manager.log  kern.log.1  speech-dispatcher  syslog.4.gz  vmware-network.1.log  vmware-vmsvc-root.1.log  wtmp
bbhushan@ubuntu:/var/log$ cat Dockerfile
# Base image
FROM ubuntu:latest

# Set the working directory
WORKDIR /var/log

# Set an environment variable
ENV NAME=CODEMIND

# Copy source_code.txt to the container
COPY source_code.txt /var/log/source_code.txt

# Set the entrypoint to display the current date in mm-dd-yy format
ENTRYPOINT ["date", "+%m-%d-%y"]

# Set the default command to display the working directory
CMD ["pwd"]

# Set the volume a mount point with the specified name and marks
RUN mkdir myvol
RUN cp /var/log/source_code.txt /var/log/myvol
VOLUME /myvol
bbhushan@ubuntu:/var/log$
```

To build the Docker image, save this file as `Dockerfile`, navigate to the directory containing the `Dockerfile`, and run the following command:

docker build -t codemind_docker_assignment_image:v1 .

```

bhushan@ubuntu:/var/log$ sudo docker build -t codemind_docker_assignment_image:v1 .
[sudo] password for bhushan:
Sending build context to Docker daemon 149.9MB
Step 1/9 : FROM ubuntu:latest
latest: Pulling from library/ubuntu
2ab09b027e7f: Pull complete
Digest: sha256:67211c14fa74f070d27cc59d69a7fa9aeff8e28ea118ef3babbc295a0428a6d21
Status: Downloaded newer image for ubuntu:latest
--> 08d22c0ceb15
Step 2/9 : WORKDIR /var/log
--> Running in 6e14530deab0
Removing intermediate container 6e14530deab0
--> c0842c98d420
Step 3/9 : ENV NAME=CODEMIND
--> Running in d987e4c736d8
Removing intermediate container d987e4c736d8
--> f59d37fa5048
Step 4/9 : COPY source_code.txt /var/log/source_code.txt
--> 3a37ac7a4e37
Step 5/9 : ENTRYPOINT ["date", "+%m-%d-%y"]
--> Running in d43428d3e2ed
Removing intermediate container d43428d3e2ed
--> d51d947bda7c
Step 6/9 : CMD ["pwd"]
--> Running in 0e46ddb1b3f0
Removing intermediate container 0e46ddb1b3f0
--> d31f0264ba98
Step 7/9 : RUN mkdir myvol
--> Running in 84b6f200e31b
Removing intermediate container 84b6f200e31b
--> 9a9547cae476
Step 8/9 : RUN cp /var/log/source_code.txt /var/log/myvol
--> Running in 7d149add5f65
Removing intermediate container 7d149add5f65
--> 134231019bb7
Step 9/9 : VOLUME /myvol
--> Running in 081d00d8b154
Removing intermediate container 081d00d8b154
--> 9abe65d3c31d
Successfully built 9abe65d3c31d
Successfully tagged codemind_docker_assignment_image:v1
bhushan@ubuntu:/var/log$

```

This command builds a Docker image with the tag **codemind_docker_assignment_image:v1** using the Dockerfile located in the current directory (.).

```

bhushan@ubuntu:/var/log$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
codemind_docker_assignment_image  v1         9abe65d3c31d  8 minutes ago  77.8MB
ubuntu              latest      08d22c0ceb15  2 weeks ago   77.8MB
bhushan@ubuntu:/var/log$ docker container ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
bhushan@ubuntu:/var/log$

```

To run a container from the image, use the following command:

```

docker run -it --name codemind_docker_assignment_container -v $(pwd)/source_code.txt:/var/log/source_code.txt codemind_docker_assignment_image:v1

```

```
Successfully built 7943ea59c5fb
Successfully tagged codemind_docker_assignment_image:v1
bhushan@ubuntu:/var/log$ docker run --name codemind_docker_assignment_container -v /path/to/source_code.txt:/var/log/source_code.txt codemind_docker_assignment_image:v1
03-26-23
```

This command runs a container with the name `codemind_docker_assignment_container`, maps the `source_code.txt` file from the current directory to `/var/log/source_code.txt` inside the container, and uses the image `codemind_docker_assignment_image:v1`. The `-it` flag runs the container in interactive mode, allowing you to see the output of the commands.