Writing the Dockerfile

```
#-----------------------------------
# Docker file Example
#-----------------------------------
# Use an alpine as base Operating System.
FROM alpine:3.18

#Get openjdk11
RUN  apk update \
  && apk upgrade \
  && apk add ca-certificates \
  && update-ca-certificates \
  && apk add --update coreutils && rm -rf /var/cache/apk/*   \
  && apk add --update openjdk11 tzdata curl unzip bash \
  && apk add --no-cache nss \
  && rm -rf /var/cache/apk/*

# Set the working directory in the container
WORKDIR /opt/cm

# Copy the golden cm directory to the container
ADD ./connection-manager /opt/cm

# ensure www-data user exists
RUN set -x ; \
    addgroup -g 82 -S www-data ; \
    adduser -u 82 -D -S -G www-data www-data && exit 0 ; exit 1

# set permissions appropriately
RUN rm -rf /opt/cm/logs/* && \
    rm -rf /opt/cm/uoj_trace.log && \
    touch /opt/cm/uoj_trace.log && \
    chown -R www-data:www-data /opt/cm/. && \
    chmod -R 700 /opt/cm/.

# Make port available to the world outside this container
#EXPOSE 7077

USER www-data

ENV JAVA_OPTS -DTYPE=MVCM

# Run when the container launches
#CMD ["sh", "-c", "java ${JAVA_OPTS} -jar cm-admin.jar"]
CMD ["sh", "-c", "exec java ${JAVA_OPTS} -jar cm.jar"]
#CMD ["java", "$JAVA_OPTS", "-jar", "cm.jar", "MVCM"]
```

```
#CMD ["nohup", "sh", "-c", "./start-admin.sh"]
#CMD ["sh"]

#-----------------------------------------------------------------------
FROM    : Base operation systme or pacakges
RUN     : Install necessary libraries required for the applicaiton.
WORKDIR   : The directory supposed to created inside the container to work it.
ADD         : Copy local files into container direcory in the process of creating the
container itself.
EXPOSE      : Make port available to the world outside this container, this can be a default
port,
        which can be override with the option -p while running the docker.
USER    : Setting the user which for the application
ENV     : SET THE env variable can be override with the -e option with run command.
CMD         : Command Run when the container launches
CMD VS ENTRYPOINT  : CMD is replacble but ENTRYPOINT is not replaceble.
ENTRYPOINT WITH CMD: CMD works as default agrument for the entrypoint command.
-------------------------------------------------------------------------------
CMD
------------------
FROM ubuntu
MAINTAINER sofija
RUN apt-get update
CMD ["echo", "Hello World"]

docker run test
>> Hello world
docker run test hostname
>> localhost

The command hostname overrides the CMD of container.

--------------
ENTRYPOINT
--------------
FROM ubuntu
MAINTAINER sofija
RUN apt-get update
ENTRYPOINT ["echo", "Hello World"]

docker run test
>> Hello world
docker run test hostname
>> Hello World localhost

The command hostname get appended with the ENTRYPOINT command it can't be override.

----------------------
ENTRYPOINT with CMD
----------------------
```

```
FROM ubuntu
MAINTAINER sofija
RUN apt-get update
ENTRYPOINT ["echo", "Hello"]
CMD ["World"]

docker run test
>> Hello world
docker run test RAVI
>> Hello RAVI

CMD work as a default argument for the ENTRYPOINT command.

------------------------
Create a Node js application with Docker
https://phoenixnap.com/kb/docker-node
------------------------
Dockerfile
FROM node:10-alpine
RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
WORKDIR /home/node/app
COPY package*.json ./
USER node
RUN npm install
COPY --chown=node:node . .
EXPOSE 8080
CMD [ "node", "app.js" ]
--------------------------------------l

    #After creating the Docekr command to create run and push the image to the Docekr hub.
    #1. docker build -t nodeapp .
    # docker build -t nodeapp:1.1 .
    #2. docker run -p  3000:80 --name <<name of the container you want to give>>  nodeapp:latest <Image with tag>
    #3. docekr tag nodeapp:latest tibcoranja/tibcorajan:nodeapp:1.1
    #4. docker login
    #5. docker push tibcoranja/tibcorajan:nodeapp:1.1
    #6. This image can be used in Kubernetes to create the Deployment or Pod for the nodeapp application.

#------------------
#Dokcer run command for the application which requires input at the time of start
#------------------
from random import randint
min_number = int(input('Please enter the min number: '))
max_number = int(input('Please enter the max number: '))
if (max_number < min_number):
  print('Invalid input - shutting down...')
else:
  rnd_number = randint(min_number, max_number)
  print(rnd_number)
```

```
#-----------------------------------------
ENV and ARG value in Docker
#-----------------------------------------
docker run -p  3000:80 --name nodeappcontainer nodeapp:latest
docker run -p  3000:8080 -e PORT=8080 --name nodeappcontainer nodeapp:latest
docker run -p  3000:8080 --env-file ./env-file --name nodeappcontainer nodeapp:latest
docker run -d --user root -p 7077:7077 -e "JAVA_OPTS=-DTYPE=MVCM -DFS_DIR=/data" -v
/home/mvis/demo:/data  --name=mvis_cm_admin cmadmin:1.3.0.00654
#The ARG gives flexibiity to change certain paramerter of Docerfile without changing it.
docker build -t feedback-node:dev --build-arg DEFAULT_PORT=8080 .
```

```
#-------------------------------------
#Docker container management
#-------------------------------------
1. List the running containers.
docker ps

2. List all the containers running or stopped.
doceker ps -a

3. Create a container without starting it.
[root@centos-8-master webapp]# d images
REPOSITORY                      TAG       IMAGE ID       CREATED        SIZE
mywebapp                        3.0       71597feaf172   2 hours ago    648MB
[root@centos-8-master webapp]# docker create mywebapp:3.0
5e593247d1da55597dec14cf2482d6e4a90a31acf66a4fb01dc89f53295259cc
[root@centos-8-master webapp]# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS    PORTS
NAMES
5e593247d1da   mywebapp:3.0   "python3 /opt/dockwo…"   5 seconds ago   Created
cool_poitras
[root@centos-8-master webapp]#

4. Start a container with the created one.
[root@centos-8-master webapp]# d container start 5e593247d1da
5e593247d1da
[root@centos-8-master webapp]# d ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS
PORTS      NAMES
5e593247d1da   mywebapp:3.0   "python3 /opt/dockwo…"   About a minute ago   Up 5 seconds
8085/tcp   cool_poitras
[root@centos-8-master webapp]#

5. Rename a container.
[root@centos-8-master webapp]# docker rename cool_poitras started-after-creat
[root@centos-8-master webapp]# d ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS
```

```
 PORTS      NAMES
5e593247d1da   mywebapp:3.0   "python3 /opt/dockwo…"   2 minutes ago   Up About a minute
8085/tcp   started-after-creat

6. docker diff <container id> : Show changes to files or directories on the filesystem.
[root@centos-8-master webapp]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED       STATUS       PORTS
 NAMES
5e593247d1da   mywebapp:3.0   "python3 /opt/dockwo…"   2 hours ago   Up 2 hours   8085/tcp
started-after-creat
[root@centos-8-master webapp]# docker exec -it 5e593247d1da  touch /tmp/test.log

[root@centos-8-master webapp]# docker diff 5e593247d1da
C /tmp
A /tmp/test.log

7. docker cp [file-path] CONTAINER:[path] : Copy a local file to a directory in a container.
[root@centos-8-master webapp]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED       STATUS       PORTS
 NAMES
5e593247d1da   mywebapp:3.0   "python3 /opt/dockwo…"   2 hours ago   Up 2 hours   8085/tcp
started-after-creat

[root@centos-8-master webapp]# docker cp /root/CKAD/CKAD-exercises-
main/RAVI_Practise/podDesing/nginx-primary.yaml 5e593247d1da:/tmp
Successfully copied 2.05kB to 5e593247d1da:/tmp

[root@centos-8-master webapp]# docker exec -it 5e593247d1da ls -l /tmp
total 8
-rwx------ 1 root root 836 Nov 13  2020 ks-script-DrRL8A
-rw-r--r-- 1 root root 486 Nov 19  2022 nginx-primary.yaml
-rw-r--r-- 1 root root   0 Jun 20 13:13 test.log
-rw------- 1 root root   0 Nov 13  2020 yum.log

8. Docker port <container id> : Provides port information.
[root@centos-8-master webapp]# d images
REPOSITORY                      TAG       IMAGE ID       CREATED        SIZE
mywebapp                        3.0       71597feaf172   5 hours ago    648MB

[root@centos-8-master webapp]# docker run -d -p 38085:8085 -v /data:/tmp/data --name flask-
webapp webapp:1.0
8d4fc4c6a021a787b37f11afe61f49056d974c7d8243b6358c091539af9615d6
[root@centos-8-master webapp]# d ps
CONTAINER ID   IMAGE        COMMAND                  CREATED         STATUS         PORTS
                                    NAMES
8d4fc4c6a021   webapp:1.0   "python3 /opt/dockwo…"   5 seconds ago   Up 3 seconds
0.0.0.0:38085->8085/tcp, :::38085->8085/tcp   flask-webapp
[root@centos-8-master webapp]# d port 8d4fc4c6a021
8085/tcp -> 0.0.0.0:38085
8085/tcp -> [::]:38085
```

```
#-------------------------------------
# Docker image management.
#-------------------------------------
docker build [dockerfile-path]                    : Create an image from a Dockerfile.
docker build .                                 : Build an image using the files from the current
path.
docker build -t [name]:[tag] [location]          : Create an image from a Dockerfile and tag
it.
docker build -f [file]                   : Specify a file to build from.
docker pull [image]                    : Pull an image from a registry.
docker push [image]                    : Push an image to a registry.
docker export [container id]               : Create a tar file from the running container.
docker import [url/file]                 : Create an image from a tarball.
docker commit [container] [new-image]          : Create an image from a running container.
docker tag [image] [image]:[tag]             : tag a image


Show all locally stored top
level images.
Show history for an image.
Remove an image.
Load an image from a tar
archive file.
Save an image to a tar
archive file.
Search Docker Hub for
images.
Remove unused images.

[root@centos-8-master webapp]# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS
            PORTS                                       NAMES
893cf381b95c   webapp:1.0     "python3 /opt/dockwo…"   55 minutes ago      Up 55 minutes
             0.0.0.0:8085->8085/tcp, :::8085->8085/tcp   flask-webapp

[root@centos-8-master webapp]# docker export 893cf381b95c -o mywebapp.tar
[root@centos-8-master webapp]# ls -lrt
-rw------- 1 root root 455197696 Jun 20 13:43 mywebapp.tar


[root@centos-8-master webapp]# docker import mywebapp.tar mywebapp:2.0
[root@centos-8-master webapp]# d images
REPOSITORY                        TAG       IMAGE ID       CREATED             SIZE
mywebapp                          2.0       3cb19b1a65d0   17 seconds ago      444MB

[root@centos-8-master webapp]# d save mywebapp -o save-webapp-from-image-mywebapp.tar
[root@centos-8-master webapp]# ls -lrt
-rw------- 1 root root 455197696 Jun 20 13:43 mywebapp.tar
-rw------- 1 root root 455205888 Jun 20 13:52 save-webapp-from-image-mywebapp.tar
```

```
[root@centos-8-master webapp]# d load -i save-webapp-from-image-mywebapp.tar
Loaded image: mywebapp:2.0

[root@centos-8-master webapp]# docker commit -m "Creating a new image from the running
conatiner" 893cf381b95c mywebapp:3.0
sha256:71597feaf1720ce3781226f9dc4106dd5ef72905344bc550d5b1ba5ca5e31664
[root@centos-8-master webapp]# docker images
REPOSITORY                      TAG       IMAGE ID       CREATED             SIZE
mywebapp                        3.0       71597feaf172   8 seconds ago       648MB
mywebapp                        2.0       3cb19b1a65d0   8 minutes ago       444MB

[root@centos-8-master webapp]# docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
5b5fe70539cd: Pull complete
1d40567696ba: Pull complete
d6cb3e372b06: Pull complete
fbc80fe62958: Pull complete
3da840f9e96e: Pull complete
Digest: sha256:f499227681dff576d6ae8c49550c57f11970b358ee720bb8557b9fa7daf3a06d
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
[root@centos-8-master webapp]# docker images
httpd                           latest    ad303d7f80f9   5 days ago       168MB

[root@centos-8-master webapp]# docker push tibcoranjan/tibcopublic:httpd
The push refers to repository [docker.io/tibcoranjan/tibcopublic]
e3d394052b85: Mounted from library/httpd
afb2764e050a: Mounted from library/httpd
084ceb61a963: Mounted from library/httpd
85322d06f709: Mounted from library/httpd
ac4d164fef90: Mounted from library/httpd
httpd: digest: sha256:760413d133979e7cc5ced07d8c323e7b68bb48b4e114f8898c2f2d6af82e2424 size:
1366
[root@centos-8-master webapp]# docker image ls tibcoranjan/tibcopublic
REPOSITORY                TAG       IMAGE ID       CREATED      SIZE
tibcoranjan/tibcopublic   httpd     ad303d7f80f9   5 days ago   168MB

Create Nginx Container
$ docker run -d -p 80:80 nginx

Displaying Running Container
$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
           PORTS                NAMES
df2caf9283e8        nginx               "nginx -g 'daemon of…"   35 seconds ago      Up 34
seconds       0.0.0.0:80->80/tc

$ docker export df2caf9283e8 > nginx.tar
```

```
$ docker import - mynginx < nginx.tar
sha256:aaaed50d250a671042e8dc383c6e05012e245f5eaf555d10c40be63f6028ee7b
```

If you would rather deal with images that you have already committed, you can use the load and save commands:

```
$ docker save -o mynginx1.tar nginx
$ ls -l
total 218756
-rw-------    1 root     root     112844800 Dec 18 02:53 mynginx1.tar
-rw-r--r--    1 root     root     111158784 Dec 18 02:50 nginx.tar
```

Now delete all images and containers running and try to run the below command to load Docker image into your system:

```
$ docker rmi mynginx
Untagged: mynginx:latest
Deleted: sha256:aaaed50d250a671042e8dc383c6e05012e245f5eaf555d10c40be63f6028ee7b
Deleted: sha256:41135ad184eaac0f5c4f46e4768555738303d30ab161a7431d28a5ccf1778a0f

$ docker rmi nginx
Untagged: mynginx:latest
Deleted: sha256:aaaed50d250a671042e8dc383c6e05012e245f5eaf555d10c40be63f6028ee7b
Deleted: sha256:41135ad184eaac0f5c4f46e4768555738303d30ab161a7431d28a5ccf1778a0f

$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE

$ docker load < mynginx1.tar
Loaded image: nginx:latest
[node1] (local) root@192.168.0.33 ~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx               latest              568c4670fa80        2 weeks ago         109MB
[node1] (local) root@192.168.0.33 ~
$
```

Cleaning up
To clean or prune unused (dangling) images

```
docker image prune
```

To remove all images which are not in use containers , add – a

```
docker image prune -a
```

To Prune your entire system

```
docker system prune
```

To kill all running containers

```
docker kill $(docker ps -q )

docker container prune
-----------------------------------------------------------------------
Run options for a container
-----------------------------------------------------------------------
docker run --rm <image id>   : Run a container and remove it after it stops.
docker run -it <image id>    : Run an interactive process e.g a shell in a container.
docker start/stop/restart/pause/unpause <container id>
```

```
-------------------------------------
docker network command
-------------------------------------
docker network ls                  : View available networks.
docker network rm [network]           : Remove a network.
docker network inspect  [network]       : Show information about a network.
docker network connect [network] [container]  : Connect a container to a network.
docker network disconnect [network] [container] : Disconnect a container from a network.
```