- Display the contents of the Dockerfile
  FROM nginx:latest

  COPY index.html /usr/share/nginx/html
  COPY linux.png /usr/share/nginx/html

  EXPOSE 80 443

  CMD ["nginx", "-g", "daemon off;"]

  ANS:FROM – This command helps user to specify the base image to use for the build. In this is case, it is nginx latest
  COPY- This command helps user to copy files/directories from index.html (source/host file system) to /usr/share/nginx/html (destination)
  COPY- copy files/directories from linux.png (source/host file system) to /usr/share/nginx/html (destination)
  EXPOSE – Expose specify the user to give network ports that container listen at runtime. In the above case container listens to 80, 433
  CMD – CMD is the key word that used for running a command when the container is started. In this we are asking to run ngnix container.

- What is the output of the Docker daemon executing each line in the Dockerfile? You should explain all steps of the layers.
  ANS: There are 5 steps that it has run in docker file. As mentioned above the definitions of above commands, it ran steps in docker files.
  1. Pulled and downloaded new latest image ngnix as we mentioned lastest
  2. Copied index.html file to /usr/share/nginx/html
  3. Copied linux.png file to /usr/share/nginx/html
  4. Exposed container to ports 80 for running.
  5. The last cmd ran the container and tagged it with our name given in dockerID from docker image command.

- How is this possible that you were able to modify the index.html and you never actually built a new image after modifying the running website? Explain your finding.
  ANS: We don't need to create a new image as we have linux_tweet_app1.0 which runs from docker local host system repo, looks for local repo, so we are acquiring the contents , index.html file to index-new.html in image linux_tweet_app1.0 which all of them are in local repo. It doesn't require to build a new image.

- Now that you have built the second image with an updated index.html file. What have you noticed about the time it took to build the image? Explain your finding.
  ANS: It was very quick this in terms of time build it. It is building image on same index.html file in local docker host repo instead of connecting to docker hub from linux_tweet_app1.0, copying files, that are mentioned in docker file, it took the 1.0 version as reference, went ahead created the copy of the image as all the layers were created.

- Have a close look at the output as you push the image. Explain in detail about your findings.
  ANS: When we pushed the first image into docker hub, for every row we write in dockerfile, it creates a layer, on top if it keeps the container for read write operations. So, when second image is pushed as it has all the layer that were created from image 1.0 version, and 2nd image has same files, so that is the reason for first image all the layers are pushed, for second image it said “Layer already exists”.