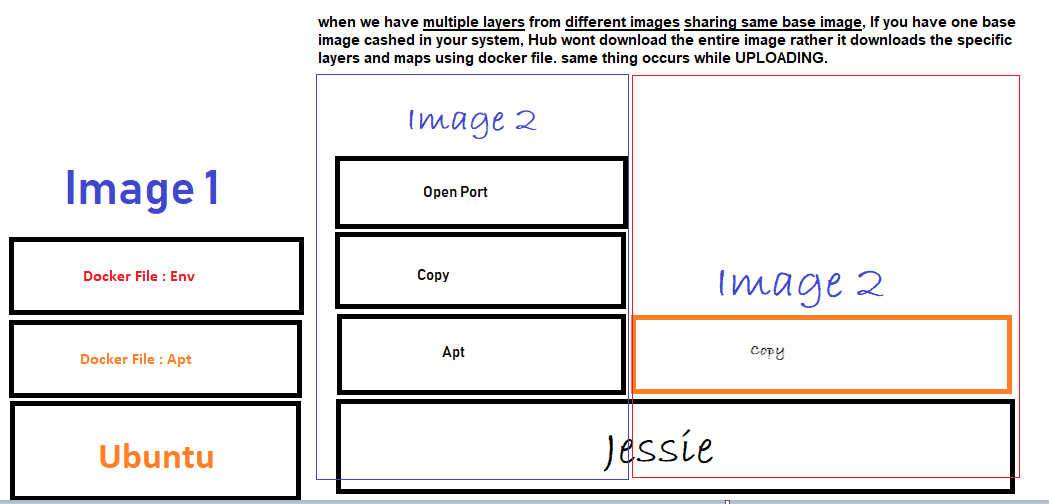
**Docker Image**

1. DOCKER REPOSITORY SHOULD BE FOLLOWED: <https://hub.docker.com/>
2. docker image history --human nginx **[Returns the change log for the image]**
3. Docker image visual Representation : **Each layer is uniquely identified & only stored once on a host**

****

1. [**Creating OWN image on existing image**]: Format [**docker image tag SOURCE\_IMAGE[:TAG] TARGET\_IMAGE[:TAG]** ]

docker image tag nginx roysoha/nginx:<ADD A Tag-Optional>

1. [**PUSH to Docker: Needs login] :** docker push roysoha/nginx-with-html
2. **Sample Docker File**

[D:\Docker\DockerFileDemo](file:///D:\Docker\DockerFileDemo)

1. **[Building Dockerfile]**

docker image build -t customnginx . (. Stands for build the dockerfile in this same directory)

1. **Building a docker image with Dockerfile**

**[**D:\Docker\DockerFileDemo\dockerfile-sample-2**]**

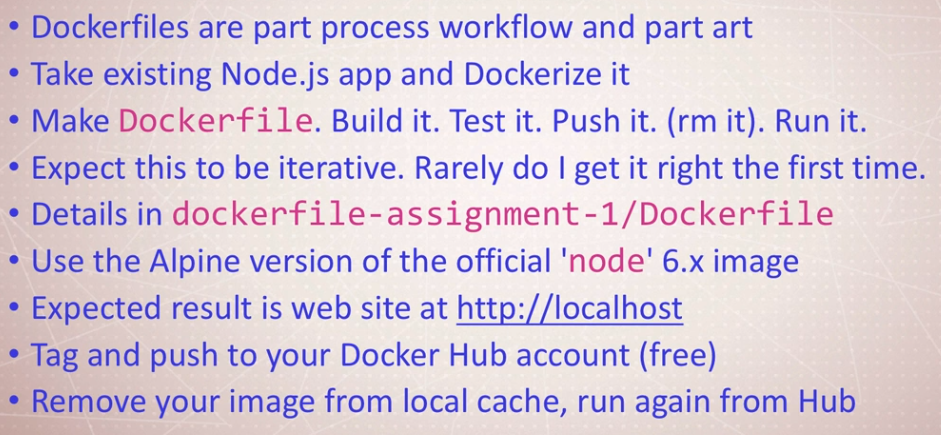
* docker image build -t nginx-with-html . **[build the image with HTML file]**
* docker container run -p 83:80 --rm nginx-with-html **[Run the container]**
* docker image ls **[verify the image]**
* check on localhost:83 **[run the image on browser to verify the container]**
* Push to Docker Repository
* docker image tag *nginx-with-html:latest* roysoha/nginx-with-html:first

{ *nginx-with-html:latest 🡺* Source Name}

{ roysoha/nginx-with-html:first 🡺 Target name}

* docker push roysoha/nginx-with-html

Dockerfile Challenge



**# Instructions from the app developer**

**# - you should use the 'node' official image, with the alpine 6.x branch**

**FROM node:alpine3.10**

**# - this app listens on port 3000, but the container should launch on port 80**

**# so it will respond to http://localhost:80 on your computer**

**EXPOSE 3000**

**# - then it should use alpine package manager to install tini: 'apk add --update tini'**

**RUN apk add --update tini**

**# - then it should create directory /usr/src/app for app files with 'mkdir -p /usr/src/app'**

**RUN mkdir -p /usr/src/app**

**# - Node uses a "package manager", so it needs to copy in package.json file**

**WORKDIR /usr/src/app**

**COPY package.json package.json**

**# - then it needs to run 'npm install' to install dependencies from that file**

**# - to keep it clean and small, run 'npm cache clean --force' after above**

**RUN npm install**

**# - RUN npm audit fix**

**# - then it needs to copy in all files from host directort to current directory**

**COPY . .**

**# - then it needs to start container with command '/sbin/tini -- node ./bin/www'**

**CMD [ "/sbin/tini", "--", "node", "./bin/www" ]**

**# - in the end you should be using FROM, RUN, WORKDIR, COPY, EXPOSE, and CMD commands**

1. Build Docker file : docker build -t testnode .
2. Run : docker container run -p 91:3000 testnode
3. Push on dockerhub : docker push roysoha/testing-node
4. Kill containers : docker container kill -f 49b79ce10c0a 248a7dad5966 f8fd8b0dfbde 196c935f5e33 1e789c4415c0
5. Listing images : docker image ls -a
6. Remove local image: docker image rm roysoha/testing-node
7. Pulling it from docker hub : docker container run -p 91:3000 roysoha/testing-node