DockerFile

# Instructions

Common components and instructions in docker file. Each instruction by default creates a different layer in docker image.

FROM: Where to pull base image from for creating our image

CMD: Sets commands to be executed at the start when running the newly created image. This is also used in combination with ENTRYPOINT to provide default arguments to commands mentioned in entrypoint.

RUN: Executes commands what we specify which are required to build the image

EXPOSE: Open networking ports from image

VOLUME: Sets disk share for image/container (when its running)

COPY: Copy file into image from local host disk

LABEL: Provide metadata to image in key-value format mostly author details are mentioned here. Previously MAINTAINER used for this purpose but its old way now.

ENV : sets environment variable in container when its running

ENTRYPOINT : sets what instructions to run when container starts. If any arguments required for the instructions mentioned here then CMD section is used to pass the arguments. It is used at the end of docker file followed by CMD.

Ex. when docker container from this image starts it will ping the google.com

FROM debian:latest

LABEL maintainer="Ashpak Mulani"

RUN apt update && \

    apt install -y traceroute && \

    apt install -y curl && \

    apt clean

ENTRYPOINT ["curl"]

CMD ["www.google.com"]

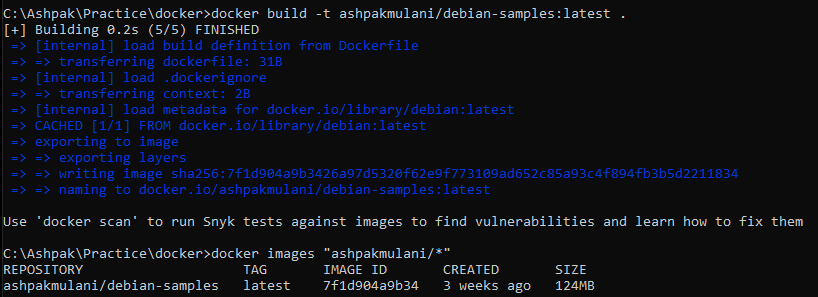
# Build image

Docker build command is used to build the images

Docker build -t <<namespace>>/<<repository>>:<<tag>> <<dockerfilepath>>

Ex. docker build -t ashpakmulani/dabien-sample:latest .

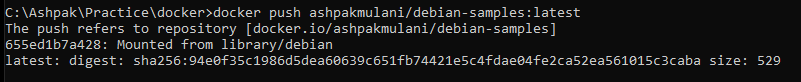
As shown below docker creates a mentioned image.



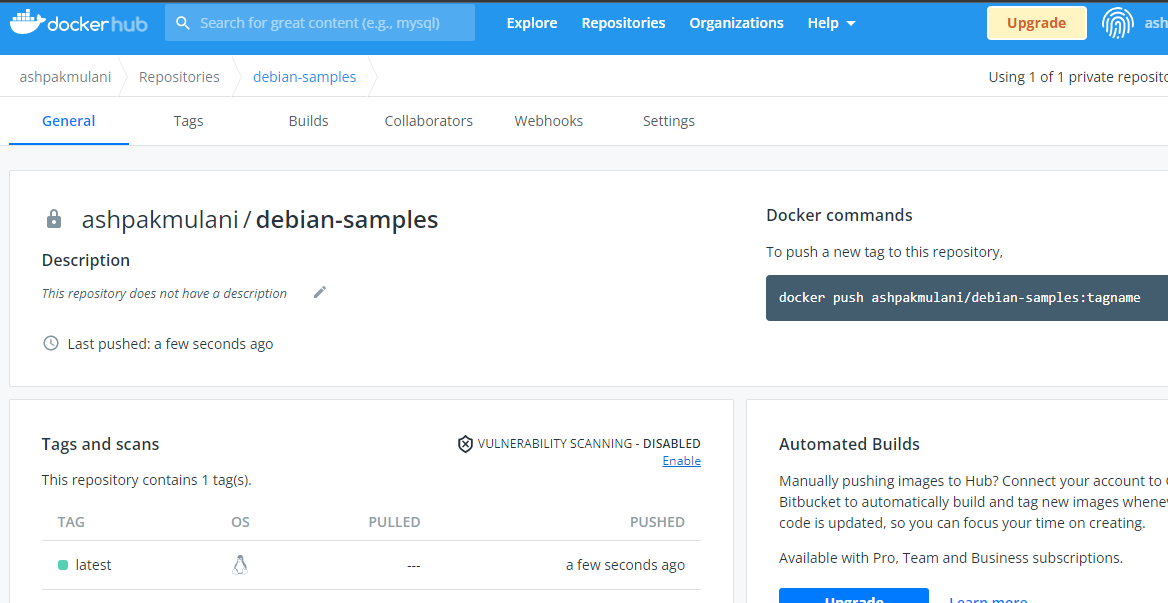
We can push the created image to docker hub in our user profile. We need to be logged into current console using ‘docker login’ command first.

Docker push <<namespace>>/<<repository>>:<<Tag>>

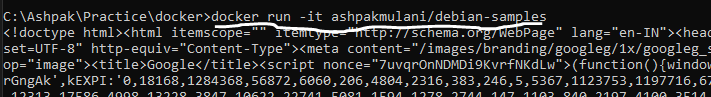
Ex. Docker push ashpakmulani/debian-samples:latest



Before running command, we need to make sure ‘Debian-samples’ repository is already created in our own dockerhub account. ‘ashpakmulani’ is namespace and it is usually user name in docker account. We can see our pushed image in dockerhub repo with latest tag.



If we run our created image it is going to execute **curl** [**www.google.com**](http://www.google.com) command inside container because of added ENTRYPOINT and CMD instructions.



CURL is not present in debian image by default so we installed required package using RUN instructions. In RUN instructions if we use ‘&&’ then it concatenates the multiple instructions in single instruction and docker does not create separate individual layer for multiple instructions, it combines them in single instruction.

## Override entrypoint

Sometimes when we have image with ENTRYPOINT instruction and we don’t want to run those instructions while starting container then we can override them using –entrypoint switch. Below command will use newly provided entry point and open bash when container runs.

Docker run –entrypoint /bin/shell -it <<imagename>>