Docker from scratch

Saturday, April 18, 2020 1:17 PM

docker pull $USER_ID/$IMAGE_NAME:$TAG.
image name is the name of the project,
Tag is version name, default is latest.
eg. docker pull ubuntu, ubuntu is image name, there
is no user name, tag is latest, i.e. ubuntu:latest.
docker pull pytorch/pytorch:1.4-cuda10.1-cudnn7-devel
organization name is pytorch, project name is pytorch,
tag name is 1.4-cuda10.1-cudnn7-devel.
We can see the images we pull using
docker images.

docker run -it --name=base_ubuntu ubuntu:16.04 /bin/bash
-it: enter interactive mode
-name: container's name, we use base_ubuntu.
ubuntu:16.04 is image name. and tag name.
/bin/bash: enter command line.

if we use ctrl+p+q to exit container, the container is
still running in the background, and the modifications are kept.
We can also use ctrl+d or exit, but the container will be
stopped and all changes are lost.
We can use docker ps -a to check all containers.
to remove the container, use:
docker stop $CONTAINER_ID
docker rm $CONTAINER_ID
we can use the id or name for container.

docker commit -m="$MESSAGE" -a="AUTHOR" $CONTAINER_ID $USER_ID/$IMAGE_NAME:$TAG.
(same with github -m, -a; container id; user id; image name; tag)
docker tag $OLD_NAME $NEW_NAME
this will change the version/tag name

docker push username/my_ubuntu:latest
this will push the docker image to dockerhub.

example: FADNet:
docker run --runtime=nvidia -it -v ${HOST_DATA}:/data --ipc=host
--name=fadnet paopaorobot/fadnet:v1 /bin/bash
cd /root/FADNet
dnn=fadnet ./train.sh.

Dockerfile
FROM:
this sets the base image, eg. in ros/kinetic-ros-core, it's
FROM ubuntu:xenial

RUN:
RUN apt-get update && apt-get install -q -y \
dirmngr \
gnupg2 \
&& rm -rf /var/lib/apt/lists/*
RUN apt-key adv ...
RUN echo ...

ENV:
set environment variable
ENV ROS_DISTRO kinetic (used here)
RUN rosdep init && \
rosdep update --rosdistro $ROS_DISTRO

COPY
COPY ./ros_entrypoint.sh /
copy the local file named ros_entrypoint.sh to image's root dir.

ENTRYPOINT
ENTRYPOINT ["/ros_entrypoint.sh"]
This will set the script to run after starting the container

CMD
CMD ["bash"]
when create container, a bash will start the terminal.

docker build -t IMG_NAME DOCKERFILE_PATH
this will create the image based on dockerfile.

example: LARVIO
docker pull paopaorobot/larvio
or we can create from dockerfile
git clone \
https...larvio.git
cd ./docker_larvio
docker build -t paopaorobot/larvio .
download VNC, EuRoC dataset.
Put V1_02's asl and bag at path/V1_02_medium, then
docker run -itd -v \
Path/V1_02_medium:/root/Dataset/V1_02 \
-p 5900:5900 paopaorobot/larvio
(we mount the dataset here in container)
open VNC, enter 127.0.0.1:5900 to connect.
open a terminal, enter
cd /root/LARVIO/build
./larvio \
/root/Dataset/V1_02/mav0/imu0/data.csv \
/root/Dataset/V1_02/mav0/cam0/data.csv \
/root/Dataset/V1_02/mav0/cam0/data \
../config/euroc.yaml
this will show the Pangolin display.
(run this in local, not in VNC)
docker exec -itd \
`docker ps | grep paopaorobot/larvio | awk '{print $1}'` \
/bin/bash -c \
'cd /root/LARVIO/ros_wrapper && \
. devel/setup.bash && \
roslaunch larvio larvio_euroc.launch'
this will run the ROS nodelet.
cd /root/LARVIO/ros_wrapper (run this in VNC)
. devel/setup.bash
roslaunch larvio larvio_rviz.launch
back to local:
docker exec -it \
`docker ps | grep paopaorobot/larvio | awk '{print $1}'` \
/bin/bash -c \
'. /opt/ros/melodic/setup.bash && \
rosbag play \
/root/Dataset/V1_02/V1_02_medium.bag'