# Docker for SLAM

Saturday, April 18, 2020 9:48 AM

image ≈ class
container ≈ class instance
Install Docker for ubuntu.
Go to Dockerhub, find paopaorobot, docker pull the relvevant one.
Can use 16.04, 18.04... and kinetic, melodic...
dock image ls: shows image in local.
dock run -it (image name): creates a container for the image.
in the container, use cat /etc/issue to check the version.
exit: exit from container.
dock run -it -p 5900:5900 (image name): use vnc to visualize,
5900 is the port name.
open vnc viewer, enter :5900, then can see the desktop.
if use the ros version:
docker run -it -p 5900:5900 paopaorobot/ros-vnc.
then we can run roscore and rviz in vnc.
docker run -it -p 5900:5900 -e RESOLUTION=1920x1200 paopaorobot/ubuntu-xfce-vnc.
docker run -it -P 2222:22 paopaorobot/ubuntu-openbox-vnc : this enables the ssh.
open another terminal, use
ssh -o 'UserKnownHostsFile=/dev/null' root@localhost -p 2222
2222 is port name, this means do not save key.
docker run -it -p 2222:22 -e SSHPW=abc123 paopaorobot/ubuntu-openbox-vnc.
SSHPW sets the password for SSH
docker run -it -p 5900:5900 -p 2222:22 -v /···/test:/home/test test:paopao
then we can use both ssh and vnc.

Demo:
docker pull paopaorobot/dvo
docker run -it -p 5900:5900 -v local data path: container data path paopaorobot/dvo
we use -v to mount data in container
in vnc, use
roslaunch dvo_benchmark benchmark.launch dataset:=/root/data
dockerfile:
FROM paopaorobot/ros-vnc:fuerte base on which image
COPY . /root/fuerte_workspace/dvo/
RUN rosws init ~/fuerte_workspace /opt/ros/fuerte \
    && cd ~/fuerte_workspace \
    && rosws set -y ~/fuerte_workspace/dvo \
    && /bin/bash -l -c 'source ~/fuerte_workspace/setup.bash && \
        cd ~/fuerte_workspace && \
        rosmake dvo_core dvo_ros dvo_benchmark' newer ros uses catkin make
RUN /bin/bash -c 'echo "source ~/fuerte_workspace/setup.bash" >> ~/.bashrc' source setup
Alternatively, can create a container, eg paopaorobot/ros-vnc:fuerte, and setup the
dvo, then save the container as an image.

How to use Docker for development:
IDE is VS code.
1. connect to Docker:
use remote-containers plugin.
docker run -it paopaorobot/ubuntu-xfce-vnc:latest
    -v path to code: code in container
we need to mount the code to the container
In VS code, use remote-containers: attach to running container.
then select the container currently running.
To open the code, open the path in container in VS.
Since we have the environment, we can have the auto completion now.
2. use SSH.
In VS, use remote-ssh.
add -p 2222:22 to docker run.
in VS, use remote-ssh, container to host, add new SH host
then enter:
ssh -o 'UserKnownHostFile=/dev/null' root@localhost -p 2222

How to create the image:
Dockerfile:
FROM ubuntu:latest base on a ubuntu official image
ENV HOME /root
ENV DEBIAN_FRONTEND noninteractive
RUN apt-get update \
    && apt-get install -y supervisor \ } for ssh
       openssh-server vim-tiny \
       openbox \ openbox
       x11vnc xvfb \ vnc, xvfb is virtual display
       firefox \
       pwgen \ password generation for ssh.
    && apt-get autoclean \ } delete extra stuff
    && apt-get autoremove \ } to decrease the docker size.
    && rm -rf /var/lib/apt/lists/* also the fewer the command like
                                   FROM, ENV, ADD... the better
WORKDIR /root
ADD startup.sh ./ } copy these to root
ADD supervisord.conf ./
EXPOSE 5900 } open ports
EXPOSE 22
ENTRYPOINT ["./startup.sh"] exe. this script when start

startup.sh:
mkdir -p /var/run/sshd
sed -i 's/PermitRootLogin/c PermitRootLogin yes' /etc/ssh/sshd_config allow root login
if [ ! $SSHPW ]; then
  SSHPW=`pwgen -c -n -1 12`
fi } generate ssh password.
echo "root:$SSHPW" | chpasswd
echo "ssh login password: $SSHPW"
if [ -n "$RESOLUTION" ]; then
  sed -i 's/1024x768/$RESOLUTION/' /root/supervisord.conf } set resolution.
fi
/usr/bin/supervisord -c /root/supervisord.conf start vnc, display, etc.
/bin/bash. start shell
supervisor can manage several processes. edit the supervisord.conf to set.

If we want to use Cuda, we need to use nvidia-docker

-p 2222:22 sets the port. manually we can use -P to auto-assign port.
then use
docker port (container id) 22 (or 5900)