Docker Container Set-Up and Run on AWS

Fan Wang

2020-09-13

Contents

1 Docker Setup

Go to the **RMD**, **PDF**, or **HTML** version of this file. Go back to Python Code Examples Repository (bookdown site) or the pyfan Package (API).

1.1 Install Docker on AWS

Installation Instructions

1. Putty
2. access to .pem key
3. conda aws environment below

For Amazon Linux 2:

```
# SSH into EC2 Instance
ssh ec2-user@ec2-52-23-218-117.compute-1.amazonaws.com
# Update
sudo yum update -y
# install
sudo amazon-linux-extras install docker -y
# start service
sudo service docker start
sudo service docker status
# execute docker commands without sudo
sudo usermod -a -G docker ec2-user
# log out and in reboot does not change public address
sudo reboot
# if docker info does not work, docker start again
docker info
```

1.2 Create a Dockerfile and build it

Create a Dockerfile and build it. Building a dockerfile generates a docker image:

```
# docker folder
mkdir ~/docker
# cd into docker folder
cd ~/docker
# create a Dockerfile in the docker folder
# copy the Example Dockerfile below to the Dockerfile
vim Dockerfile
# in the docker directory build the docker file
docker build -t hello-world .
```

Example Dockerfile:

```
FROM ubuntu:12.04

# Install dependencies
RUN apt-get update -y
RUN apt-get install -y apache2

# Install apache and write hello world message
RUN echo "Hello World!" > /var/www/index.html

# Configure apache
RUN a2enmod rewrite
RUN chown -R www-data:www-data /var/www
ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2

EXPOSE 80

CMD ["/usr/sbin/apache2", "-D",  "FOREGROUND"]
```

1.3 Run, Enter and Exit

docker build generates a docker image, we start the docker container, run using the image created.

```
# list docker images available to run
docker images
```

```
These could be some images that are shown after running *docker images*:

REPOSITORY                                                TAG                 IMAGE ID            C
XXXX7367XXXX.dkr.ecr.us-east-1.amazonaws.com/fanconda     latest              5d1a0df0796e        8 da
XXXX7367XXXX.dkr.ecr.us-east-1.amazonaws.com/fanconda2020 latest              2db5e859d70c        2 we
fanconda2020                                              latest              2db5e859d70c        2 we
fanconda5                                                 latest              fa55672e7753        2 we
fanconda3                                                 latest              2083f1124465        2 we
```

Run the image and enter into it (use an image name) and run commands with programs, upon exit, container is stopped. Entering back into the container, the data was generated before now is no longer there, it is a new container based on the same image.

```
# will be inside now the conda image (base)
docker run -t -i fanconda /bin/bash
# can run programs inside here that have been loaded into the image
python /fanProg/invoke/run.py -A ng_s_t=kap_m0_nld_m_simu=2=3 -B b -C 20180814_beta -D esti_param.beta
# review generated outputs inside docker, results are stored by the run.py program and associated file,
cd /data
ls
# To exit the currently running docker
exit
# show docker container exited
docker ps -a
```

Root directory in conda docker: > fanProg bin boot data dev etc home lib lib64 media mnt opt proc pyfan root run sbin srv sys tmp usr var

Run the image with program without entering into it.

```
docker run fancondajmp python /ThaiJMP/invoke/run.py -A ng_s_t=kap_m0_nld_m_simu=2=3 -B b -C 20180814_b
```

- Docker container will automatically stop after "docker run -d"

1.4 Status and Cleaning

1.4.1 Docker file and Git Repo

To have docker file access a git repo without exposing git repo password. Generate a private token, and access as below. See stackoverflow-23391839.

```
RUN git clone https://b123451234dfc025a836927PRIVATETOKEND1239@github.com/FanWangEcon/ThaiJMP.git /Thai
```

1.4.2 Docker Status, Space and Clean

First, start service:

```
# start docker
sudo service docker start
# see status
sudo service docker status
```

Second, list all docker related space usages, containers and images:

```
# check disk usage
docker system df
```

Third, clean containers

```
# see docker containers
docker container ls -a
# Remove all stopped docker containers
docker rm $(docker ps -a -q)
```

Fourth, clean images

```
# list all images
docker images
docker images --all
# Clean all images not referenced by a container
docker image prune
```