Docker Containers

Docker

- a system for building and running containers
- containers look a little like virtual machines, virtual environments

Very useful for work in DL/ML:

- create complex software installs independent of host (including Windows and MacOS)
- solve problems with driver/version incompatibilites for CUDA
- assure reproducibility
- basis for reliable distributed computation using Kubernetes

Docker Implementation

- containers actually share a kernel
- sophisticated version of `chroot`
- just manipulate name spaces
- provide isolation and (theoretically) security (BSD jails)
- multiple namespaces in Linux
  - file system
  - users
  - IPC
  - networking

Creating Docker Containers

- Docker containers are created with the `docker build` command
- this packages software into a container
- it's a cross between "linking an executable" and "installing a machine"

In [2]:

```
cat <<"EOF" |
FROM nvcr.io/nvidia/pytorch:19.11-py3
ENV DEBIAN_FRONTEND noninteractive
RUN apt-get update
RUN apt-get install -qqy daemon expect tmux dnsutils iputils-ping graphicsmagick
RUN python3 -m pip install git+git://github.com/tmbdev/webdataset
RUN python3 -m pip install git+git://github.com/tmbdev/torchmore
RUN python3 -m pip install git+git://github.com/tmbdev/tarproc
RUN python3 -m pip install git+git://github.com/NVlabs/tensorcom
RUN sed -i '3,29d;34,35s/.*/true/;43,44s/.*/true/;72,74s/.*/true/;98,101s/.*/true/;s/echo *$/true/'
/usr/local/bin/nvidia_entrypoint.sh
EOF
docker build -t myimage - > _log
tail _log
```

```
 ---> Using cache
 ---> e821490671e2
Step 8/9 : RUN python3 -m pip install git+git://github.com/NVlabs/tensorcom
 ---> Using cache
 ---> e4ec4e7ba81a
Step 9/9 : RUN sed -i '3,29d;34,35s/.*/true/;43,44s/.*/true/;72,74s/.*/true/;98,101s/.*/true/;s/echo
*$/true/' /usr/local/bin/nvidia_entrypoint.sh
 ---> Using cache
 ---> c2e40f3e6618
Successfully built c2e40f3e6618
Successfully tagged myimage:latest
```

Docker Containers Consist of Layers

In [3]:

```
docker history myimage | sed 10q
```

```
IMAGE               CREATED             CREATED BY                                      SIZE
COMMENT
c2e40f3e6618        5 hours ago         /bin/sh -c sed -i '3,29d;34,35s/.*/true/;43,…   68.1kB
e4ec4e7ba81a        5 hours ago         /bin/sh -c python3 -m pip install git+git://…   124kB
e821490671e2        6 hours ago         /bin/sh -c python3 -m pip install git+git://…   140kB
6eba85d1740a        6 hours ago         /bin/sh -c python3 -m pip install git+git://…   136kB
1c957b3afb1c        6 hours ago         /bin/sh -c python3 -m pip install git+git://…   9.95MB
43c2cc98b0c7        6 hours ago         /bin/sh -c apt-get install -qqy daemon expec…   114MB
01148ad8e25a        6 hours ago         /bin/sh -c apt-get update                       27.7MB
66dca9e4af22        6 hours ago         /bin/sh -c #(nop)  ENV DEBIAN_FRONTEND=nonin…   0B
c97490b12436        3 weeks ago         |2 NVIDIA_BUILD_REF=b16656a96f38aa26442503e8…   0B
```

Running Docker Containers

You run Docker containers with the `docker run` command. To get GPU support, you need to use the `nvidia` runtime.

Docker containers generally behave mostly like executables, scripts, and/or commands.

In [4]:

```
docker run --runtime=nvidia myimage nvidia-smi -L
```

```
GPU 0: TITAN X (Pascal) (UUID: GPU-a964bb9a-cb1a-5036-e1d8-1217c1faa8e7)
GPU 1: TITAN X (Pascal) (UUID: GPU-a16b9686-b668-e8d5-ff5f-f85aea86d034)
```

Interactive Usage

You can run Docker containers interactively as well.

In [5]:

```
# INTERACTIVE USAGE:
# docker run --runtime=nvidia -t -i myimage
# docker ps
# docker exec ... -t -i /bin/bash
```

Environment

By default, code inside a container executes as root, in a private namespace and directory only visible within the container.

In [6]:

```
docker run myimage bash -c 'id; pwd; ls'
```

```
uid=0(root) gid=0(root) groups=0(root)
/workspace
README.md
docker-examples
examples
tutorials
```

Sharing with Host

Most containers need to share something with their host. Here we share:

- networking namespace
- IPC namespace (shared memory etc.)
- the current directory
- the user and group ids and the corresponding passwd/group files

In [10]:

```
docker run -t -i \
    --network host \
    --ipc host \
    -v `pwd`:`pwd` \
    -w `pwd` \
    -v /etc/passwd:/etc/passwd \
    -v /etc/group:/etc/group \
    -u `id -u`:`id -g` \
    myimage \
    bash -c 'id; pwd; ls|sed 3q'
```

```
uid=1000(tmb) gid=1000(tmb) groups=1000(tmb)
/home/tmb/exp/bigdata19
000-data-prep.ipynb
010-introduction.md
020-image-classifier.md
```

Convenience Script

In [11]:

```
cat > myimage <<'EOF'
#!/bin/bash
exec docker run -t -i \
    --runtime=nvidia \
    --network host \
    --ipc host \
    -v /etc/passwd:/etc/passwd \
    -v /etc/group:/etc/group \
    -v `pwd`:`pwd` -w `pwd` -u `id -u`:`id -g` myimage "$@"
EOF
chmod 755 myimage
```

Using the Convenience Script

In [12]:

```
./myimage bash -c 'id; pwd; ls|sed 3q'
```

```
uid=1000(tmb) gid=1000(tmb) groups=1000(tmb)
/home/tmb/exp/bigdata19
000-data-prep.ipynb
010-introduction.md
020-image-classifier.md
```

Training with the Convenience Script

This uses the `python3` executable from within the container to run the script `training.py` in the current directory.

In [13]:

```
./myimage python3 training.py -m resnet18
```

```
/opt/conda/lib/python3.6/site-packages/torchvision/io/_video_opt.py:17: UserWarning: video reader ba
sed on ffmpeg c++ ops not available
  warnings.warn("video reader based on ffmpeg c++ ops not available")
Tue Dec 10 04:17:06 UTC 2019; bragi; tmb; /home/tmb/exp/bigdata19; GPU 0: TITAN X (Pascal) (UUID: GP
U-a964bb9a-cb1a-5036-e1d8-1217c1faa8e7); GPU 1: TITAN X (Pascal) (UUID: GPU-a16b9686-b668-e8d5-ff5f-
f85aea86d034);
creating resnet18
       0 bs   128 per sample loss 5.49e-02 loading 1.16e-02 training 3.51e-02
    2560 bs   128 per sample loss 5.51e-02 loading 3.60e-03 training 4.81e-03
    5248 bs   128 per sample loss 5.48e-02 loading 2.47e-03 training 1.07e-03
    7552 bs   128 per sample loss 5.47e-02 loading 2.85e-03 training 6.92e-04
    9856 bs   128 per sample loss 5.47e-02 loading 2.68e-03 training 6.21e-04
```

Running Jupyter

```
# change to home directory so that Jupyter picks up config
cd
# generate a configuration if you don't have one already
jupyter notebook --generate-config

# set a password so you can connect easily
jupyter notebook password

# run the actual server
./myimage jupyter lab
```