Cornell University
Computer Systems Laboratory

Developing In Docker

Mark Buckler

September 8, 2017

Cornell University
Computer Systems Laboratory

Motivation

1. Modern code is heavily dependent on packages/libraries
2. Installing dependencies can be time consuming, especially if installation instructions are incomplete or wrong
3. Multiple dependency versions can be cumbersome
4. Bit rot can break make environments

Summary:

Huge headache for people reproducing research as well as general users and developers

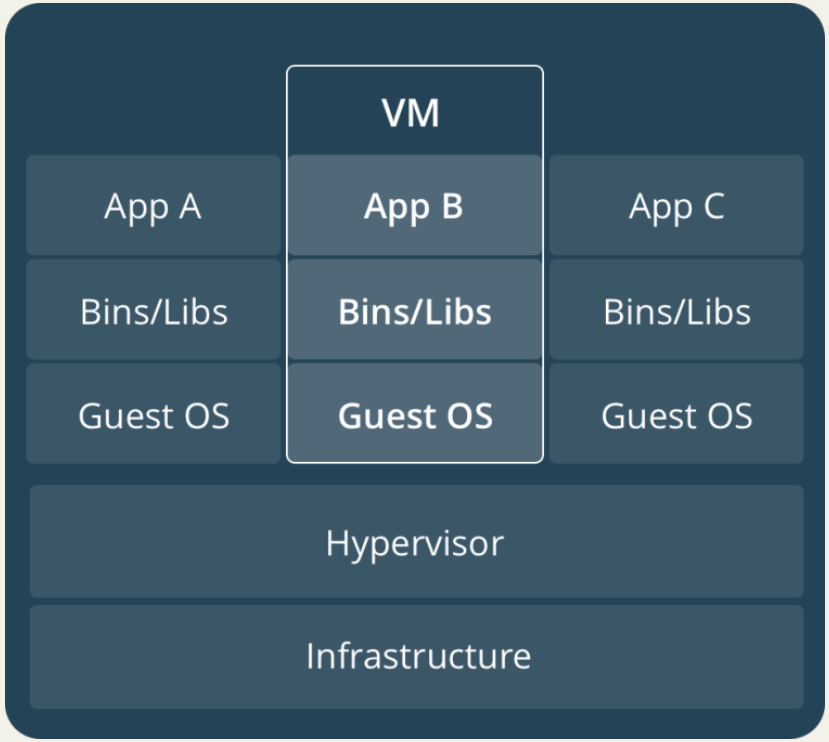
The Solution = Docker

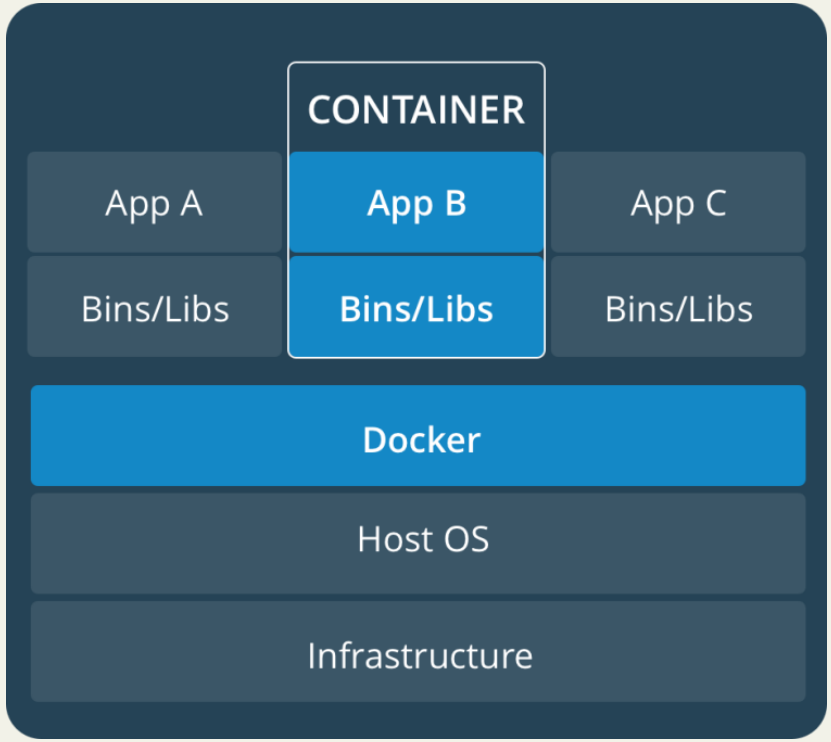
1. Users/co-developers can run your code without needing to install dependencies on their machine

2. Users/co-developers can maintain multiple dependency versions on a server without conflicts

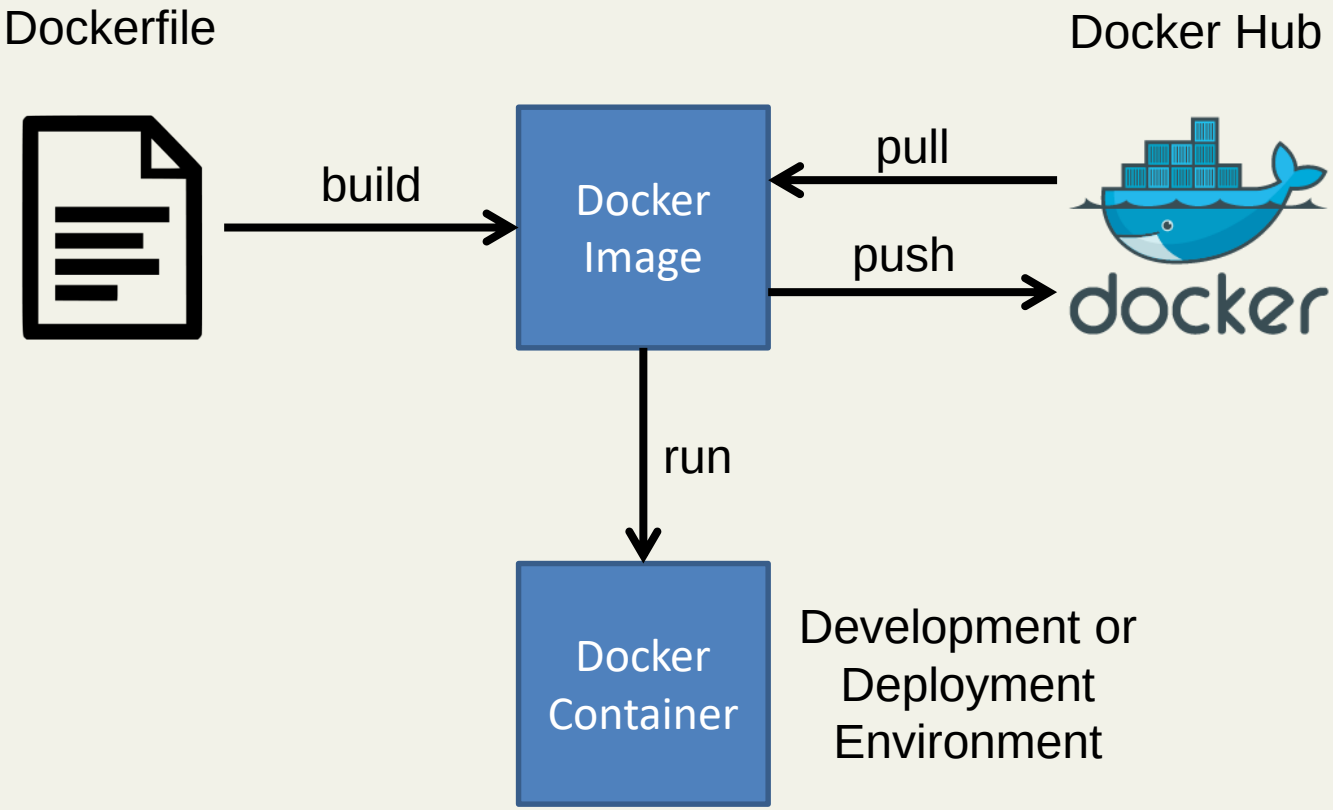
3. Environments can be "pulled" as simply as cloning from a GitHub repository, and therefore can easily be moved between multiple machines

What is Docker?





How Do I Use It?



Writing a Dockerfile

```
# Start with an existing Docker image to give yourself a head start
FROM ubuntu:14.04

# Change to a given directory
WORKDIR /root

# Run some shell commands to install dependencies
RUN apt-get update && apt-get upgrade && apt-get install -y \
    build-essential \
    git \
    etc...

# Clone and build code not available through your package manager
RUN git clone https://github.com/halide/Halide.git
WORKDIR /root/Halide
RUN make -j8

# Set any necessary environment variables
ENV LD_LIBRARY_PATH /root/Halide/bin:$LD_LIBRARY_PATH

# Lastly, set your workdir to be your default when starting the image
WORKDIR /root
```

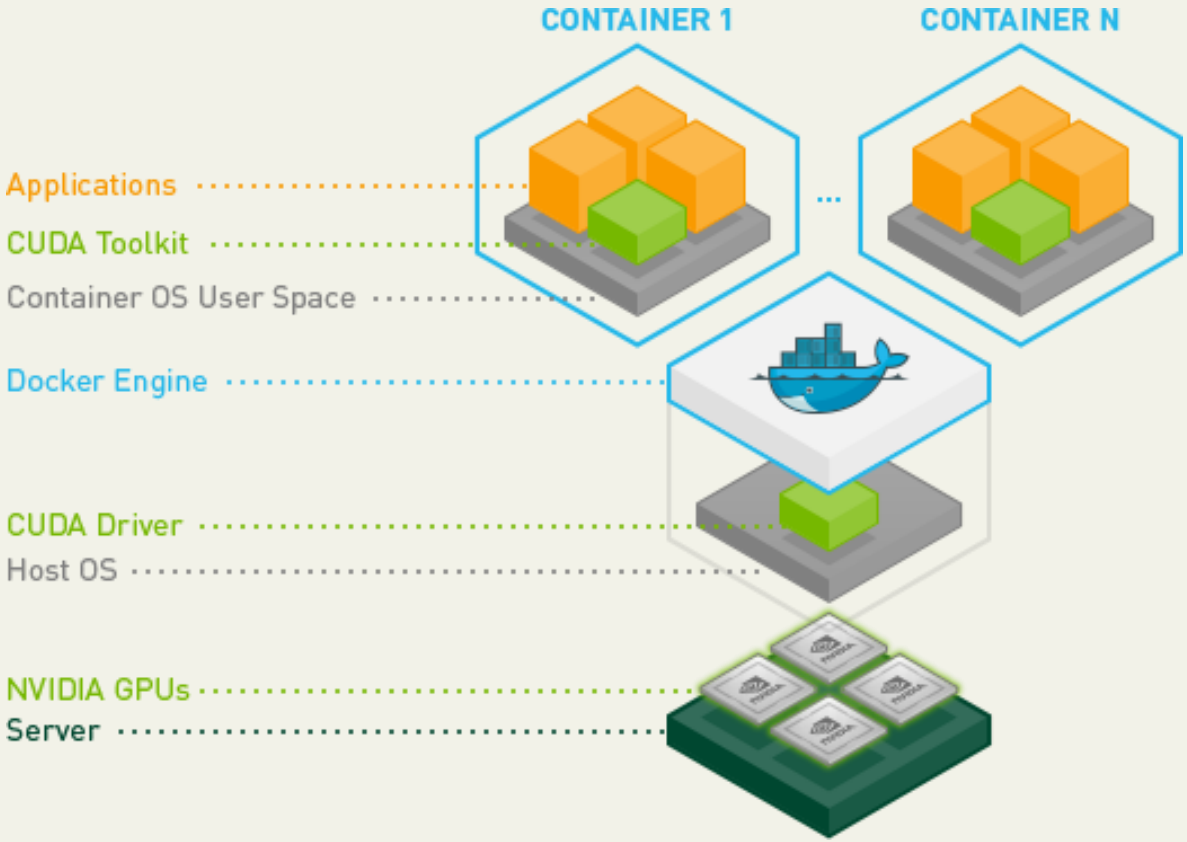
Running a Docker Container: Volumes

```
docker pull mbuckler/approx-vision


docker run \
-v <path to datasets>:/datasets \
-v <path to approx-vision>:/approx-vision \
-it mbuckler/approx-vision \
/bin/bash
```

What About GPU Use?

- NVIDIA Docker



What About X Windows (GUIs)?

1. Include the X packages in your Docker image

```
RUN apt-get update
RUN apt-get install -qqy x11-apps
```

2. Set the path to your .Xauthority file (and make if no ssh)

```
XAUTH=$HOME/.Xauthority
touch $XAUTH
```

3. Use host's networking stack & connect your displace and Xauthority environment file when you run the container

```
docker run --network=host --env DISPLAY=$DISPLAY -v $XAUTH:/root/.Xauthority
 -it <dockerimagename>
```

Docker Downsides

- **Storing many Docker images can fill up disk space**

- **Docker users have sudo privilege on the local machine when running containers. Two options:**
  1. Require that docker users run containers by invoking sudo
  2. Use docker group - Users have sudo privileges in container, not out

Conclusion

- **Docker is very useful!**
  - I use it every day in my research
- **Highly recommended for people who…**
  - Want their code to be easy to run (Docker for brg/pymtl?)
  - Need to install many diverse sets of dependencies (deep learning)
- **These instructions will be on the CSL wiki and my website**
  - www.markbuckler.com/post/docker-use/