The Container Talk

Steven R. Brandt

This talk …

- git clone https://github.com/stevenrbrandt/containertalk.git
- cd containertalk

What is docker?

- A lightweight virtual machine
- Leverages the existing linux kernel (on linux machines)
- Thus, images that are based on the newest Ubuntu or Fedora won't run on (for example) shelob.hpc.lsu.edu, because it's kernel is too old.
- Written with golang

A first docker script…

```
FROM fedora

RUN dnf install -y cowsay

CMD ["cowsay", "-f", "ghostbusters", "Who", "you", "gonna", "call?"]
```

- Build it by typing: “docker build –f ghost.docker –t ghost .”
- If your docker file is named “Dockerfile” you don’t need to type “-f Dockerfile”. It is assumed.
- The “ghost” is any name you want to make up to identify the image.
- To run it: “docker run –it –rm ghost”

A Second Docker script

```
FROM fedora

RUN dnf -y install curl perl bzip2 ncurses-devel ncurses-compat-libs SDL gtk2
SDL_image mesa-libGLU SDL_ttf

RUN curl -kLO http://www.bay12games.com/dwarves/df_44_12_linux.tar.bz2

RUN tar xjf df_44_12_linux.tar.bz2

WORKDIR /df_linux

# The game loads the wrong glibc if you leave this file in....

RUN rm -f libs/libstdc++.so.6

# Only text mode runs from this docke image...

RUN perl -p -i -e 's/PRINT_MODE:2D/PRINT_MODE:TEXT/g' data/init/init.txt

CMD ["bash", "./df"]
```

Docker: Creating a Dockerfile

- **Creating a Dockerfile is simple!**
- FROM fedora # Or other base image
- RUN dnf install –y findutils … # Install packages
- ENV LD_LIBRARY_PATH /my/lib64 # set environment variables
- WORKDR /some/directory
- RUN useradd –m someuser
- USER someuser
- RUN g++ -c foo.cc # Run a command as someuser
- CMD [“sleep”, “infinity”] # default command to run at startup

Creating a Docker image

- Docker saves each step, so when when you modify your Dockerfile and build again, you start building from the last successful step.
- When building, use the “—no-cache” option if you want to rebuild from scratch
- Often, it’s a good idea to use a specific version of things, e.g. instead of using “FROM fedora” use “FROM fedora:29”
- After you are done: “docker images|head” will show you your most recent images.
- Get the latest image: “docker images|head -2|tail -1|awk '{print $3}‘”
- Save this in a shell command named “docker-last”

Docker: Tagging and Pushing

- Create a dockerhub account
- “docker login”
- “docker tag myimage mylogin/myimage”
- “docker push mylogin/myimage”
- By default, the most recent builds of all images have a tag called “latest”
  - “docker tag myimage mylogin/myimage:tagname” if you want to specify
- “docker rmi image” removes an image

Docker: Running an Image

- To run: “docker run –it –-rm imagename”
- The “-it” means interactive
- The “—rm” means remove afterward
- Docker is always making a mess. Run “docker system prune –f” frequently.
- The above command runs docker with the default command. You can override this as follows: “docker run –it –rm imagename bash”
- Now you get a bash shell.

Docker: Danger!

- When you exit from “docker run” all your changes will be lost!
- What to do about that…
- Try: “docker run –it --rm –v /home/sbrandt:/home/jovyan imagename bash”
  - This command mounts /home/sbrandt from the host machine to /home/jovyan inside the docker image
  - It doesn’t work on Windows
  - It works differently on Mac

Docker: Danger!

- You can copy your files out of the container
- To do this, first run “docker ps”

```
CONTAINER ID        IMAGE                                COMMAND                  CREATED             STATUS              PORTS                                      NAMES
738c5c02072a        stevenrbrandt/et-juphub:latest       "/usr/local/bin/dumb..."   3 weeks ago         Up 21 hours         0.0.0.0:80->80/tcp                         et-juphub
679df8ebe14f        traefik:latest                       "/traefik --configFi..."   3 weeks ago         Up 21 hours         80/tcp, 0.0.0.0:443->443/tcp               traefik
```

Docker: Danger

- Now we can use the container id to copy files into and out of the image:
  - “docker cp 738c5c02072a:/path/file.cc .” to copy out
  - “docker cp myfile.cc 738c5c02072a:/path” to copy in
- Don’t like messing with container id’s? There’s a better way. When you start your container…
  - “docker run –it –rm –name foo someimage bash”
  - Now you can run…
  - “docker cp foo:/path/file.cc .”
  - “docker cp myfile.cc foo:/path”

A Third Docker Example…

- We will run a Jupyter Notebook in Docker!
- “docker build –f notebook.docker –t notebook .”
- “docker run –it –-rm notebook”
- You’ll see http://127.0.0.1:8009/?token=bdfd2cc…
- But connection won’t work…

Running notebooks and web servers

- By default, ports aren't exposed…
- "docker run –it –-rm –p 8080:80 someimage"
- The above command exposes port 8080 on the host machine and connects it to port 80 inside the image. If you're running Apache or something inside the container, you will be able to see it on port 8080 of localhost.
- So a docker command might look like
- "docker run –it –-rm –p 8080:80 –v /home/sbrandt:/home/sbrandt –-name foo someimage command"
- It's getting long!

Docker Compose…

```
version: '2'
volumes:
    jup-notebk-home: # A way to persist my work
services:
    notebook:
        build: # Optional build instructions
            context: .
            dockerfile: notebook.docker
        image: notebook
        container_name: nbk
        ports:
            - '8009:8009‘ # Expose the port
        volumes:
            - jup-notebk-home:/home/jovyan
        restart: on-failure
```

Docker Compose…

- Now we can say…
  - “docker-compose up”
  - “docker-compose down”
  - “docker-compose up –d”

Things I use Docker for…

- http://einsteintoolkit.org
  - When I inherited the website, it was running RHEL4
  - For security reasons, it needed an update
  - No one knew how it was put together. I used trial and error…
  - https://github.com/stevenrbrandt/et-websites/blob/master/etk-website.docker
- https://docs.einsteintoolkit.org
  - Similar to the above
- http://tutorial.cct.lsu.edu/hpx
  - The cling notebook doesn't seem to build right without the notebook anymore..
- http://tutorial.cct.lsu.edu/etk,
- http://tutorial.cct.lsu.edu/beowulf

Things I use Docker for…

- expression_trees
  - https://github.com/kawilliams/expression-trees.git
- CMR – Coastal Model Repository
  - https://github.com/ysboss/agave-model.git
  - Docker image is auto-built by dockerhub with a git hook
- PhyalanxBuilder
  - https://github.com/stevenrbrandt/PhylanxBuilder.git
  - Ensures a complete working build environment for Phylanx
  - Includes Apex, Tensorflow, Keras, CNTK

Using phylanx.devenv

- “docker run --name devenv --privileged -v devenv_homefs-phylanx:/home/jovyan -d --rm stevenrbrandt/phylanx.devenv:working”
- The “CMD” in the image is [“sleep”, “infinity”]
- The “-d” runs the image in the background
- I “login” to the image by typing “docker exec –it devenv bash”
- You can think of “docker exec” as being like “ssh”
- The other way to use the phylanx.devenv image is with Singularity…

What is Singularity?

- Docker needs access to root.
  - Inherently insecure.
- Not used on HPC systems.
- Singularity is an alternative.
  - Also built on golang
- “singularity build –F ~/images/phylanx.devenv docker://stevenrbrandt/phylanx.devenv:working”
- “singularity shell ~/images/phylanx-devenv.simg”
  - Now you can type “build.sh” to build Phylanx.