Running Containers at NERSC with Shifter

Shane Canon
NERSC Early User Training Day 2019
January 25, 2019
Outline

• Quick Intro to Containers
• Role of Shifter
• Walk through of using Docker and Shifter
Intro to Containers and Shifter
Docker Basics

- Build images that captures applications requirements.
- Manually commit or use a recipe file.
- Push an image to DockerCloud, a hosted registry, or a private Docker Registry.
- Share Images
- Use Docker Engine to pull images down and execute a container from the image.
What’s in an Image

• Directory tree
  – Base Linux OS
  – Libraries, binaries, tools, scripts, etc
  – User code
  – Data

• Run-time Settings
  – Environment variables
  – Working Directory
  – Default execution and parameters

• Other things (not relevant to Shifter)
  – Network-related (e.g. ports)
  – Run User
Why not just run Docker

• Security: Docker currently uses an all or nothing security model. Users would effectively have system privileges

```
> docker run -it -v /:/mnt --rm busybox
```

• System Architecture: Docker assumes local disk
• Integration: Docker doesn’t play nice with batch systems.
• System Requirements: Docker typically requires very modern kernel
• Complexity: Running real Docker would add new layers of complexity
Shifter

• NERSC R&D effort, in collaboration with Cray, to support Docker Application images
• “Docker-like” functionality on the Cray and HPC Linux clusters
• Addresses security issues in a robust way
• Efficient job-start & Native application performance
Why Users will like Containers and Shifter

• Develop an application on your desk top and run it on Cori and Edison
• Enables you to solve your dependency problems yourself
• Run the (Linux) OS of your choice and the software versions you need
• Improves application performance in many cases
• Improve reproducibility
• Improve sharing (through sites like Dockerhub)
Containers and Science

• Reproducibility
  – Everything you need to redo a scientific analysis
  – Image manifest contains all information about environment
    • Scripts, portable input files can be managed with version controller for greater control

• Portability
  – Runs on every system

• Reduction of Effort
  – Compile takes 10 hours? Just do it once and share it with everyone
  – System doesn’t have the right library version? Yum install or apt-get it yourself in the container
Create an image with Docker

FROM ubuntu:14.04
MAINTAINER Shane Canon scanon@lbl.gov

# Update packages and install dependencies
RUN apt-update -y && \
    apt-get install -y build-essential

# Copy in the application
ADD . /myapp

# Build it
RUN cd /myapp && \
    make && make install

laptop> docker build --tag scanon/myapp:1.1 .
laptop> docker push scanon/myapp:1.1
Use the Image with Shifter

#!/bin/bash
#SBATCH -N 16 -t 20
#SBATCH --image=scanon/myapp:1.1

module load shifter
export TMPDIR=/mnt
srun -n 16 shifter /myapp/app

# Submit script

cori> shifterimg pull scanon/myapp:1.1
cori> sbatch ./job.sl
Shifter and MPI

- Shifter has a “built-in” approach for supporting MPI applications in containers.
- Build Applications using ABI compatibility.
- Shifter automatically maps in appropriate libraries at run time.
- No rebuild required, but may not work for all cases.
- Can provide native MPI performance.
# This example makes use of an Ubuntu-based NERSC base image
# that already has MPI built and installed.

FROM nersc/ubuntu-mpi:14.04

ADD helloworld.c /app/

RUN cd /app && mpicc helloworld.c -o /app/hello

ENV PATH=/usr/bin:/bin:/app:/usr/local/bin

> shifterimg pull scanon/myapp:1.1
> salloc -n 128 --image=scanon/myapp:1.1 --C haswell
# srun -n 128 shifter /myapp/app
Shifter accelerates Python Apps

![Graph showing Pynamic v1.3, 4800 MPI ranks]

- Median Start-up + Import + Visit Time (s)
  - Cori P1
  - Edison

<table>
<thead>
<tr>
<th>Configuration</th>
<th>scratch</th>
<th>project</th>
<th>datawarp</th>
<th>common</th>
<th>tmpfs</th>
<th>shifter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cori P1</td>
<td>500</td>
<td>300</td>
<td>200</td>
<td>100</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Edison</td>
<td>350</td>
<td>250</td>
<td>150</td>
<td>80</td>
<td>80</td>
<td>30</td>
</tr>
</tbody>
</table>
Shifter behavior versus Docker

• Processes run as your user id (not root).
• Images are mounted read-only (so you modify files in the image).
• Home directories and global file systems are automatically mounted.
• Some handling of special Dockerfile directives isn’t yet supported
Other things of Note

- Shifter supports volume mounts that allow you to map a directory (e.g. $SCRATCH) into another location in your image.

- Shifter supports per-Node write-able scratch spaces that work well for apps that want a local disk.

- NERSC runs a private registry (registry.services.nersc.gov) that can be used to store private images that you can’t put in DockerHub.
Shifter versus Spin

Shifter
- Runs processes as the user
- Runs on the HPC systems

Best for:
- Simulation or analysis runs
- Need to run at scale
- Need to read/write a lot of data

Spin
- Runs with stock Docker and Rancher
- Runs on dedicated hardware

Best for:
- Running services or processes that need to run “indefinitely”
- Services that need to be externally accessible
Measuring the Composition of the Universe

• CMB – S4
  – Ambitious collection of telescopes to measure the remnants of the Big Bang with unprecedented precision

• Simulated 50,000 instances of telescope using 600,000 cores on Cori KNL nodes.

• Why Shifter?
  – Python wrapped code needs to start at scale
Where can you learn more

- **NERSC Docs Website**
  - docs.nersc.gov
  - Running Jobs Containers Overview
    - https://docs.nersc.gov/development/shifter/overview/

- **Previous Training**
  - [https://github.com/nersc/Shifter-Tutorial](https://github.com/nersc/Shifter-Tutorial)

- **Docker Resources (Numerous)**
  - [https://docs.docker.com/get-started/](https://docs.docker.com/get-started/)