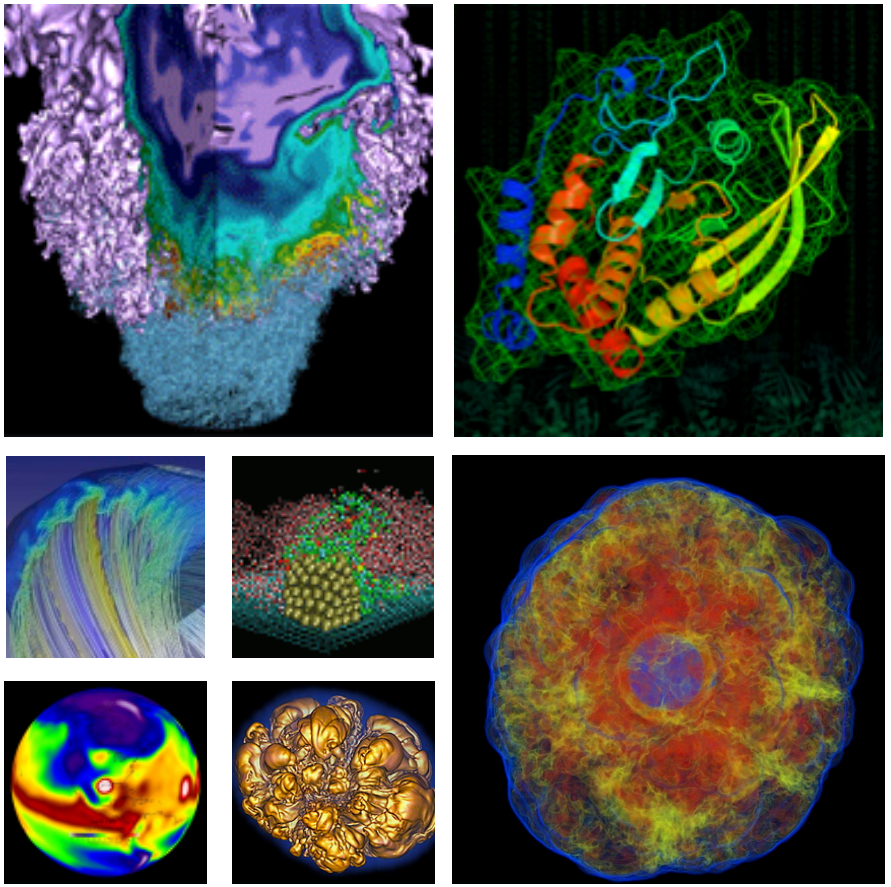
Into to Containers



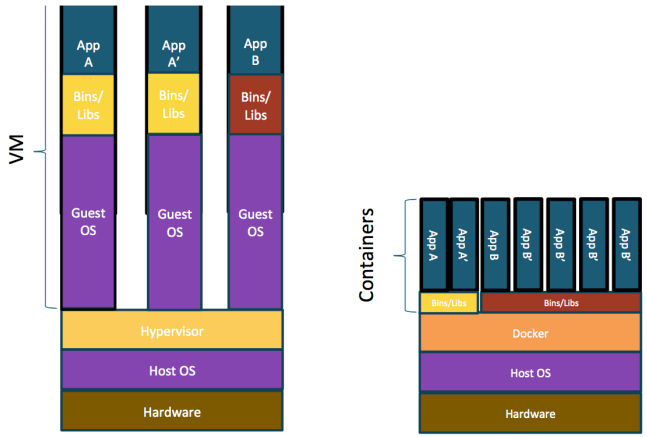
Lisa Gerhardt

Data & Analytics Group, NERSC

November 13, 2017

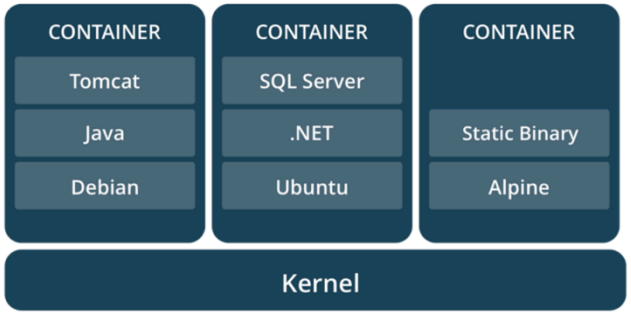
Problem Statement

- **Need a way to run isolated processes**
  - Resources
  - File system view
  - Process space
  - Network space
- **But should be lighter-weigh than a virtual machine**
  - Avoid a separate kernel
  - Avoid a full running OS “stack” (e.g. system daemons)



Why Containers?

- **Light weight, executable piece of software that contains everything you need to run it**
  - Code, system libraries and tools, environment, settings
- **All software and processes are isolated from their surroundings**
- **Portable**
- **Typically used for single instance programs**

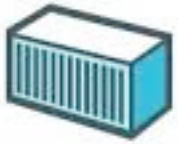


Solution - Containers

- **Uses a combination of Kernel "cgroups" and "namespaces" to create isolated containers**
- **Software to help define and manage containers**
  - 1979 – chroot introduced
  - 2000/2001 – FreeBSD Jails and Linux Vserver
  - 2006 – Process Containers (later became cgroups)
  - 2008 – LXC (cgroups and namespaces, no patches needed)
  - 2013 - LMCTFY (Google)
  - 2013 - Docker

Docker Basics

Build



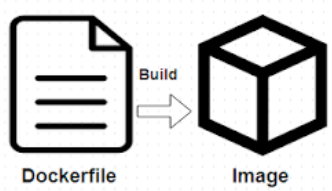
Ship

Run

- Build images that captures applications requirements.
- Manually commit or use a recipe file.

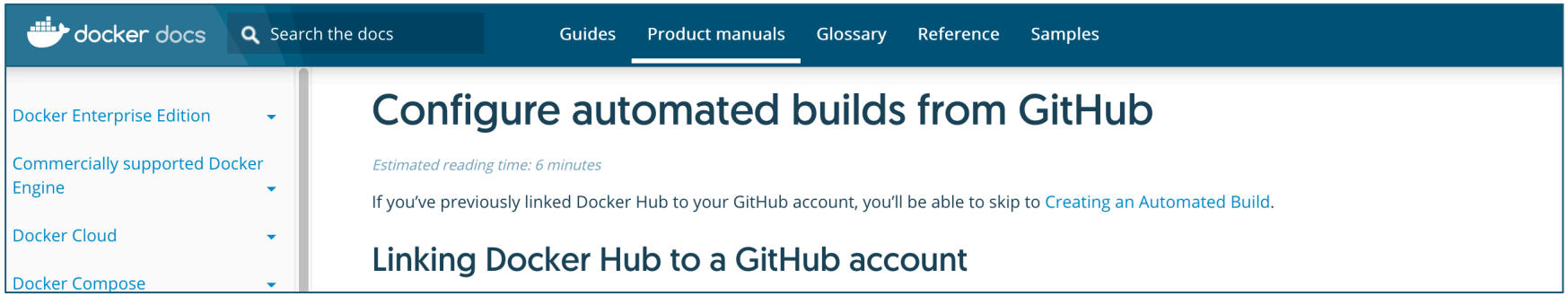
- Push an image to DockerCloud, a hosted registry, or a private Docker Registry.
- Share Images

- Use Docker Engine to pull images down and execute a container from the image.

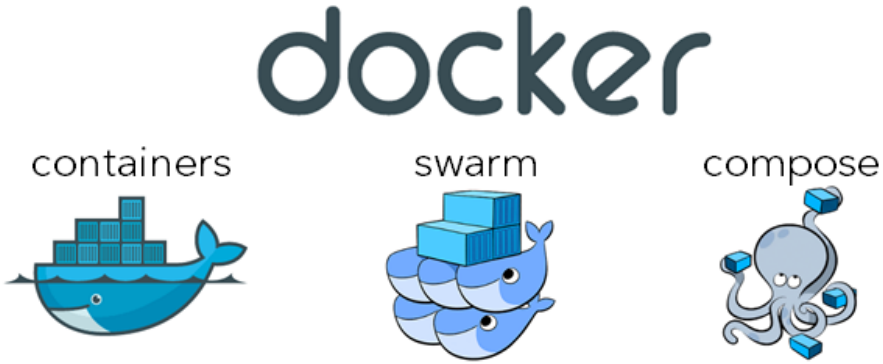


Advanced Docker Features

Continuous Integration:



Complex deployment of (mostly web) services



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 it yourself in the container

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**
  - Network-related (e.g. ports)
  - Run User

National Energy Research Scientific Computing Center