Introduction to Docker

Matt Eldridge

CRUK-CI Bioinformatics Core

What is Docker?

- ✓ A virtualization platform
- ✓ A way to package an application, and all its dependencies, and share it with others
- ✓ An isolated environment in which to install and try new software

docker

/ˈdɒkə/

Noun

1. A **container system for wrapping a piece of software in a complete file system with everything it needs to run**
2. A person employed in a port to load and unload ships

Docker Components

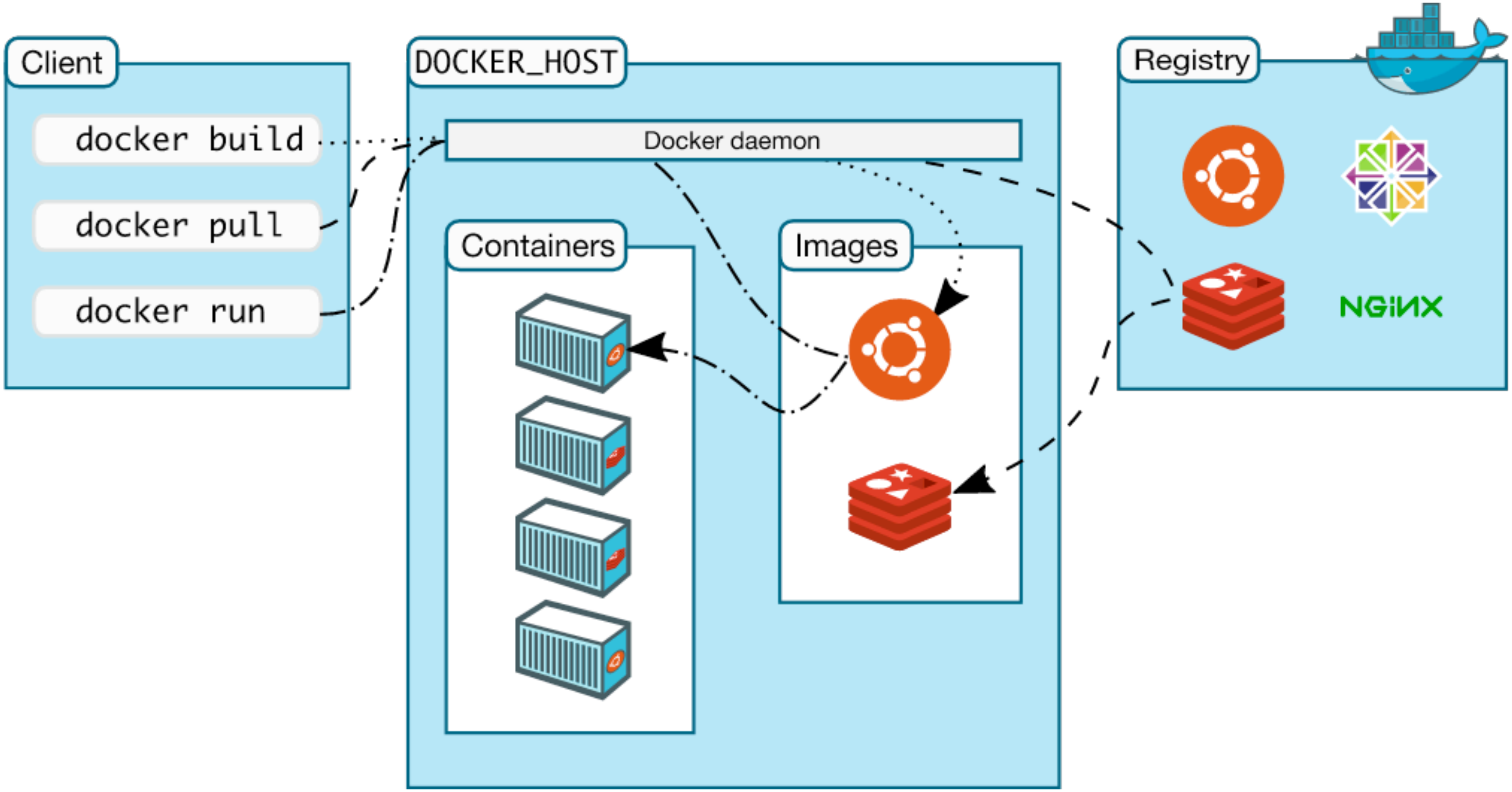
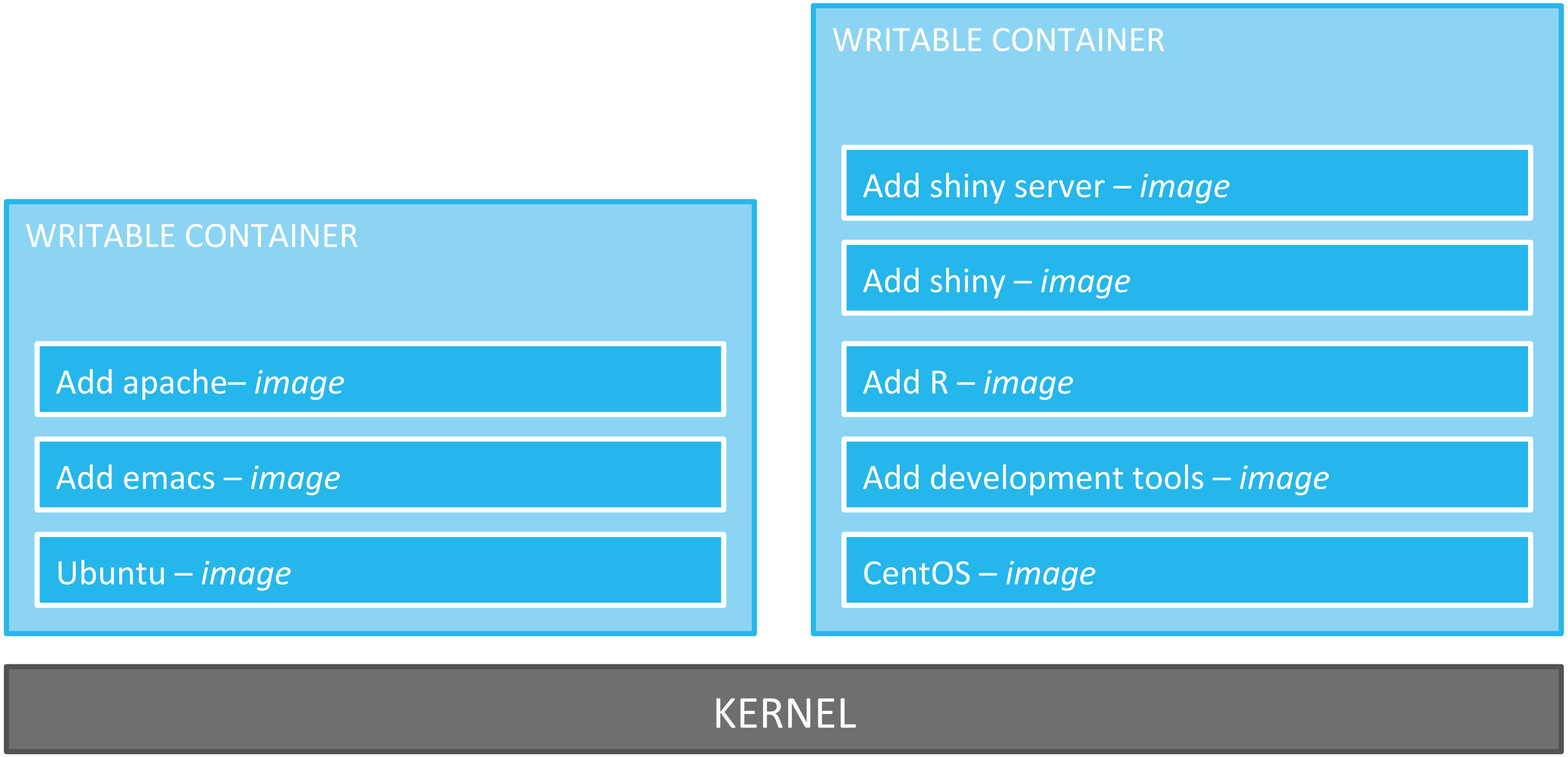


Image from Docker website https://docs.docker.com/engine/understanding-docker

Docker Containers



Layered filesystem – sharing common files for efficient disk usage and image downloads

Images can be built using **Dockerfile** templates

Getting Started

Pull an image from a repository

```
docker pull bioconductor/release_base
```

Run a command within a new container based on this image

```
docker run -it bioconductor/release_base R
```

Building an image

1 Start from an existing image, e.g. the base CentOS image, and create a container running a shell

```
docker run -it centos bash
```

2 Install new software, add data files, etc.

3 Exit from the shell and find the container ID

```
docker ps —a
```

4 Save the container as a new image

```
docker commit cranky_feynmann myimage
```

➢ Builds can be automated using a **Dockerfile**

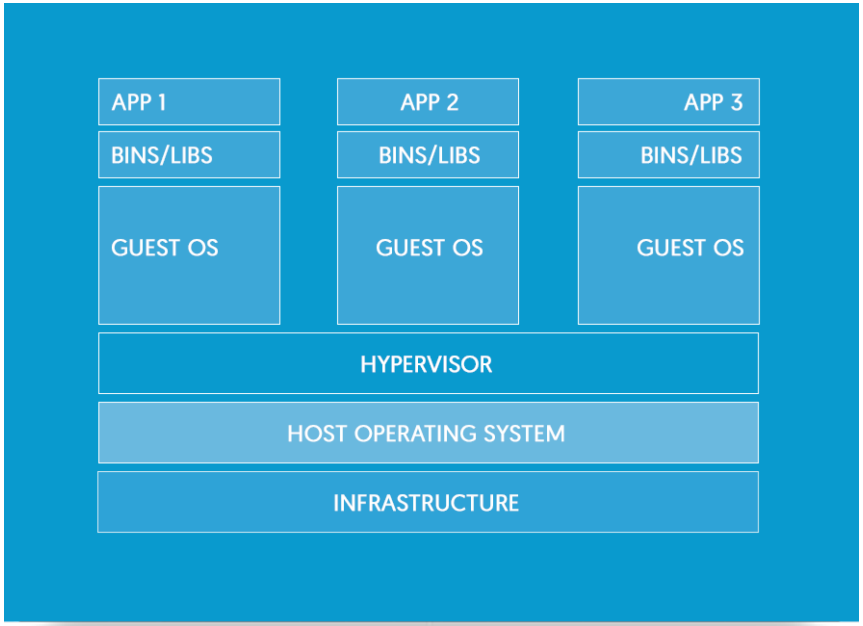
Dockerfile

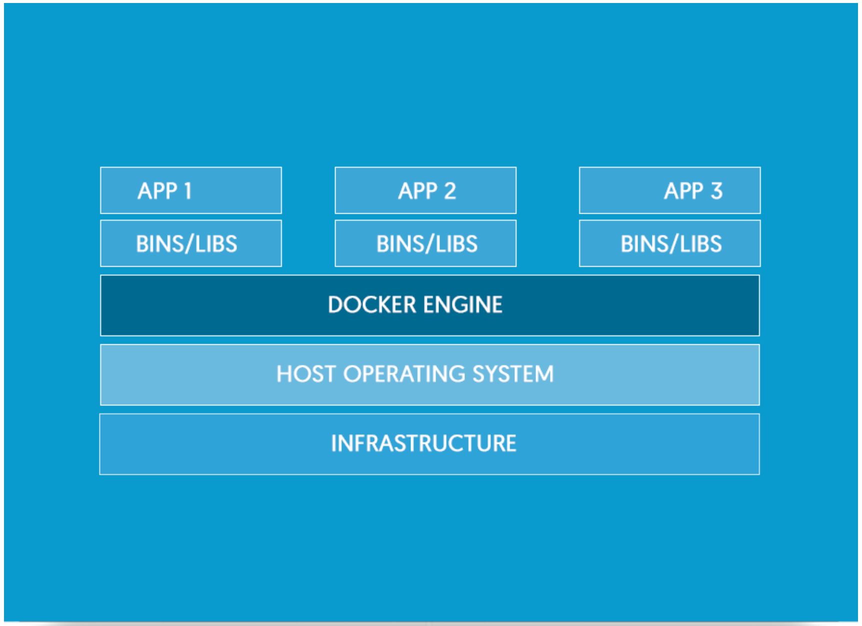
A Dockerfile is a set of instructions for assembling an image

Example: **Shiny server**

```
FROM centos:7

RUN yum groupinstall -y 'development tools'
RUN yum install -y wget

RUN rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
RUN yum install -y R

RUN R -e "install.packages(c('shiny', 'rmarkdown'), repos='https://cran.rstudio.com')"
RUN R -e "install.packages('devtools', repos='http://mirrors.ebi.ac.uk/CRAN')"

RUN wget https://download3.rstudio.org/centos6.3/x86_64/shiny-server-1.5.0.730-rh6-x86_64.rpm
RUN yum install -y --nogpgcheck shiny-server-1.5.0.730-rh6-x86_64.rpm

EXPOSE 3838

COPY shiny-server.sh /usr/bin/shiny-server.sh

CMD ["/usr/bin/shiny-server.sh"]
```

Docker vs Virtual Machines





Virtual Machines *(VMware, VirtualBox)*

Each virtual machine includes the entire guest OS – tens of GBs, can take minutes to start up

Docker

Containers run as isolated processes on the host OS – lightweight, start instantly, use less memory

Security issues

- Elevated privileges
  - Docker daemon requires root privileges on linux
  - Allows containers to have root access to the host filesystem
- Weaker isolation than VMs
  - Attacks (viruses, intrusions) can propagate down to the underlying OS and into other containers
- *Cannot run dockerized apps on the CRUK-CI HPC clusters* ☹

How have we used Docker?

Running third party software packaged as “dockerized apps”	**Sanger Cancer Genome Project analysis pipeline** (variant calling for whole genome sequencing) **Polysolver** (HLA typing)
Installing and running third party software	**MutSigCV** (mutational significance)
Deploying Shiny applications	**Proteomics TMT analysis** (Bioinformatics Core) **Breast Cancer PDTX Encyclopaedia** (Caldas lab, Bioinformatics Core)
Packaging and distributing tools developed in-house	**SVclone** tumour clonality analysis tool for ICGC-TCGA-DREAM Challenge (Geoff MacIntyre) **ParaBam** tool for optimized processing of BAM files (Henry Farmery)
Training	CRUK Summer School on cancer genome analysis

So how can Docker help me?

- Trialing new software
  - Clean, unpolluted starting point
  - Isolated environment, won’t affect other applications
  - Superuser privileges and complete control over what you install

- Bioinformatics developers increasingly using docker to package and distribute applications

- Deployment of an application during development
  - Share your environment with a colleague to run on their machine
  - Update a production system with minimum downtime