Cloud Computing and Big Data

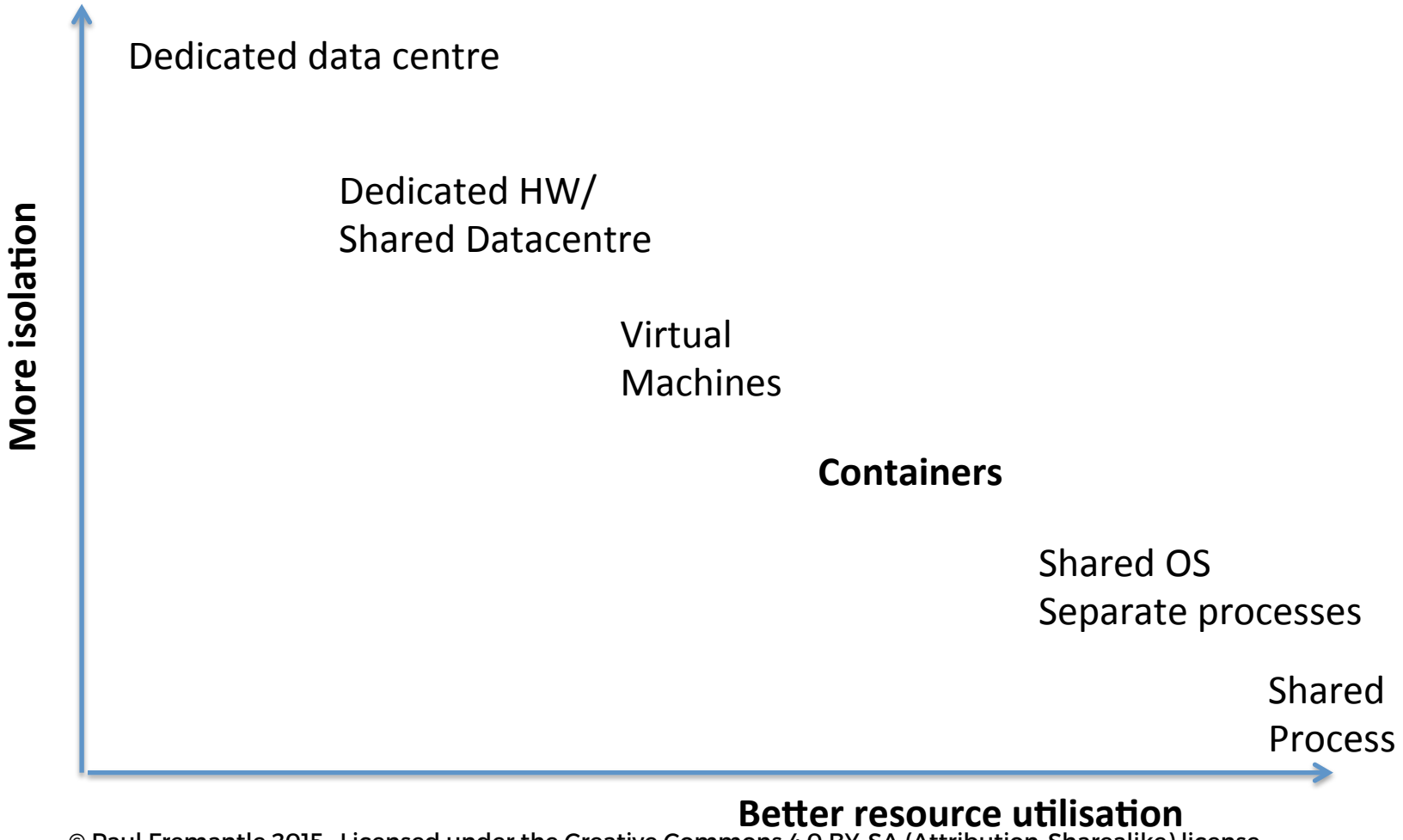
Containers and the evolution of PaaS

Oxford University
Software Engineering
Programme
Nov 2015

Contents

- Containers
- History and Approach
- Docker
- Docker ecosystem
- PaaS in a container model
- Futures

Sharing of resources vs Isolation



Lightweight Virtualization history

- zSystems Virtual Servers from late 1990s
  - (the mainframe really did do everything first)
- Solaris Containers
- AIX Workload Partitions
- FreeBSD Jail
- …

What is a Container?

- A lightweight virtual server
  - Running within an Operating System
  - Providing various levels of isolation and control
  - E.g. Disk isolation and control
  - Network isolation
  - CPU and memory controls

Containers at Google

- Every GMail session is a container
  - Try doing an export and then searching your email ☺
- "Everything runs in a container"
- **2 billion** containers launched a week
- Borg
  - **Any** Google developer can instantiate their code in **10,000 instances** any time they want
  - Takes about 5 minutes to start that many
  - Never exactly 10,000 because of failures

Linux Containers (LXC)

- Virtualization inside the Linux Operating System
  - Not the only Linux option, but the most popular
- Allows virtualization including CPU, memory, disk
- Simple and effective

A sample of LXC

- apt-get install lxc
- lxc-create -t ubuntu -n cn-01
- lxc-start –n cn-01
- lxc-console –n cn-01
- lxc-freeze –n cn-01

cgroups

- Control of resources by process:
  - blkio — this subsystem sets limits block devices such as physical drives
  - cpu - access to the CPU.
  - cpuacct — this reports on CPU usage
  - cpuset — this controls usage by CPUs in a multicore
  - devices — this denies or grants access to devices
  - freezer — suspends and resumes tasks
  - memory — controls and reports on memory usage
  - net_cls — tags network packets with ids for control
  - net_prio — priority of network traffic per interface.
  - ns — the namespace subsystem.

libcontainer and the Open Container Foundation

- A standardised interface into the container layer
  - Part of runC the open runtime from Docker
  - A key basis of the Open Container Foundation

Cloud Native Computing Foundation

- A new definition of “Cloud Native”
  - Container Packaged
  - Dynamically Managed
  - Micro-Service oriented

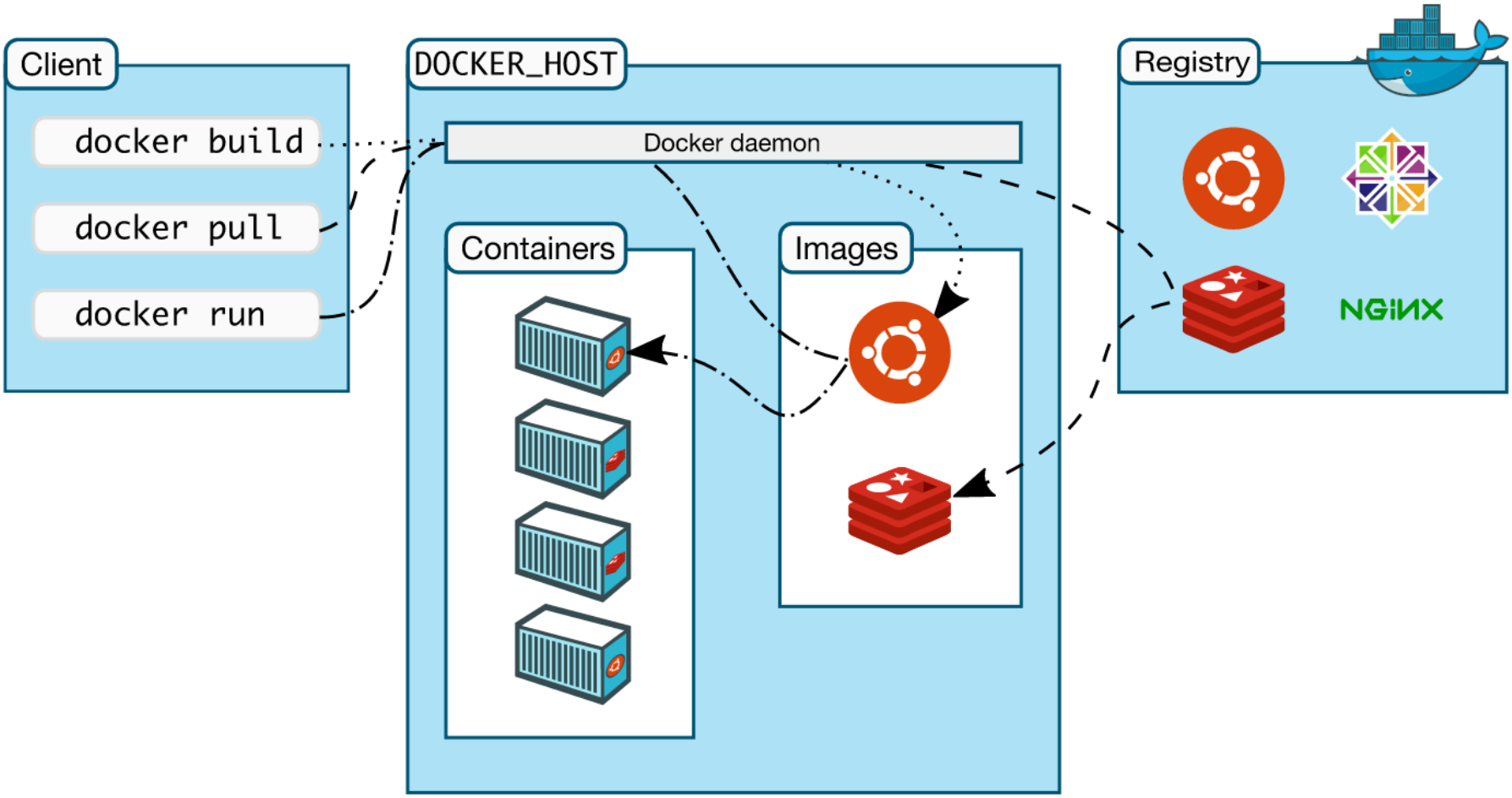
Docker on top of LXC

- Docker adds several things to LXC and containerization:
  - Copy on write filesystem
    - Layered images and the ability to extend machines easily
  - Simple textual config file
  - Portable deployment across machines
    - Creating an ecosystem of images
  - Application centric
    - Each VM is a process (roughly speaking)
  - Plus others (auto-build, etc)

Why Docker?

- The *ecosystem* has created a *network effect*
- Metcalfe's Law states
  - the value of a telecommunications network is proportional to the square of the number of connected users of the system
- There is surely a corollary for ecosystems

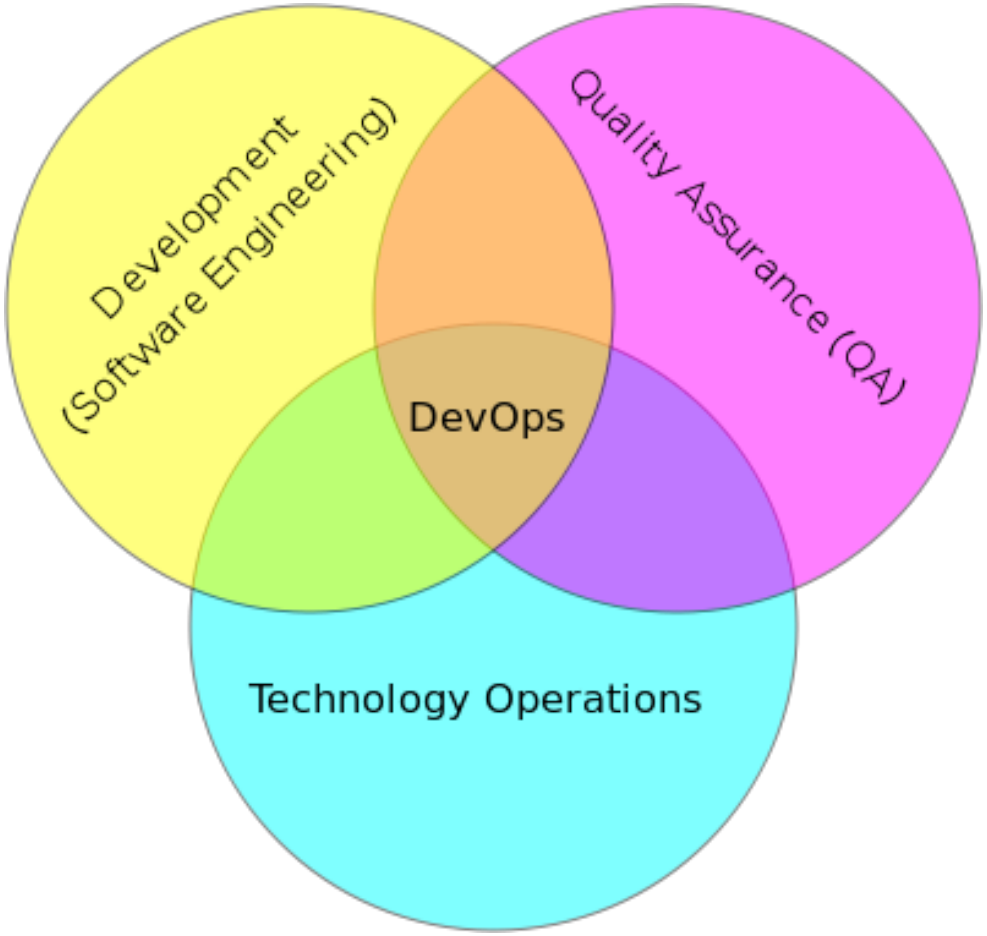
How does Docker work?



Some simple Docker commands

- apt-get install docker.io
- docker pull ubuntu
- docker run –t –i ubuntu /bin/bash
- docker ps
- docker commit funky_freo image
- docker push image

DevOps



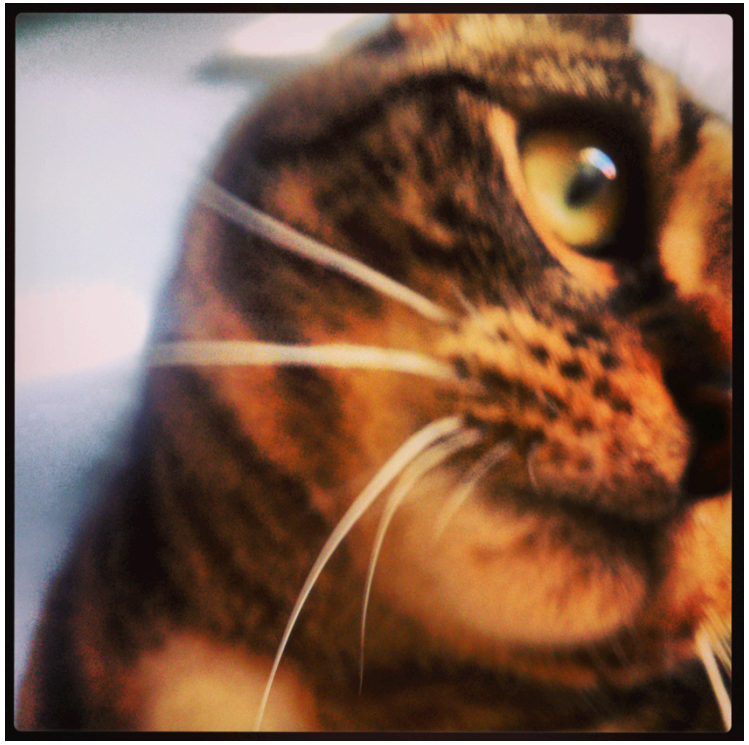
DevOps

- DevOps is the codification of the interface between Development and Operations
  - Agile
  - Repeatable
  - Collaborative
  - Versioned
  - Automated

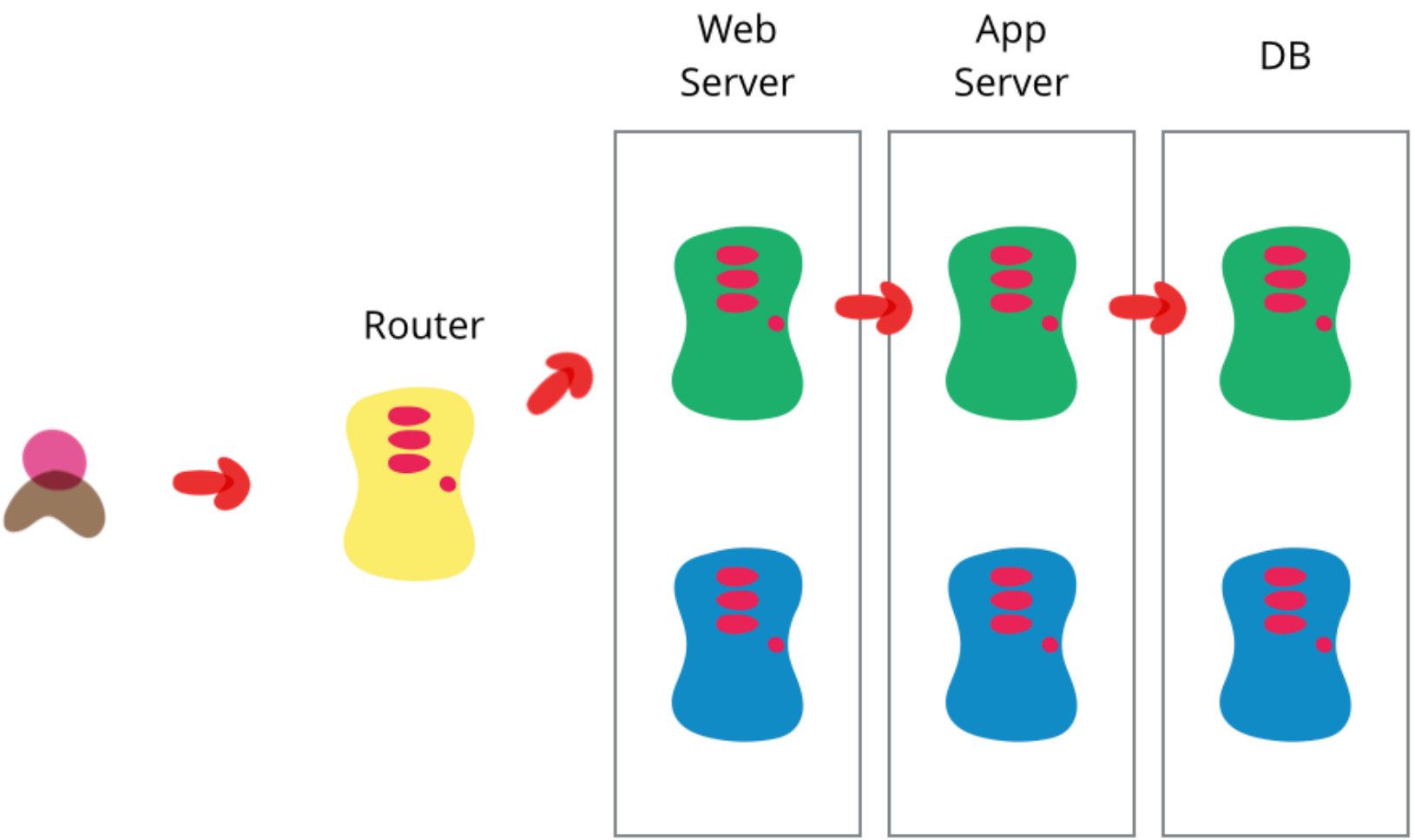
Cloud and DevOps

- It could be argued strongly that the rise of DevOps is tied to the rise of Cloud
  - Clear requirement for automated, repeatable configuration and deployment
  - Reducing the hardware provisioning time has highlighted the challenges

Kittens vs Cattle
(An unpleasant but effective analogy)



Blue Green Deployment



http://martinfowler.com/bliki/BlueGreenDeployment.html

DevOps tools

- Puppet, Chef
  - Automated configuration and deployment tools
  - Allow complex infrastructures to be re-configured automatically
- Vagrant
  - Create VMs instantly
- Plus many many more!

DevOps and Docker

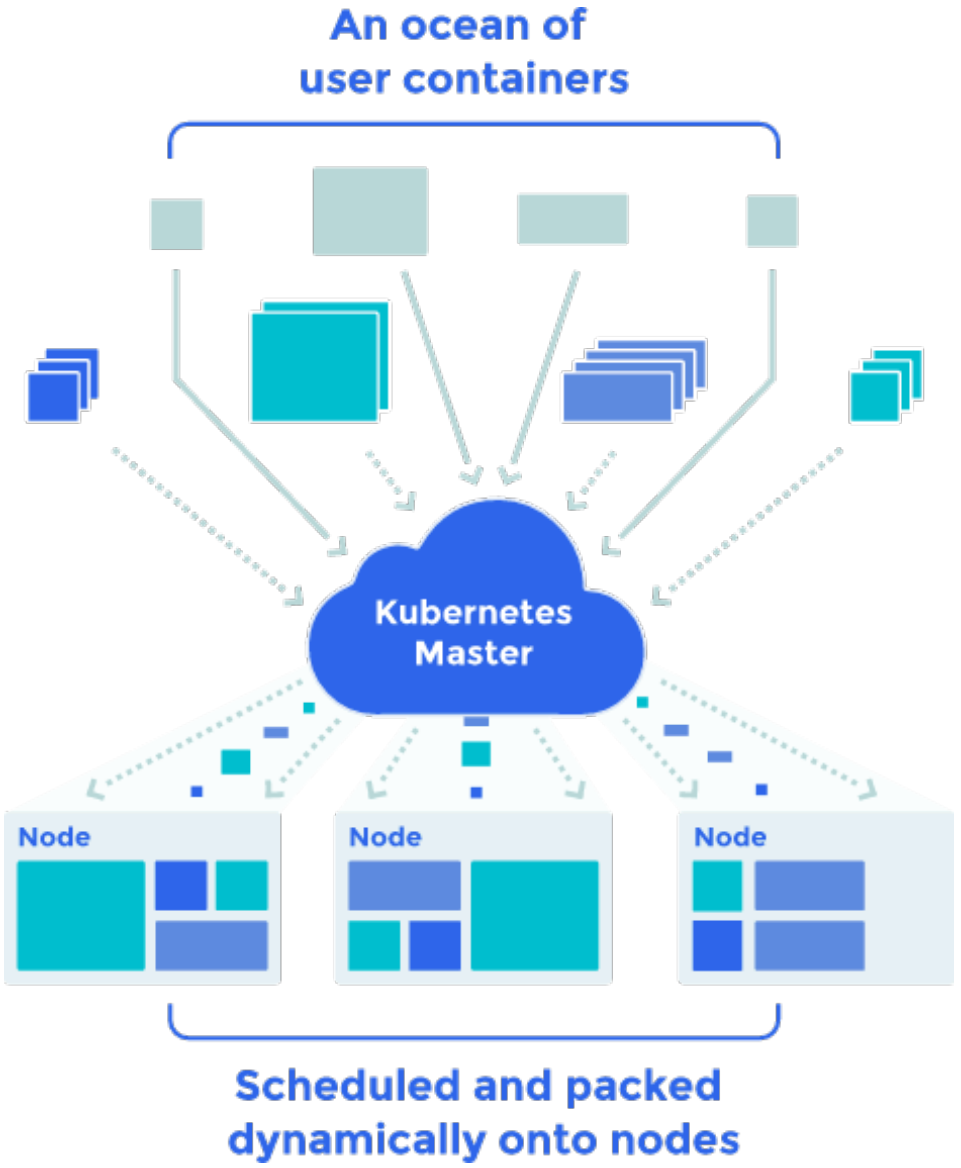
- Docker is a key DevOps tool
- Speeds up the creation of repeatable deployments
- Consistency between development, test and production
- Versioned repository
- Works with Chef, Puppet, etc

Challenges with Docker and Solutions

- Networking
  - It is very complex to connect different containers, even on a single machine
    - Weave Networks
    - SocketPlane (bought by Docker)
- Clustering
  - Docker Swarm
  - Google Kubernetes
  - CoreOS
  - Apache Mesos
- Lack of mutable file system
  - Flocker

Kubernetes

- Open Source cluster management of containers
- From Google, but separate from the Borg project



Docker ecosystem

Summary

- Docker and the Container model
  - Lightweight virtualization and repeatability
  - Blue Green deployment
  - “Warehouse Scale” computing