CONTAINERS 101

HOUSE RULES

- Ask questions at any time

- Follow us at: @IBMCodeLondon

AGENDA

Containers:

- Setting the scene

- What are containers?

- What are the advantages of containers?

Docker:

- What is Docker?

- Overview of the Docker ecosystem

Hands-on Workshop

CHANGES TO SOFTWARE CONSUMPTION

- BEFORE: "Download our software and follow the docs to configure your environment"

- NOW: "Here is the software, we'll host it, just access it over the internet"

- WHY?

  - It just works

  - We don't have expertise

  - We don't care

SaaS

CHANGES TO BUILDING SOFTWARE

- BEFORE: "Let's build this thing that can do A, B and C... and Z and then we'll test it in a year!"
- NOW: "Let's have short dev. cycles and split into smaller teams working on capabilities"
- WHY?
  - More efficient to reuse smaller components
  - More efficient to scale with smaller components
  - Easier to do agile with smaller components
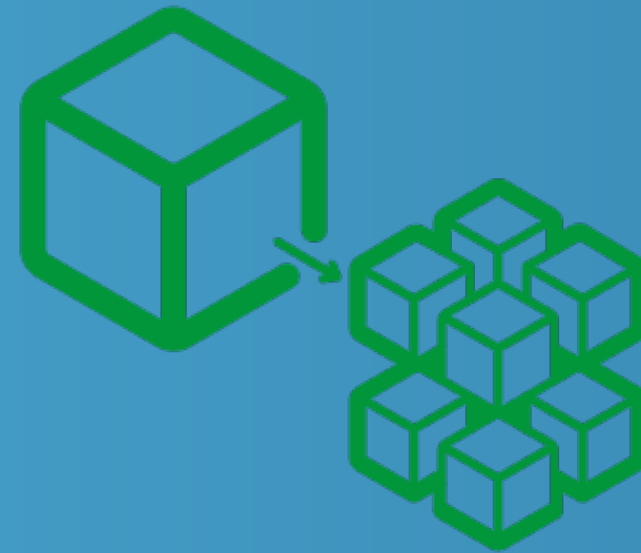
EXPERIENCE IS THE BEST TEACHER…

- VI – Processes - Execute the app as one or more stateless processes
- VII – Concurrency - Scale out via the process model
- IX – Disposability – Maximize robustness with fast startup and graceful shutdown

https://12factor.net

WHERE DO CONTAINERS FIT INTO THIS?

CONTAINER BASICS

- A container contains!
- Virtualizing the hardware OS subsystems
- Interaction with a single OS

CONTAINER ISOLATION

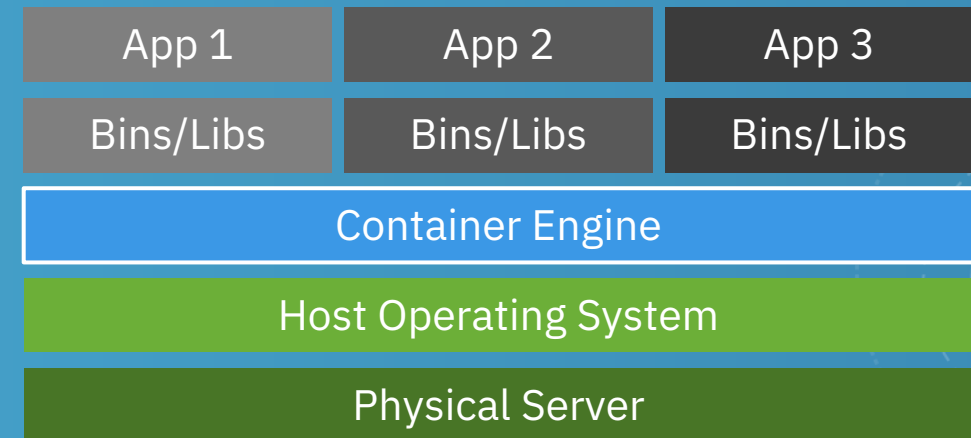**Namespaces**

"What you can see"

- Process IDs
- Filesystems
- Users
- IPC
- Networking

Cgroups

"what you can use"

- CPU
- Memory
- Disk I/O
- Network
- Device permissions (/dev)

VMS VS. CONTAINER VIRTUALIZATION

App 1	App 2	App 3
Bins/Libs	Bins/Libs	Bins/Libs
Guest OS	Guest OS	Guest OS

Hypervisor

Host Operating System

Physical Server

App 1	App 2	App 3
Bins/Libs	Bins/Libs	Bins/Libs

Container Engine

Host Operating System

Physical Server
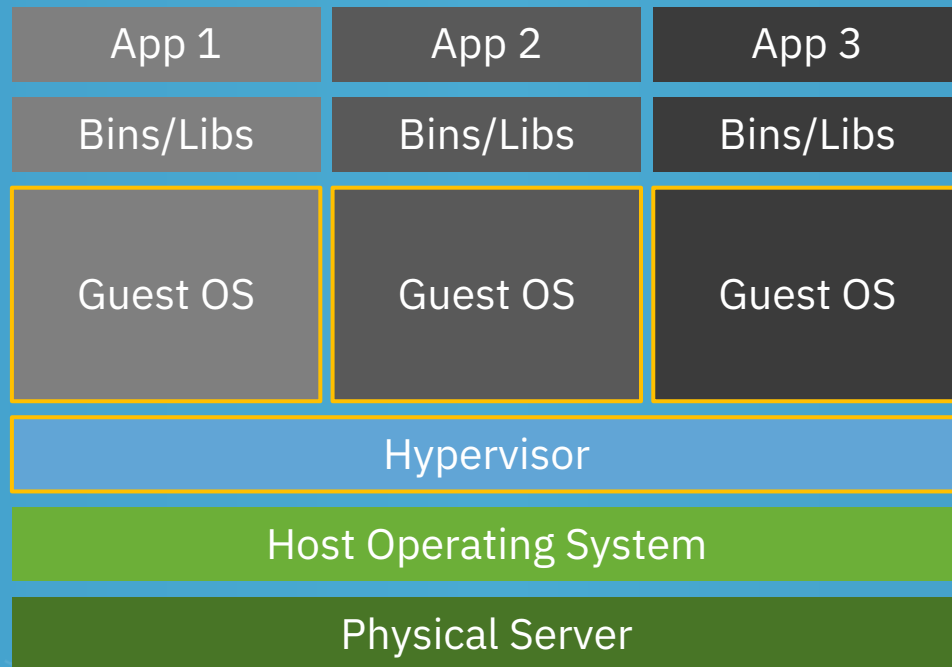
ADVANTAGES
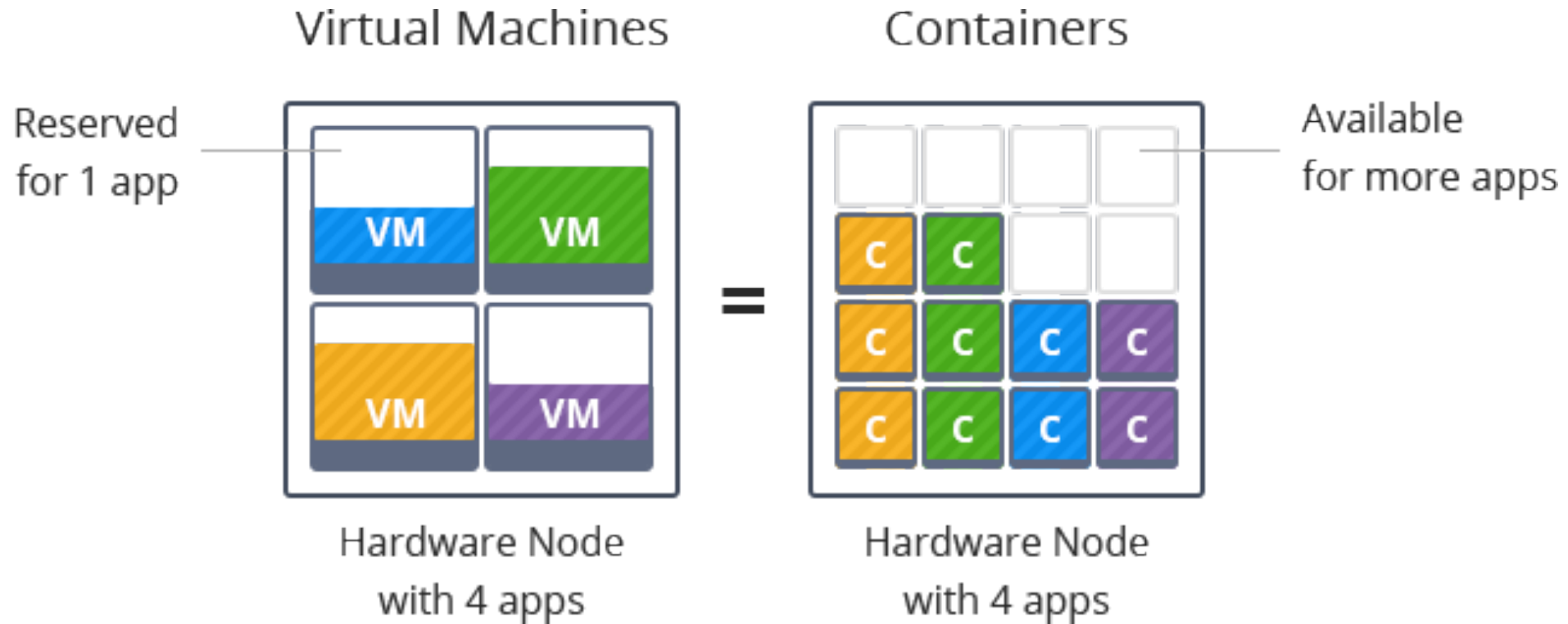
SO CONTAINERS ARE A NEW THING?

WELL, NOT SO NEW...

- Unix V7's Chroot – 1979
- Linux VServer – 2001
- Oracle Solaris Containers – 2004
- Open VZ (Open Virtuzzo) - 2005
- **Google's Process Containers (Control Groups – cgroups) – 2006**
- **LXC (Linux Containers) – 2008**
- CloudFoundry's Warden – 2011
- **Docker – 2013**
- Open Container Initiative – 2015
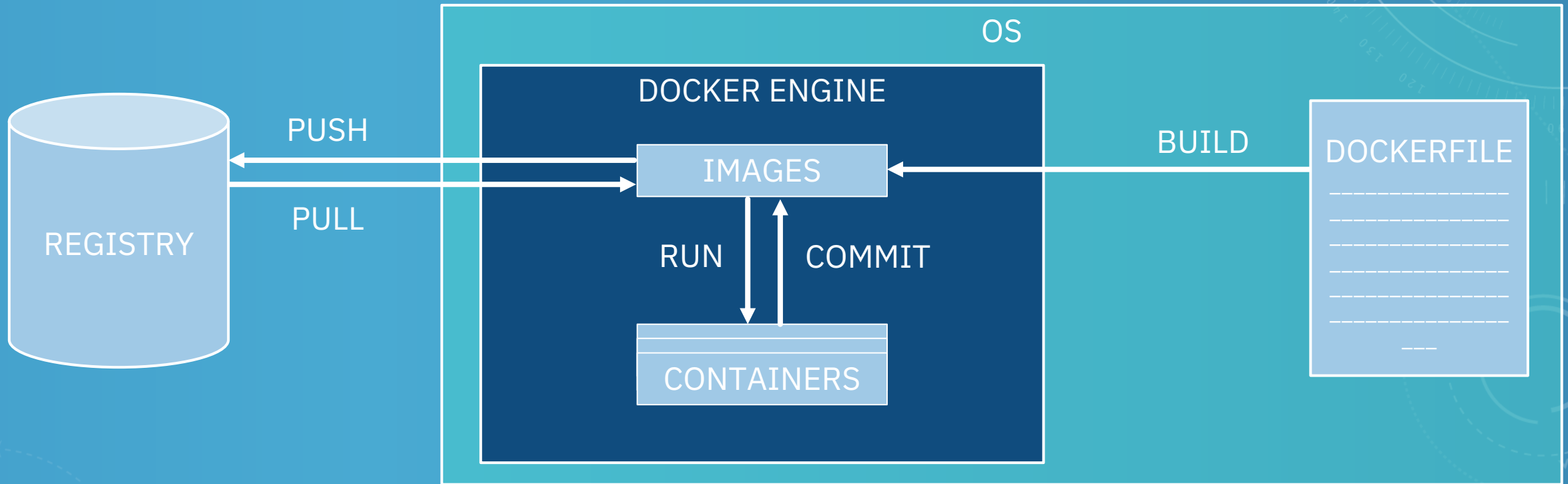- The rise of the container tools – 2017

WHAT IS DOCKER?

DOCKER BASICS

- The Docker image format

- Docker engine which instantiates containers and manages the lifecycle

DOCKER ARCHITECTURE

DOCKERFILE

- Each line is a layer
- Dockerfile commands:
  - FROM
  - LABEL
  - RUN
  - CMD/ENTRYPOINT
  - VOLUME
  - ENV
  - EXPOSE

```
FROM ubuntu

LABEL maintainer="Bob Smith (bob.smith@gmail.com)"

RUN apt-get update

RUN apt-get install -y nginx

CMD ["nginx", "-g", "daemon off;"]

EXPOSE 80
```

DOCKER IMAGES

FROM ubuntu

LABEL maintainer="Bob Smith (bob.smith@gmail.com)"

RUN apt-get update

RUN apt-get install -y nginx

CMD ["nginx", "-g", "daemon off;"]

EXPOSE 80

3d92d4c5112	EXPOSE 80	0B
C8577c27a2ef	CMD ["nginx", "-...	0B
9ee6b6aa5847	RUN apt-get inst...	57.5MB
103ccd6ad90f	RUN apt-get upd...	40.3MB
d2603e1b347d	LABEL maintaine...	0B
ad89def2e29b	FROM ubuntu	80MB

WORKSHOP TIME!

- Go to: https://github.com/IBMDeveloperUK/containers101

SO WHAT NOW?

- Liz Rice – "What is a container, really? Let's write one in Go from scratch"
https://youtu.be/HPuvDm8IC-4

- Ed Shee – "Why Developers Shouldn't Care About Containers"
CloudNativeLondon 2018 [26th-28th Sep]
https://skillsmatter.com/conferences/10160-cloudnative-london-2018#program

- *I want to orchestrate the deployment of my microservices* → Docker-compose

- *I want to learn Kubernetes on my local machine* → Minikube

- *I am thinking of running my mirco-services in production and concerned about HA* → Kubernetes

- *I have micro-services running in Kubernetes and would like to better manage the deployment* → Helm

- *None of the above actually interests me, I just want to code!* → Cloud Foundry

OTHER COOL THINGS

- IBM Coder program: https://developer.ibm.com/code/community/

- Speak to me for IBM Cloud promo codes

- Collect some IBM swag!