Containers 101

Hello!

I AM ED SHEE

Developer Advocate at IBM

You can find me at @ukcloudman

Cloud Landscape

Let’s start with a bit of history...

WHAT DOES CLOUD NATIVE EVEN MEAN?

Orchestration

Monitoring

Distributed Tracing API

Logging

Remote Procedure Call

Container Runtime

Container Runtime

Networking API

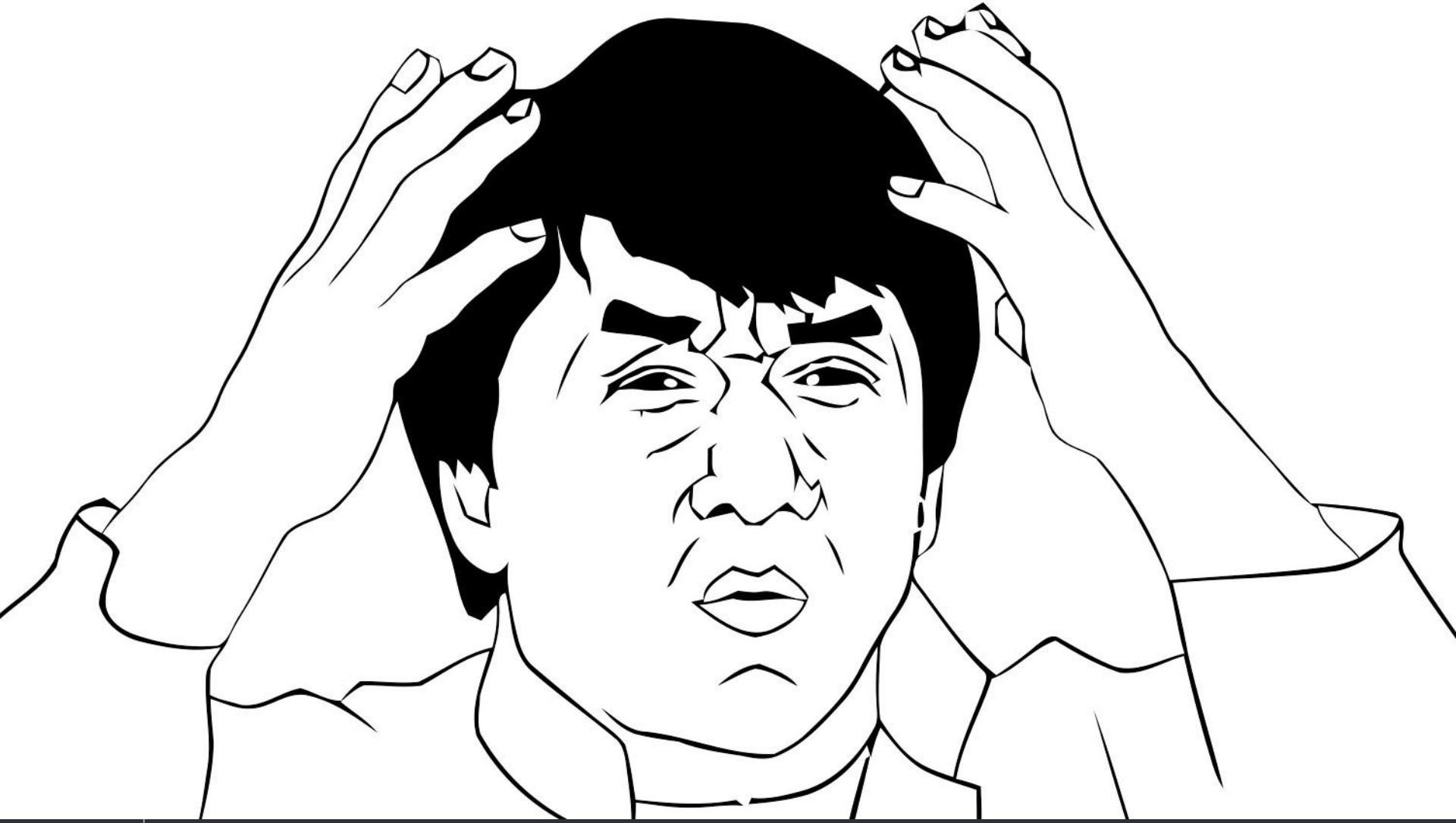
Service Mesh

Distributed Tracing

Security

Software Update Spec

WHAT DOES CLOUD NATIVE EVEN MEAN?



CLOUD COMPUTING IS EVOLVING...

Key driving forces:

- The rise of microservices
- Containerization
- Infrastructure becoming a commodity

12 Factor Applications

Building with cloud platforms in mind

THE 12 FACTOR APP

Codebase

One codebase tracked in revision control, many deploys

Dependencies

Explicitly declare and isolate dependencies

Config

Store config in the environment

Backing Services

Treat backing services as attached resources

Build, Release, Run

Strictly separate build and run stages

Processes

Execute the app as one or more stateless processes

THE 12 FACTOR APP

Port Binding

Export services via port binding

Concurrency

Scale out via the process model

Disposability

Maximize robustness with fast startup and graceful shutdown

Dev/Prod Parity

Keep development, staging, and production as similar as possible

Logs

Treat logs as event streams

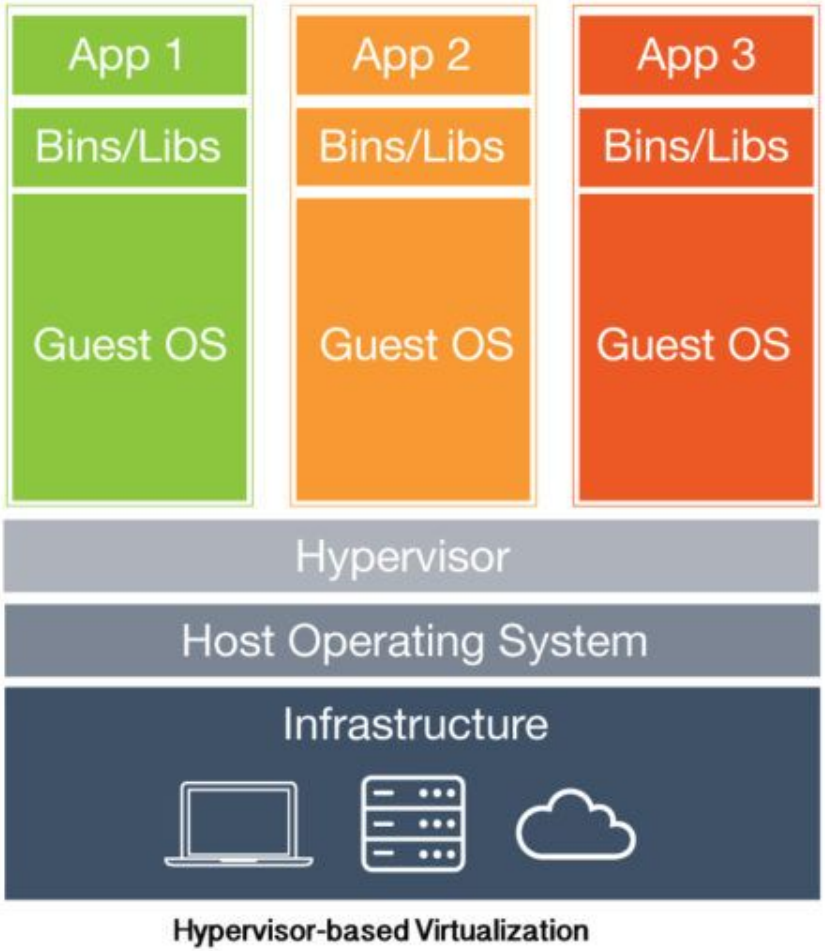
Admin Processes

Run admin/management tasks as one-off processes

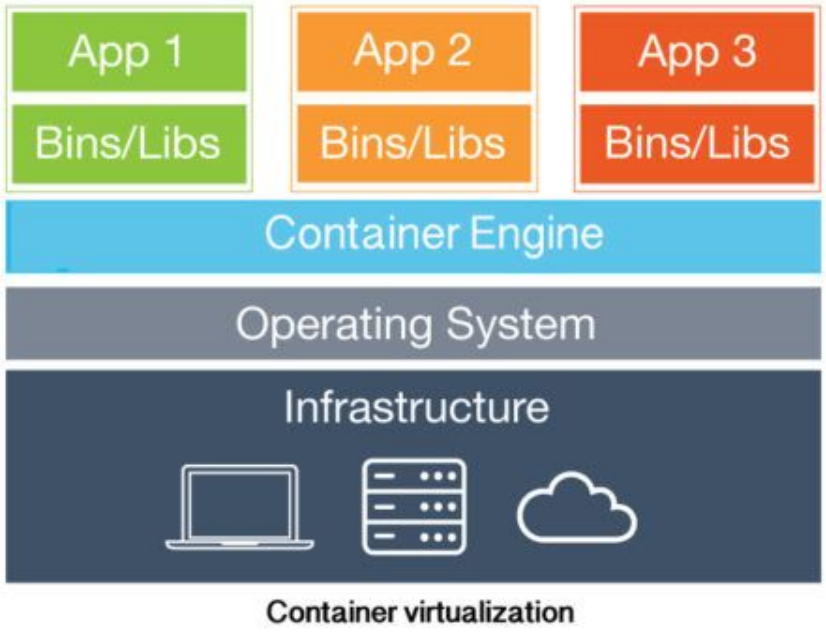
Containers

What's the big deal?

CONTAINER ARCHITECTURE



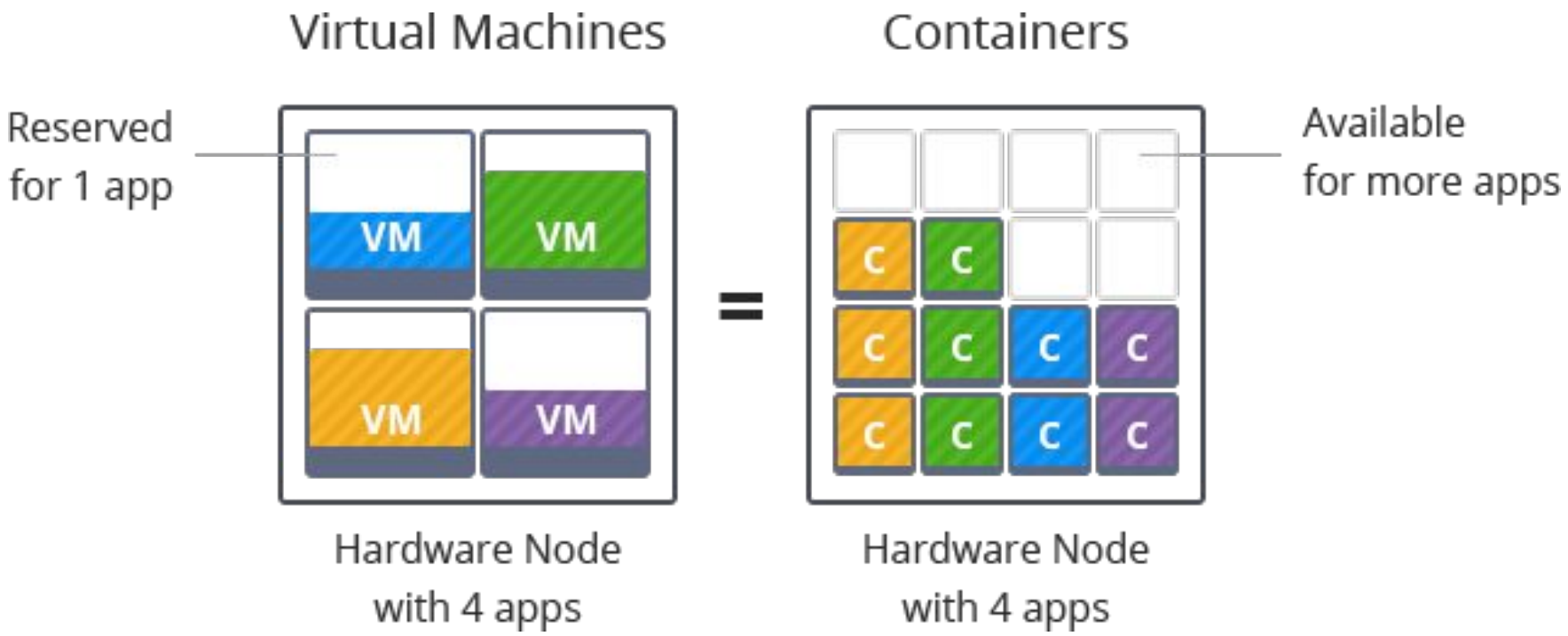
Hypervisor-based Virtualization



Container virtualization

Image credit: https://www.twistlock.com/

CONTAINER ADVANTAGES



WHY IS CONTAINERISATION USEFUL?

Consistency

Application and dependencies packaged in to the container means it will run the same regardless of where it is run.

Speed

Containers can deploy in milliseconds.

Container images are much more lightweight.

Open

Containers are open source and supported on hundreds of clouds.

Build your container once and run it anywhere!

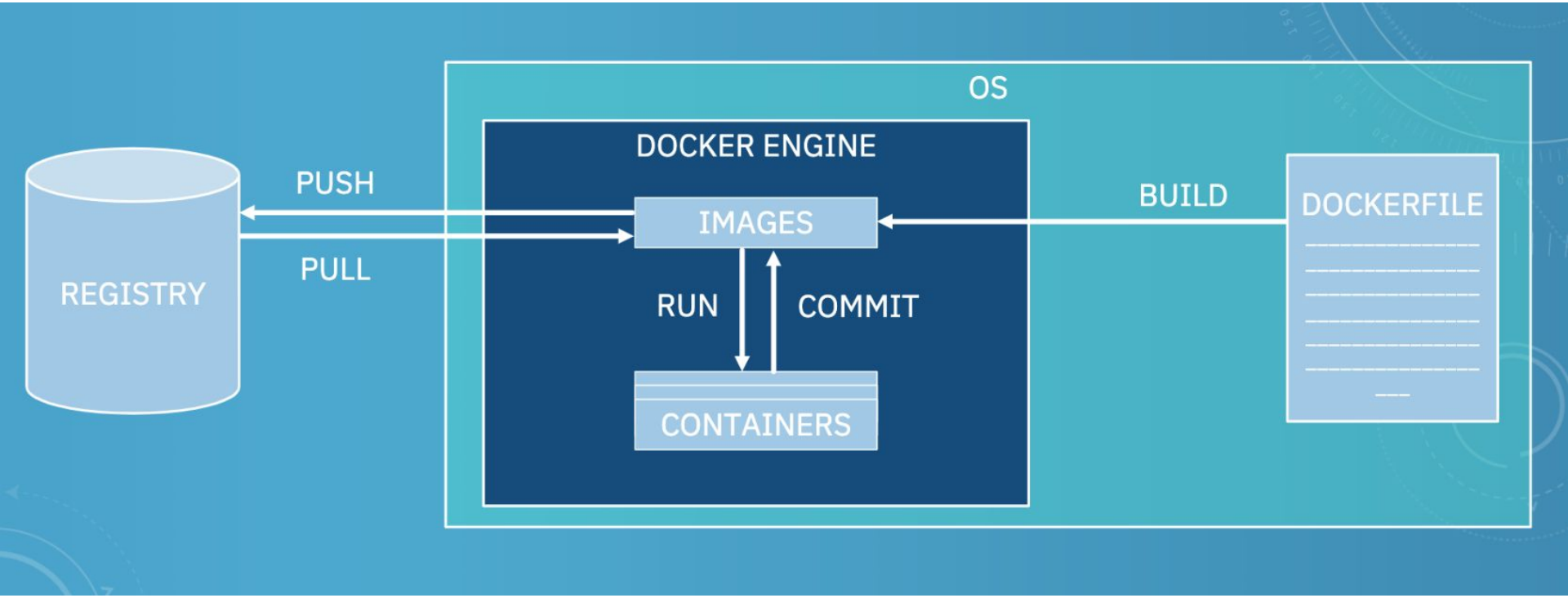
CONTAINER HISTORY

- Unix V7 Chroot - 1979
- Linux VServer - 2001
- Oracle Solaris Containers - 2004
- Open VZ (Virtuzzo) - 2005
- **Google's Process Containers (CGroups) - 2006**
- **LXC (Linux Containers) - 2008**
- CloudFoundry's Warden - 2011
- **Docker - 2013**
- Open Container Initiative - 2015
- Kubernetes - 2015

Docker

Container Lifecycle Management

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
```

DOCKERFILE

```
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

Thanks!

Workshop:

github.com/IBMDeveloperUK/containers101

ANY QUESTIONS?

You can find me at

github: IBMDeveloperUK

Twitter: @ukcloudman

Email: edmundshee@uk.ibm.com