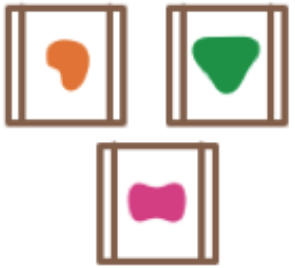
Docker, Kubernetes & Java Microservices

JHUG event, November 2017 @ Eurobank

Microservices

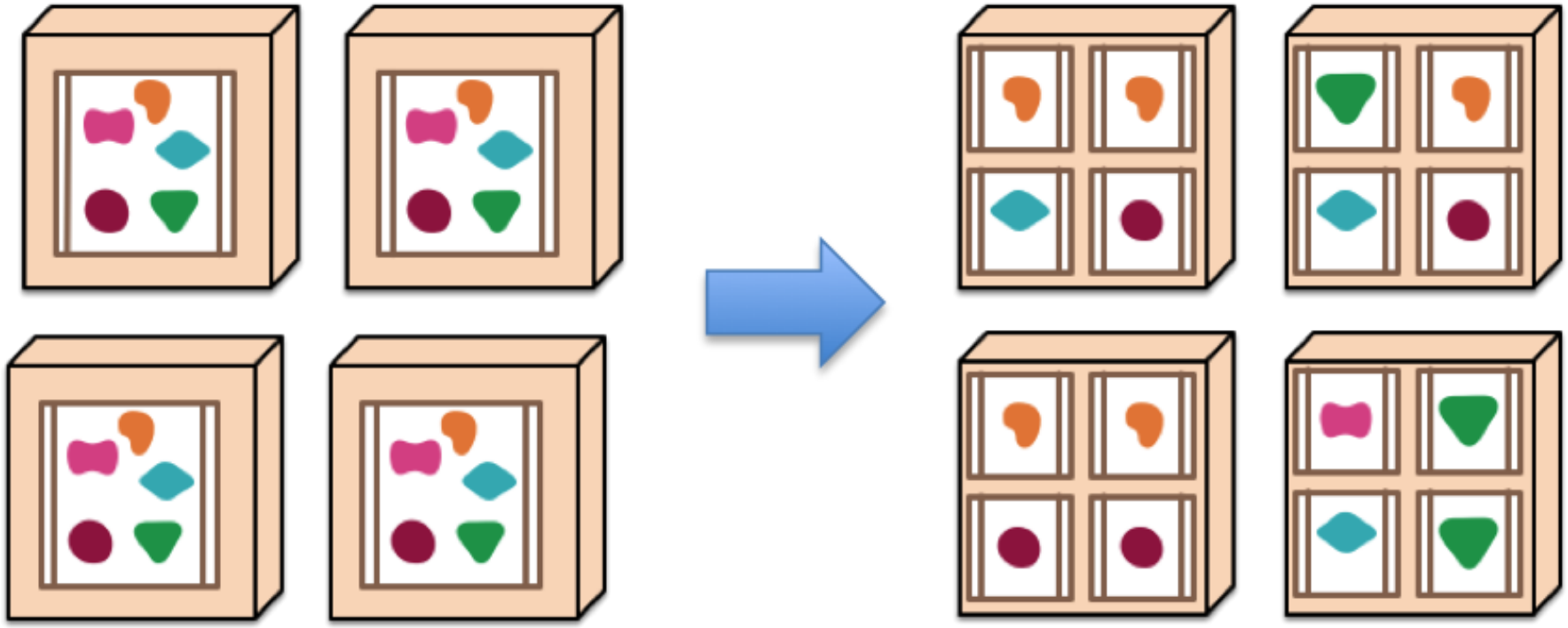
A monolithic application puts all its functionality into a single process...

A microservices architecture puts each element of functionality into a separate service...



... and scales by replicating the monolith on multiple servers

... and scales by distributing these services across servers, replicating as needed.

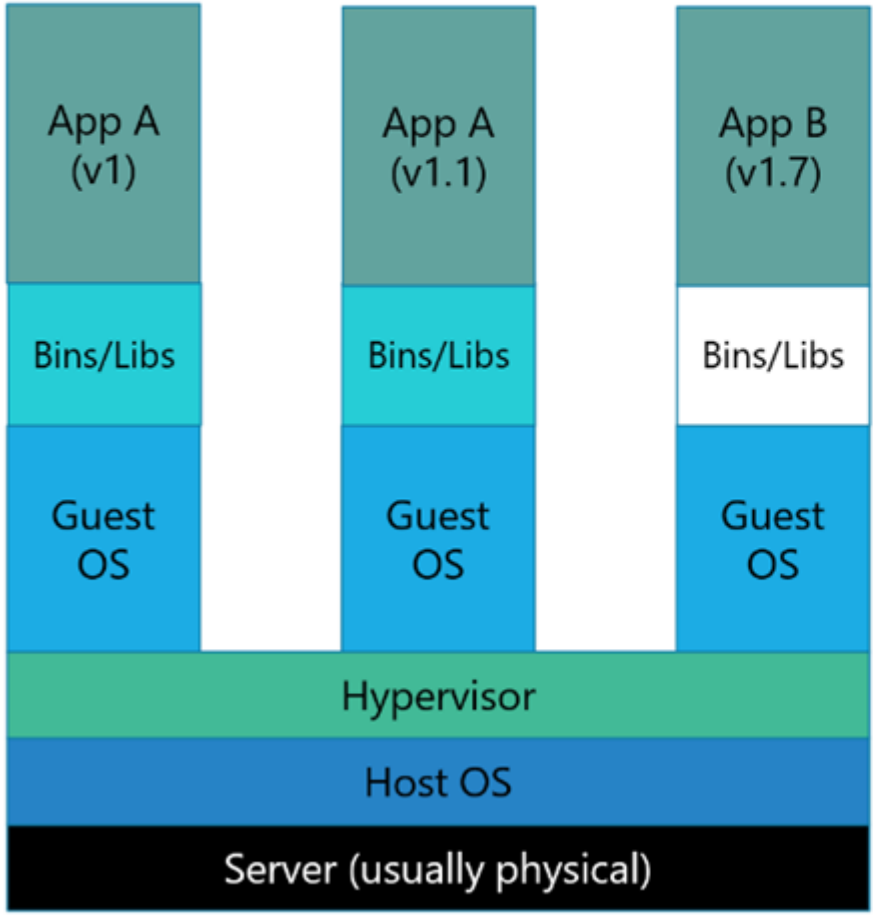


Source: martinfowler.com

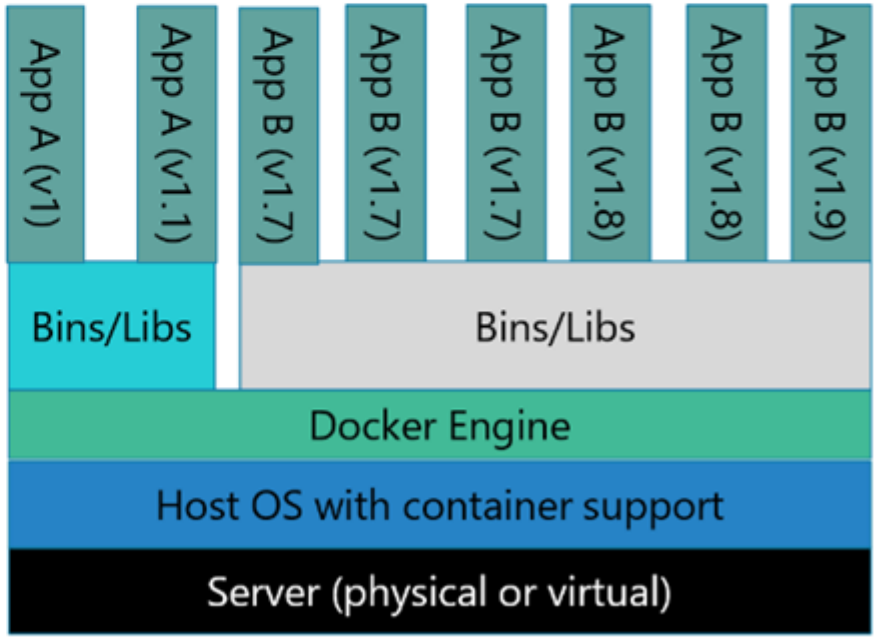
What are containers?

- Based on Linux containers technology (since 2008)
- A package of binaries, libraries, dependencies
- Runs as an isolated process in a host OS
- OS-level virtualization

Server Virtualisation: Each app and each version of an app has dedicated OS



Containers: All containers share host OS kernel and appropriate bins/libraries



Key container advantages

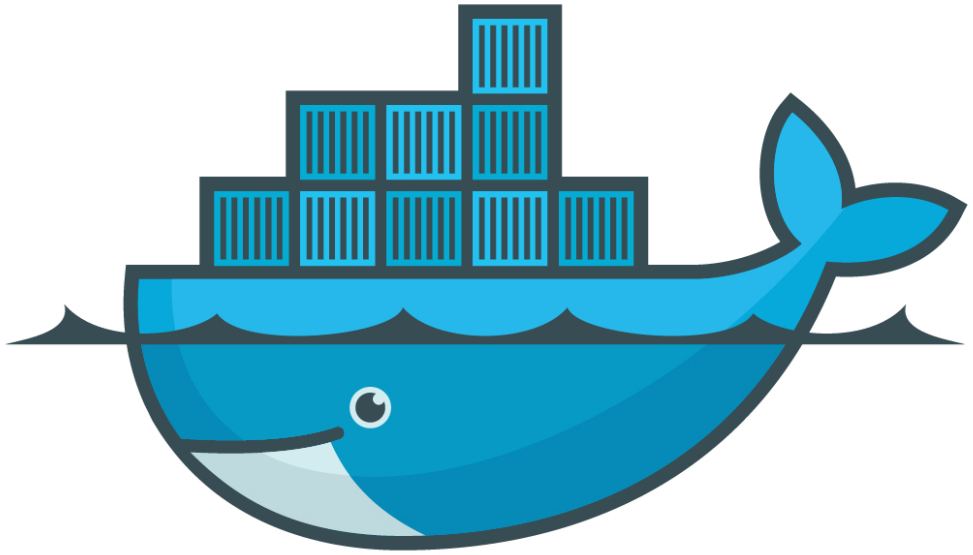
- Small footprint

  (Higher density and utilization of resources)

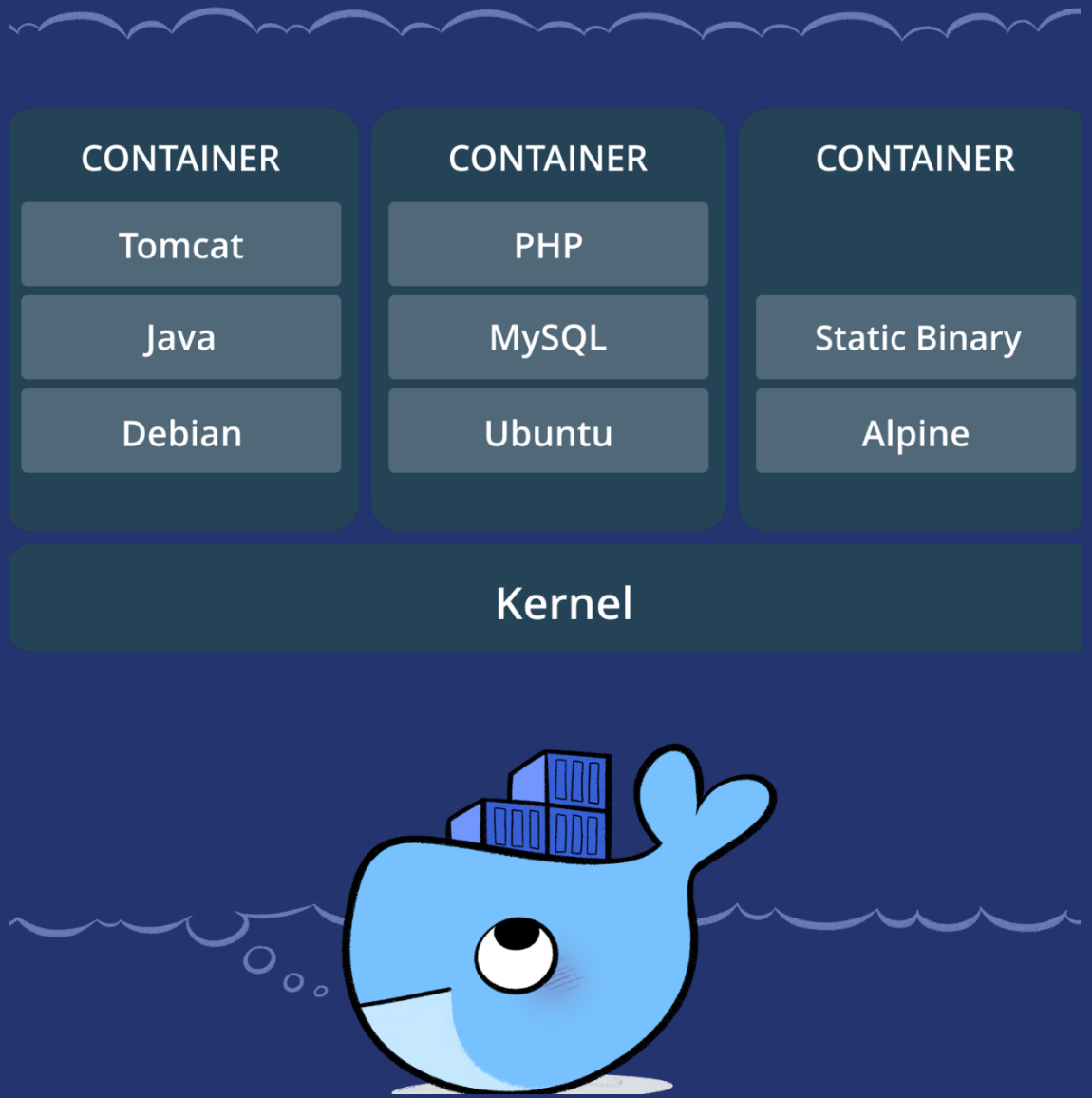
- Fast startup

- Portability

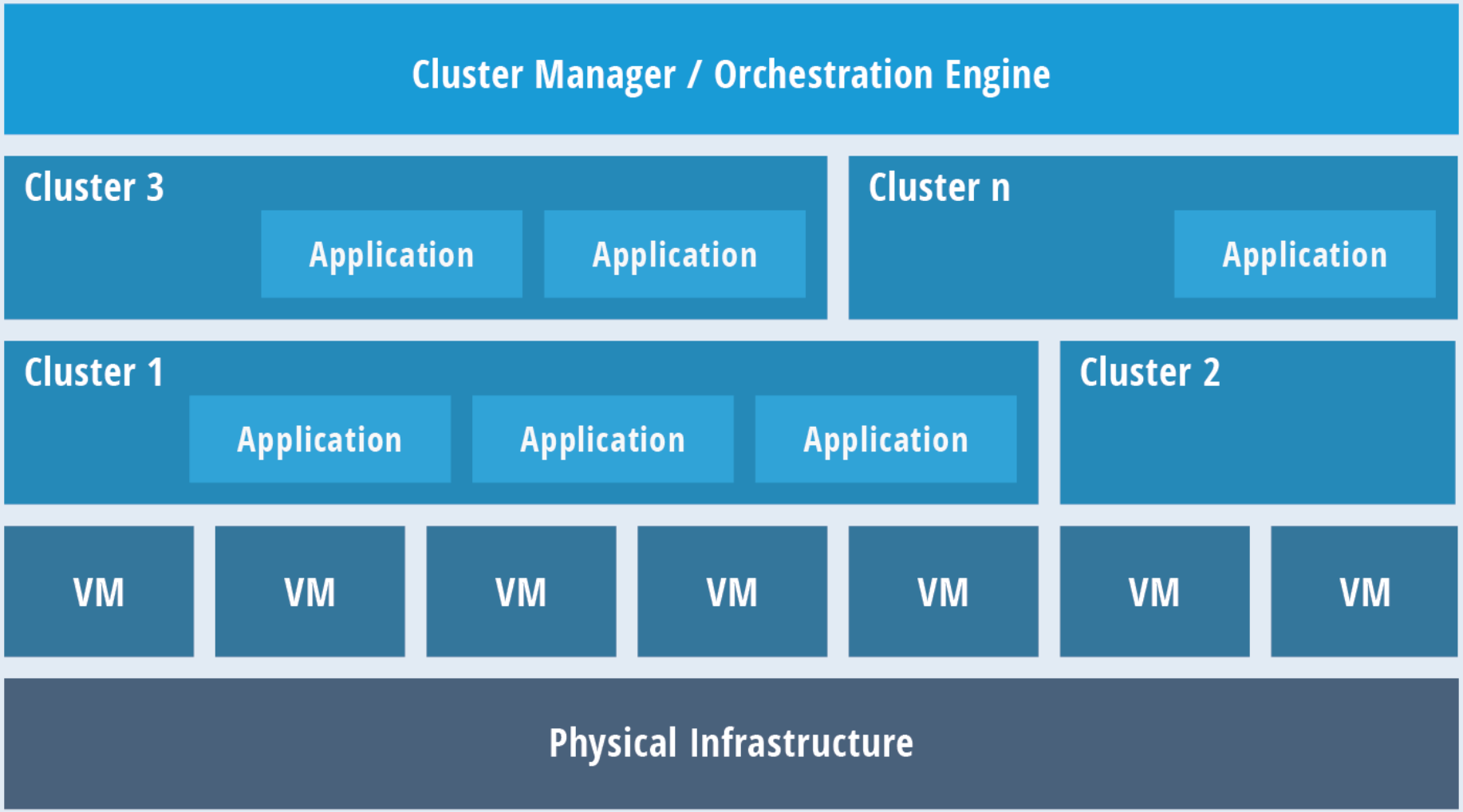
  (No more dependencies nightmare)



docker


CONTAINER
Tomcat
Java
Debian
CONTAINER
PHP
MySQL
Ubuntu
CONTAINER
Static Binary
Alpine
Kernel

Container Orchestration Engine



Source: Janakiram MSV

THENEWSTACK

Kubernetes

κυβερνήτης: *Greek for "pilot" or "helmsman of a ship"*

the open source cluster manager from Google

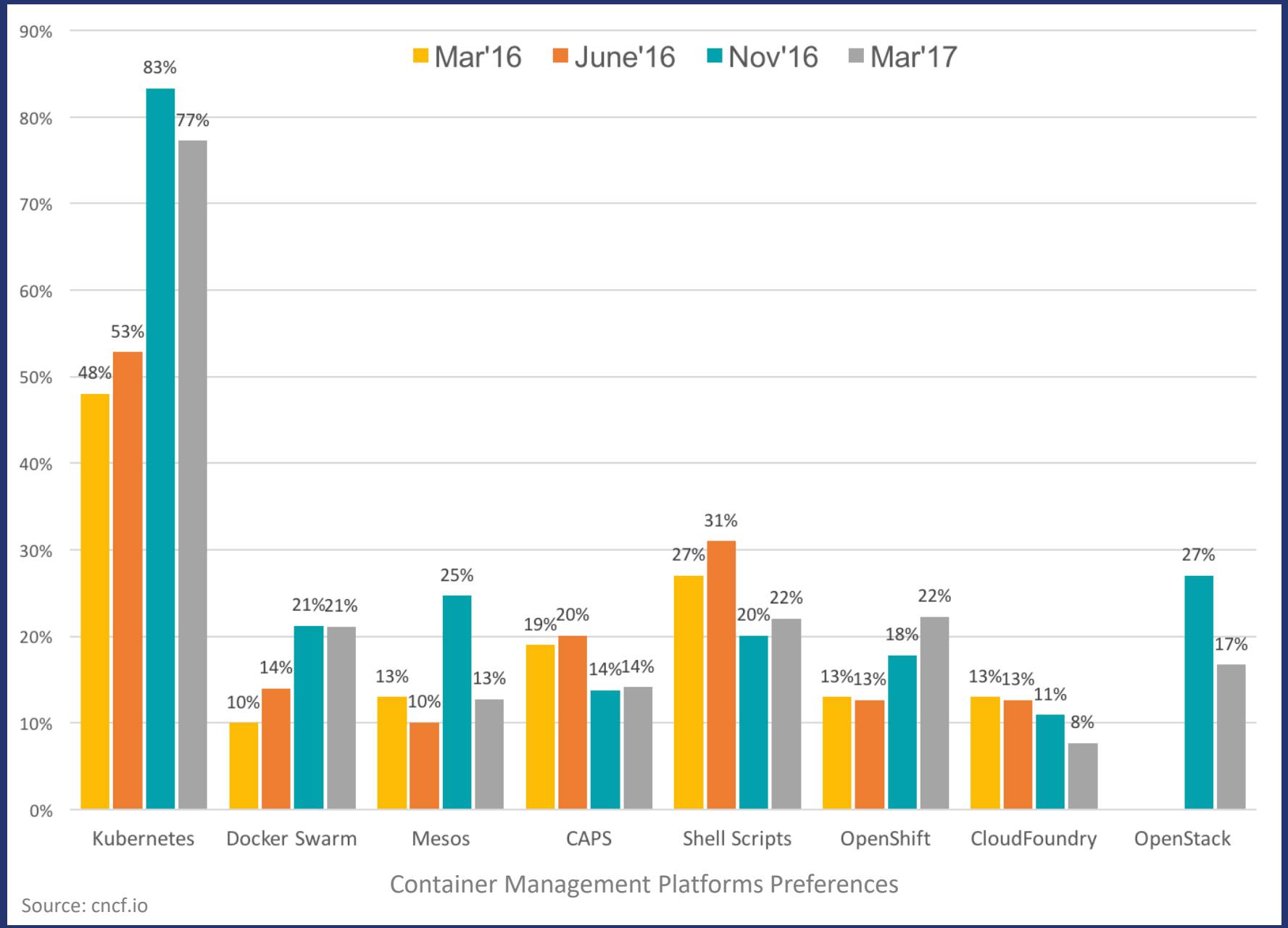
Everything at Google runs on containers

Gmail, Web Search, Maps, …
MapReduce, batch, …
GFS, Colossus, …
Google's Cloud Platform: VMs run in containers!

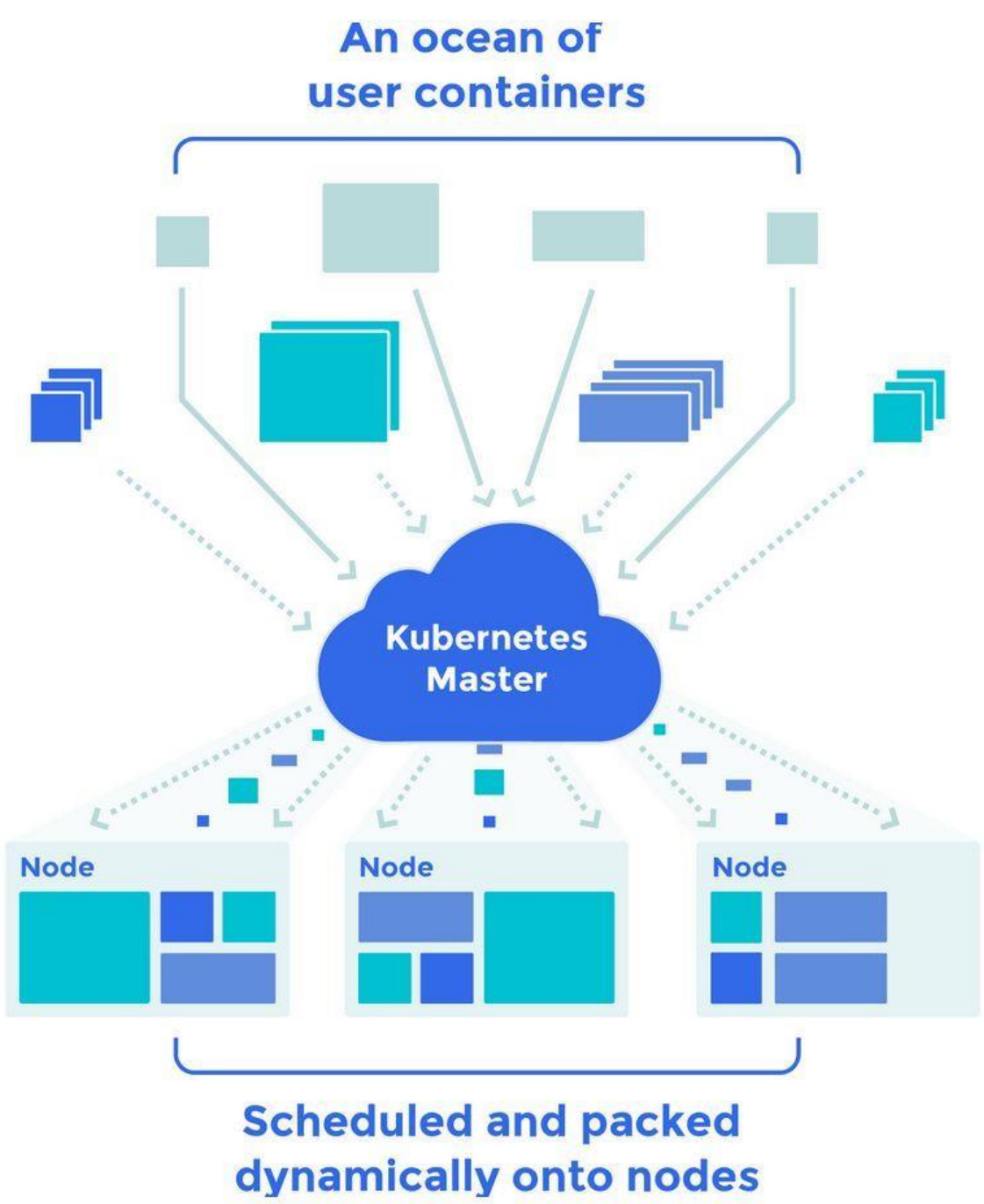
"We launch over **2 billion** containers per week"

2014, Joe Beda

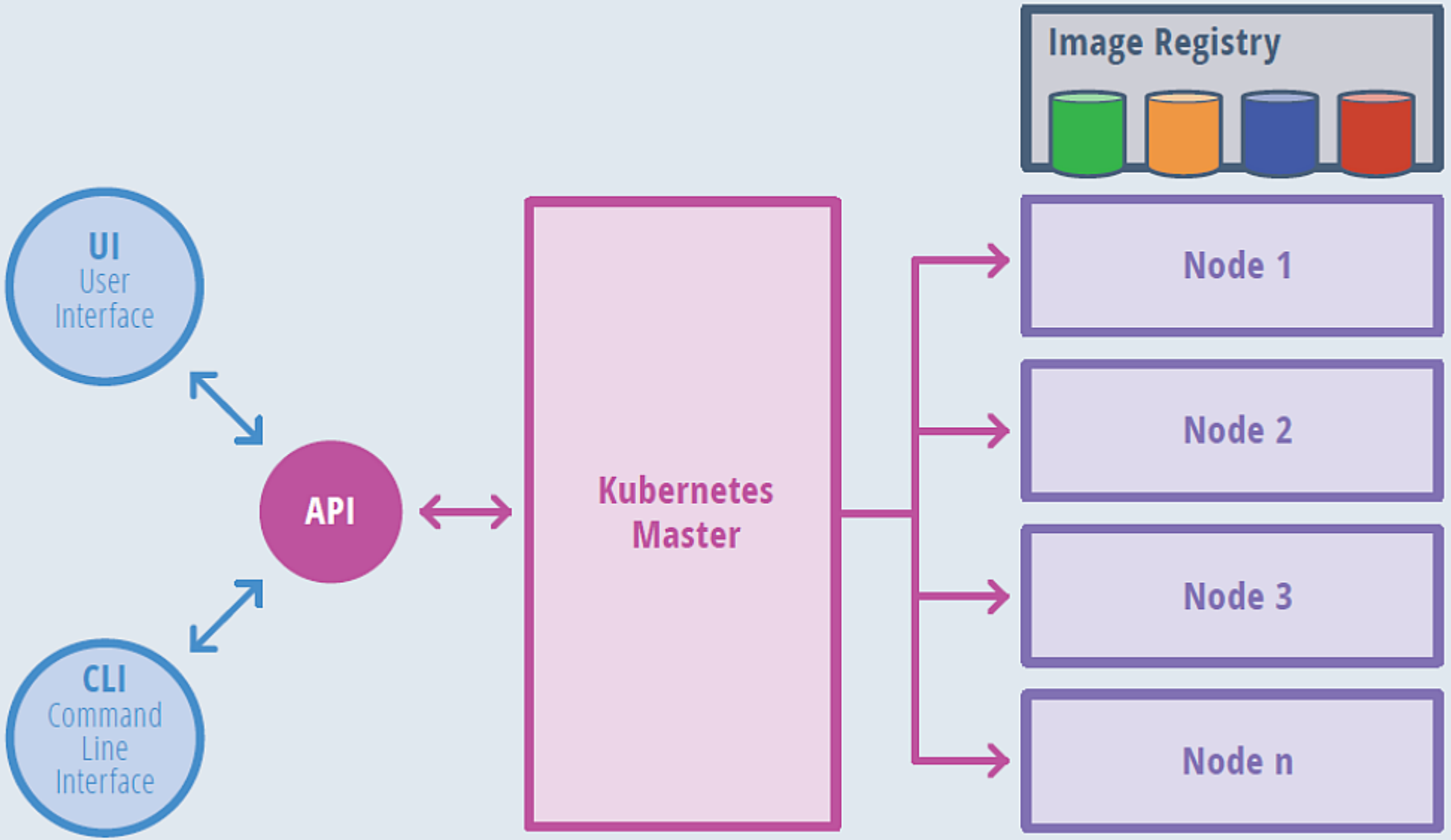
The Evolving Landscape of Cloud Management Platforms




KUBERNETES


An ocean of
user containers
Kubernetes
Master
Node
Node
Node
Scheduled and packed
dynamically onto nodes

Kubernetes Architecture



Source: Janakiram MSV

THENEWSTACK

Sample Kubernetes manifest file

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.8
        ports:
        - containerPort: 80
```

Kubernetes main task

To bring the current state
to the (declared) desired state

Kubernetes best features

- Self-healing
  (auto-placement, auto-replication, auto-scaling, auto-restart)
- Effective use of hardware resources
- Infrastructure abstraction

Part 2

Running Java Microservices on Docker & Kubernetes

Choose the proper docker base image

- Openjdk:slim (244MB instead of 700+MB)
- Exit on OOM (so that Kubernetes restarts the pod)
  - see also JvmKill project (https://github.com/cloudfoundry/jvmkill)
- Recent Java 8 versions and 9 pickup cgroups environment CPU limits
  - use `UseCGroupMemoryLimitForHeap` experimental option to pickup memory limits, too
  - use `MaxRAMFraction=1` option to use all available memory

Enable monitoring endpoints on spring-boot

- Use Spring Boot in the first place 😃
- Include starter-actuator module to have standard metrics endpoints exposed
  - Spring cloud, integration etc. also add metrics
- Setup bridge to expose metrics to JMX - see: `MetricsEndpointMetricReader`
- Include jolokia-core to expose jolokia JMX HTTP endpoint
- Use hawt.io or heapster to monitor microservices state

Buy memory for developer workstation(s)

- 10+ Spring Boot based microservices with docker based supporting services (e.g. redis, rabbitmq) need 16GB RAM
- Use IntelliJ IDEA - bite the bullet and leave Eclipse

Check out spring-cloud family of libraries

- Config server is a fast and traceable alternative to environmental variable based configuration offered by Kubernetes - supports multiple environments (DEV, TEST, PROD etc.) as well as specific overrides (e.g. feature branches)
- Hystrix is a circuit breaker that can detect and stop cascading failures due to lower level service unavailability
- Sleuth is a distributed tracing library, based on Google's Dapper paper
- More added with each release!

Go on-prem as soon as possible if planned for production

- Public cloud infrastructure, apart from dedicated (expensive!) machines, is SLOW
- Ops require time to get acquainted with new way of doing things
- CI/CD pipelines are setup in a different way, even if base platform is the same on-prem and public cloud

Involve security from the beginning

- GDPR imposes strict rules, especially for public cloud infrastructure

Explore new ways to be productive

- Kotlin is officially supported in Spring Boot 2
- Kubernetes platforms often offer productivity shortcuts (e.g. source-to-image on Openshift)
- Spock based TDD/BDD development offers a descriptive, fluent way to write unit/integration tests compared with assertj and spring-test

Thank you