

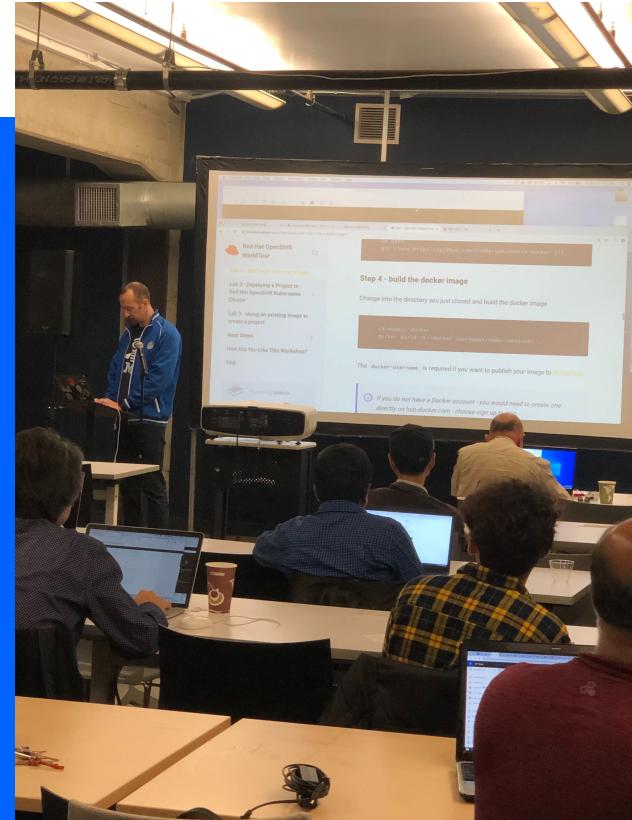
How to Run Kubernetes with OpenShift on IBM Cloud

Marek Sadowski

Developer Advocate | IBM San Francisco City Team

ibm.biz/202101-kubernetes-instructions

V1 2021 01 21



IBM Developer

IBM

Agenda:

1. microservices in a cloud
2. containers to host microservices
3. Docker to build-ship-run containers
4. orchestration of many containers
5. Kubernetes
6. Helm
7. Istio
8. OKD
9. Red Hat OpenShift vs Kubernetes
10. Red Hat OpenShift hands-on

THE POLL

How you build a microservice

IT IS A SIMPLE PROGRAM – RESPONDING TO CRUD OPERATIONS

WHAT is a microservice?

Why are there microservices?

A large system might consist of many services: HR, a basket checkout, payments, CRM operations, etc.

These services might be built out of microservices – change the prescription's content: add, remove, update, delete

- These are microservices – that might be deployed as containers

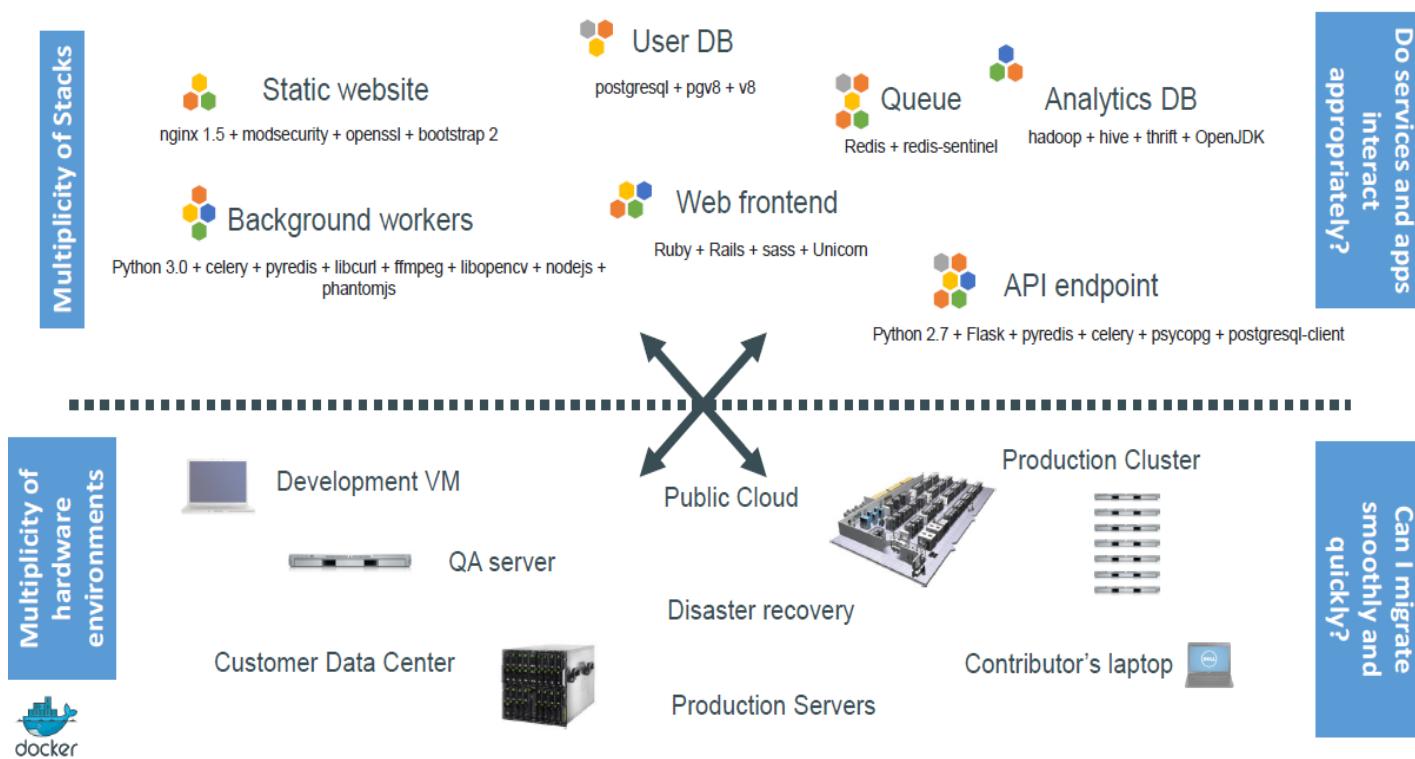
Why Containers?

Everyone Loves Containers

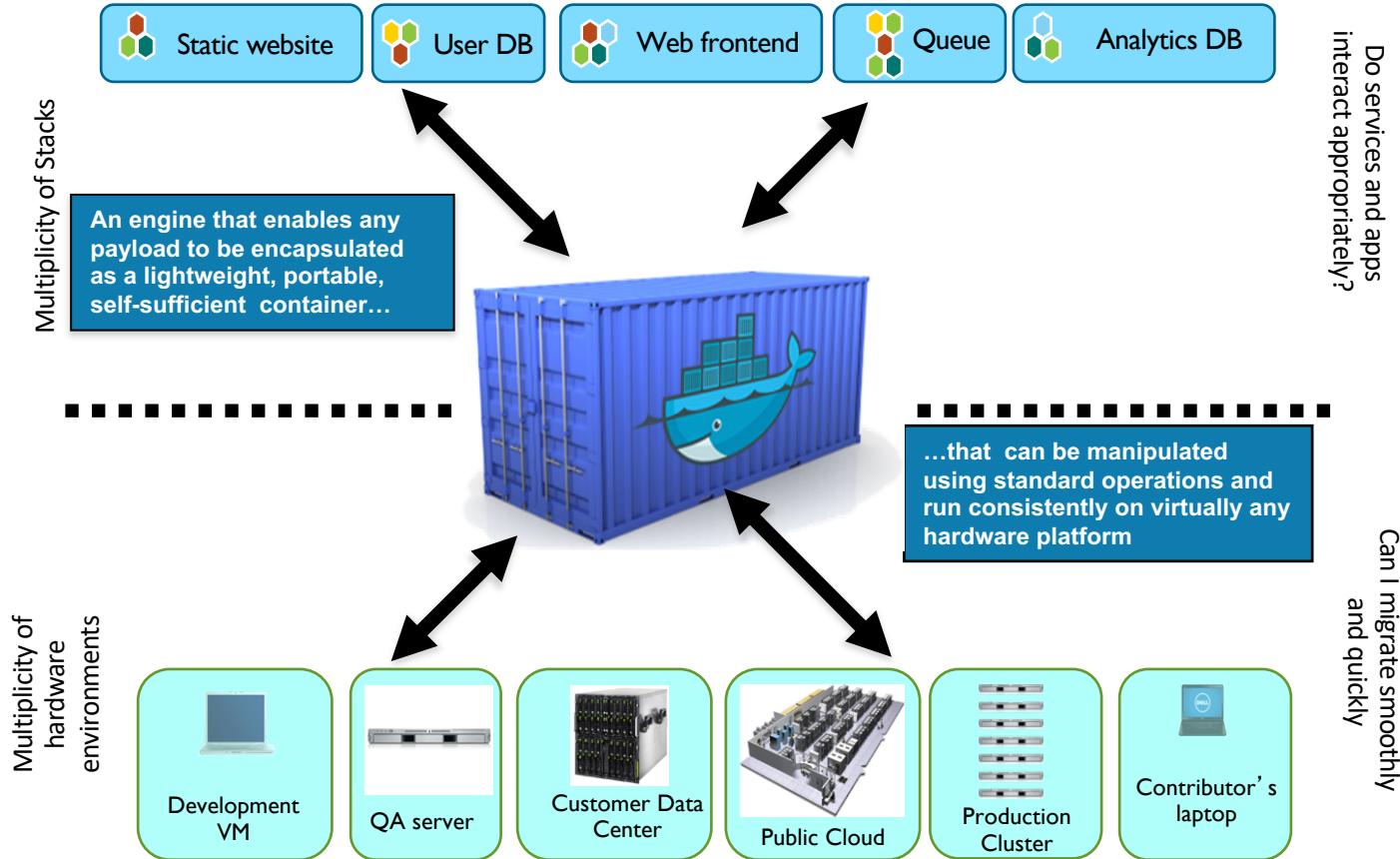


- A standard way to package an application and all its dependencies so that it can be moved between environments and run without changes
- Containers work by isolating the differences between applications inside the container so that everything outside the container can be standardized

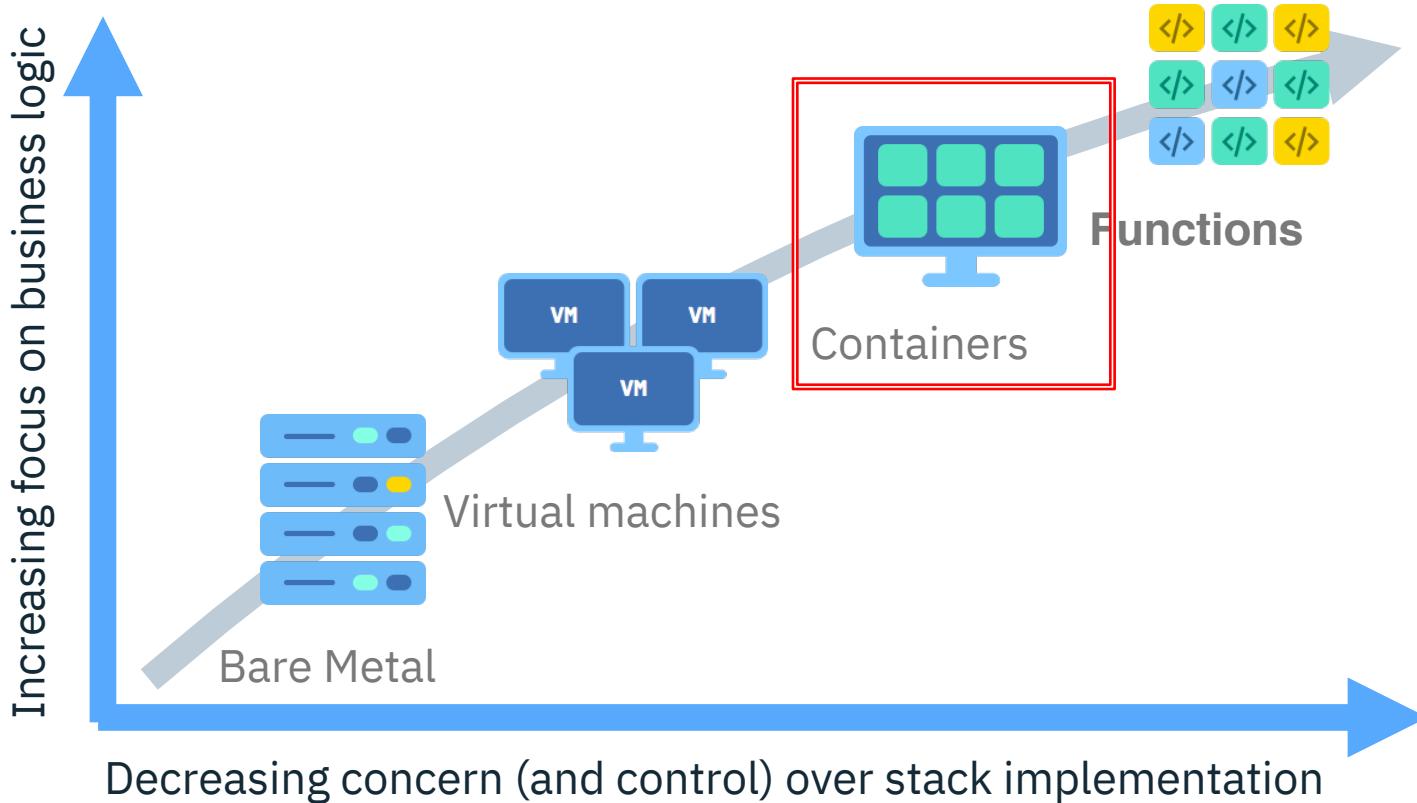
The Challenge



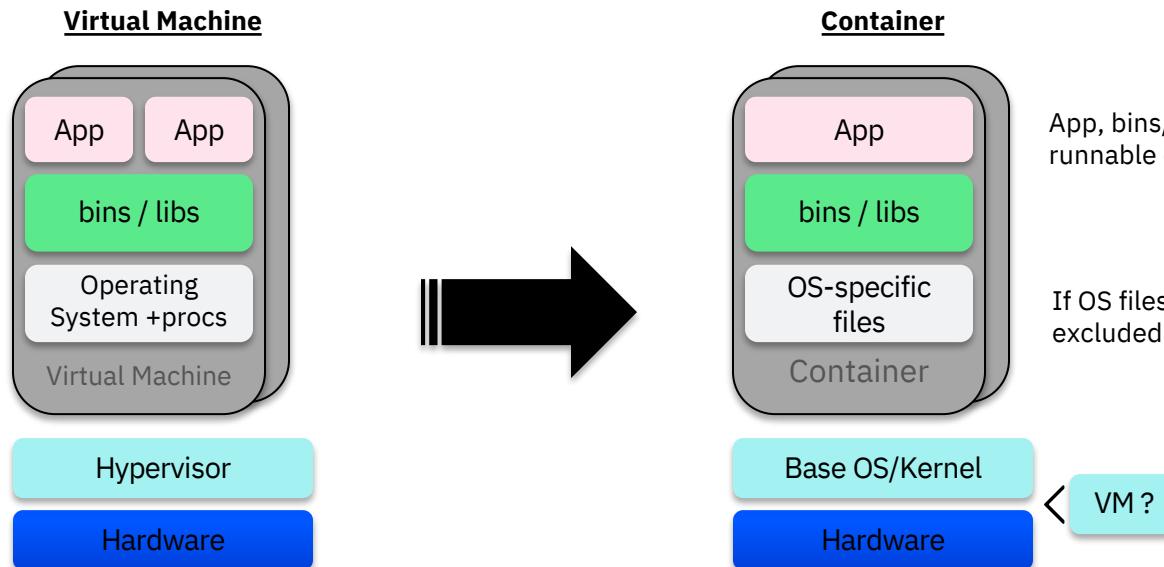
The Solution - A Shipping Container for Code



How we got here



VM vs Container



Each VM has its own
OS

Containers share
the same base
Kernel

App, bins/libs/OS must all be
runnable on the shared kernel

If OS files aren't needed they can be
excluded.

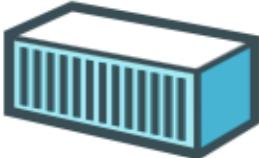
VM ?

Docker Mission

Docker is an **open platform** for building distributed applications for **developers** and **system administrators**



Build



Ship



Run



IBM Code



Any App

Anywhere

Dev vs. Ops

Dev

Ops



- Code
- Libraries
- Configuration
- Server runtime
- OS

- Logging
- Remote access
- Network configuration
- Monitoring

Separation of concerns

A container separates and bridges the Dev and Ops in DevOps

- Dev focuses on the application environment
- Ops focuses on the deployment environment

Docker Component Overview

Docker Engine

- Manages containers on a host
- Accepts requests from clients
 - REST API
- Maps container ports to host ports
 - E.g. 80 → 3582

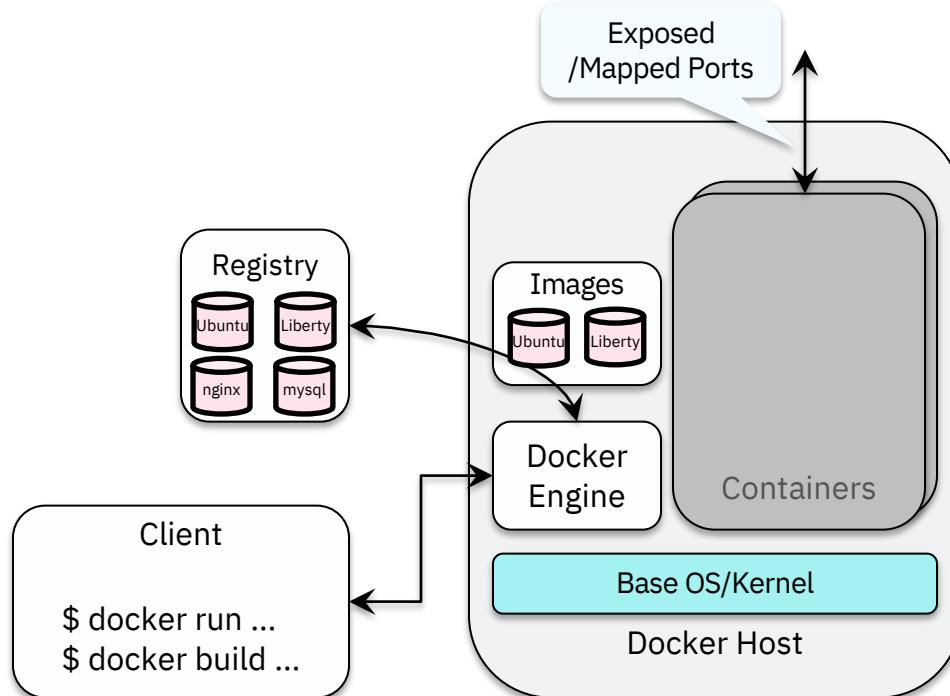
Images

Docker Client

- Drives engine
- Drives "builder" of Images

Docker Registry

- Image DB



Shared / Layered / Union Filesystems

Docker uses a copy-on-write (union) filesystem

New files(& edits) are only visible to current/above layers

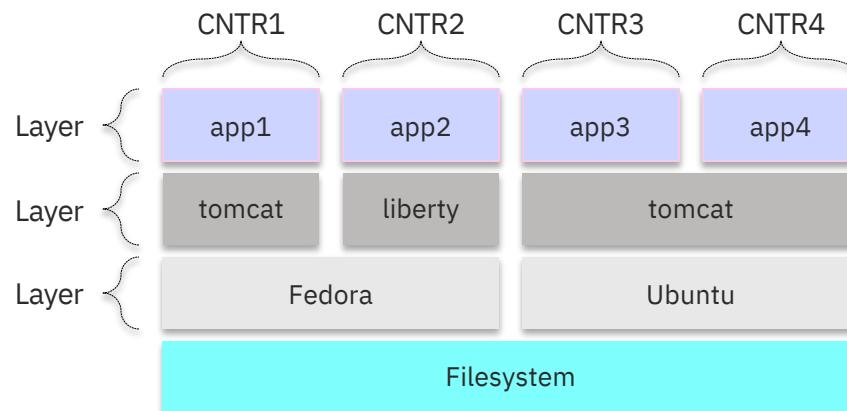
Layers allow for reuse

- More containers per host
- Faster start-up/download time

Images

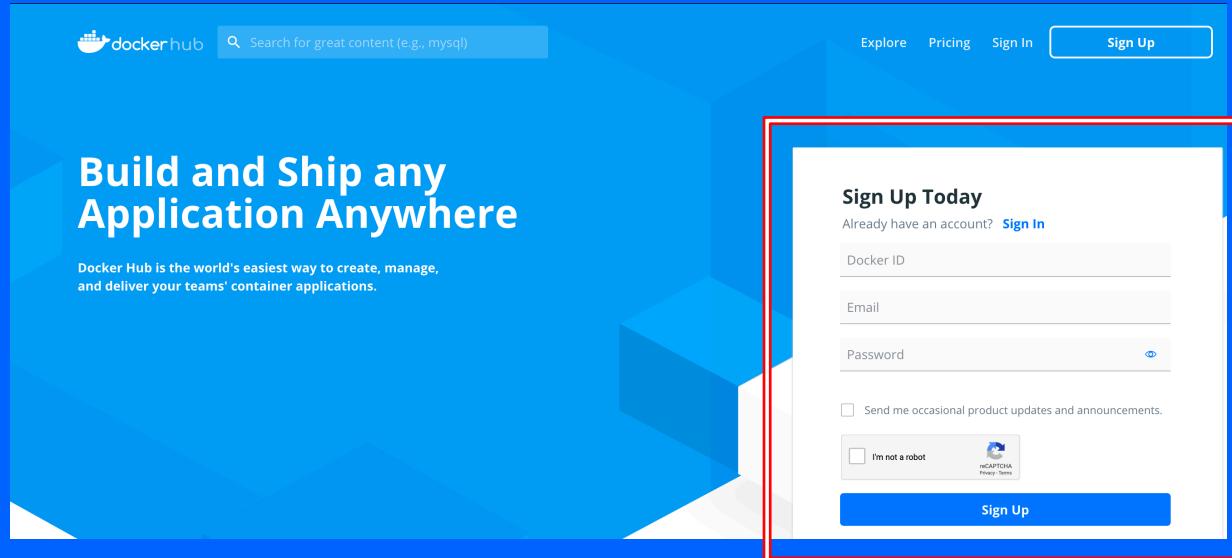
- Tarball of layers

Think: Transparencies on projector



Hands-on

Download FREE Docker Desktop and use your terminal on your computer or just sit back and watch



Our First Container

```
$ docker run ubuntu echo Hello World
```

```
Hello World
```

What happened?

Docker created a directory with a "ubuntu" filesystem (image)

Docker created a new set of namespaces

Ran a new process: echo Hello World

Using those namespaces to isolate it from other processes

Using that new directory as the "root" of the filesystem (chroot)

That's it!

Notice as a user I never installed "ubuntu"

Run it again - notice how quickly it ran

```
[Mareks-MBP:~ mareksadowski$ docker run ubuntu echo Hello World
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
6b98dfc16071: Pull complete
4001a1209541: Pull complete
6319fc68c576: Pull complete
b24603670dc3: Pull complete
97f170c87c6f: Pull complete
Digest: sha256:5f4bdc3467537cbbe563e80db2c3ec95d548a9145d64453b06
Status: Downloaded newer image for ubuntu:latest
Hello World
[Mareks-MBP:~ mareksadowski$ docker run ubuntu echo Hello World
Hello World
Mareks-MBP:~ mareksadowski$ ]
```

LAB
building a translation service with Watson:

<http://ibm.biz/202101-kube-docker>

IBM Cloud Catalog

<https://cloud.ibm.com/catalog>

The screenshot shows the IBM Cloud Catalog interface. At the top, there's a navigation bar with links for Catalog, Docs, Log in, and Sign up. Below the navigation is a search bar with the placeholder "Search the catalog...". A horizontal menu bar has "Services" selected (highlighted in blue) and "Software". To the left, a sidebar lists various service categories: All Categories, VPC Infrastructure, Compute, Containers, Networking, Storage, AI, Analytics, Databases, Developer Tools, Integration, Internet of Things, and Security and Identity. On the right, the main content area is titled "Services" and describes the broad portfolio of managed services. A "Featured" section highlights the "Kubernetes Service" with its icon, name, and a brief description: "Deploy secure, highly available apps in a native Kubernetes experience." At the bottom of the page, there are links for "Free • IAM-enabled • Service Endpoint Supported". A vertical "FEEDBACK" button is located on the far right edge.

Second part –
the orchestration
of successful API deployment

Containers



Everyone's container journey starts with one container....

Containers



At first the growth is easy to handle....



Containers



But soon it is overwhelming... chaos reigns

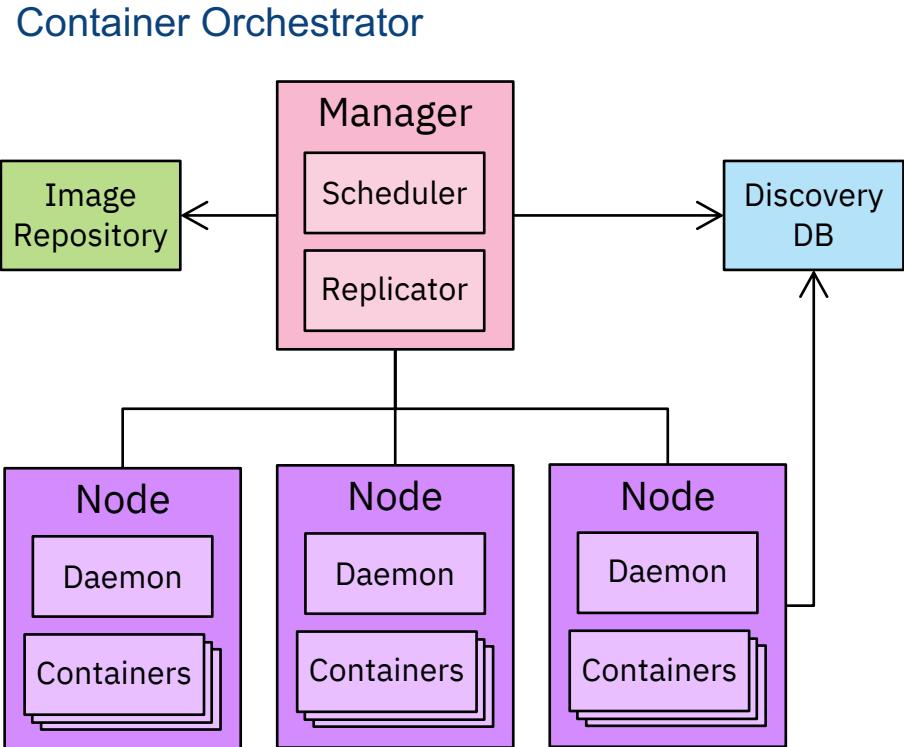
Container Orchestration

Allows users to define how to coordinate the containers in the cloud when the multi-container packaged application is deployed.

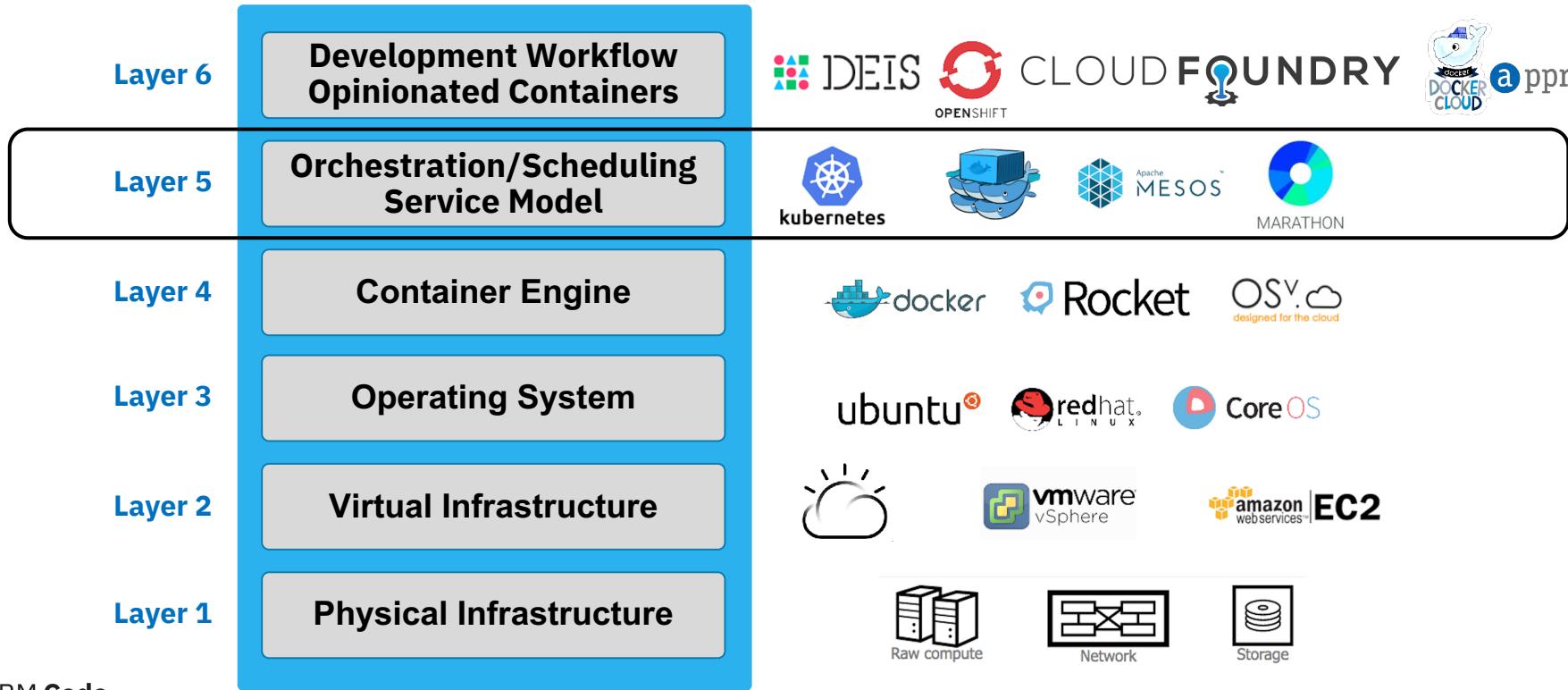
- Scheduling
- Cluster management
- Service discovery
- Provisioning
- Monitoring
- Configuration management

What is Container Orchestration?

- Container orchestration
 - Cluster management
 - Scheduling
 - Service discovery
 - Replication
 - Health management



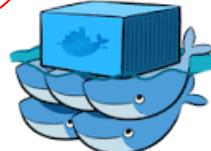
Container Ecosystem Layers



Kubernetes – Mesos – Marathon – Docker Swarm



Kubernetes is a cluster manager for containers (large deployments)



Swarm has a YAML-based deployment model, auto-healing of clusters, overlay networks with DNS, high-availability through the use of multiple masters, and network security using TLS with a Certificate Authority (1-10 containers)



MESOS is a distributed system kernel that will make your cluster look like one giant computer system to all supported frameworks and apps that are built to be run on mesos.

(Kubernetes can be run on Mesos)



Marathon is a cluster-wide init and control system for running Linux services in cgroups and Docker containers, and it is run on top of Mesos

What is Kubernetes?

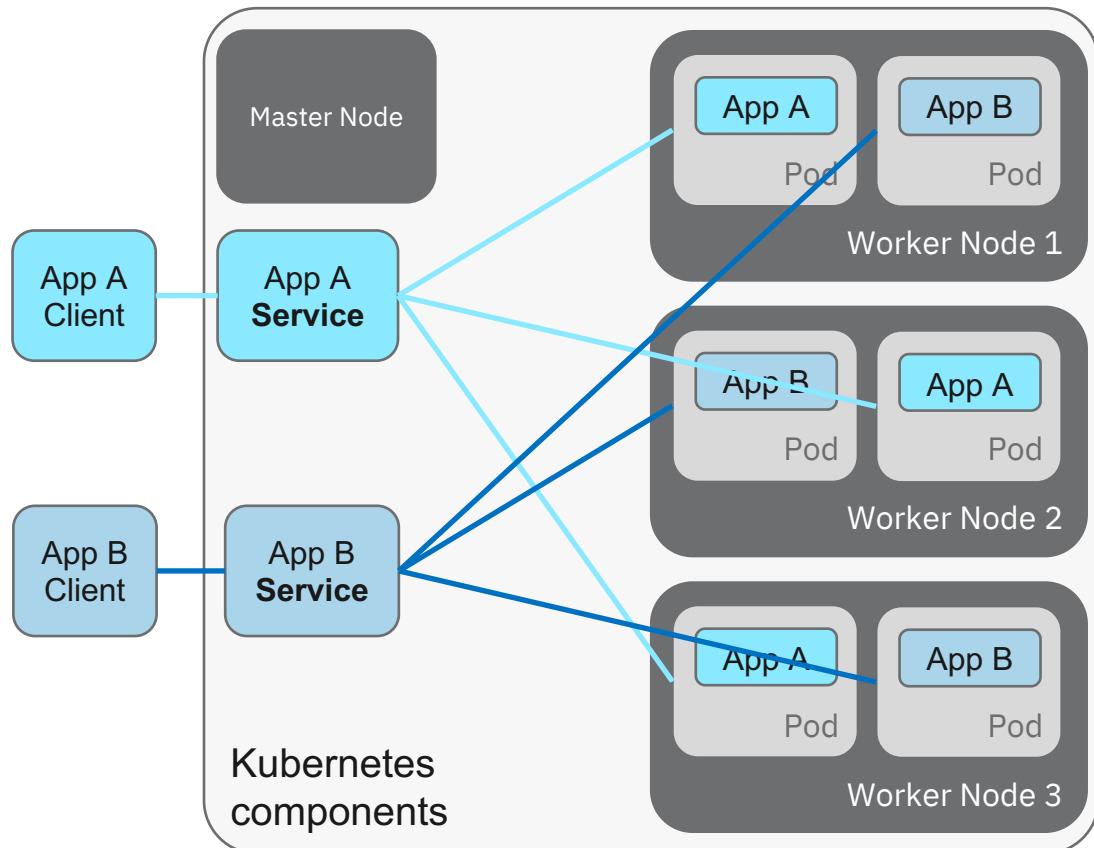


- Container orchestrator
- Manage applications, **not machines**
- Designed for extensibility
- Open source project managed by the Linux Foundation



Kubernetes Architecture: Workloads

- Container
 - Packaging of an app
- Pod
 - Unit of deployment
- Service
 - Fixed endpoint for 1+ pods



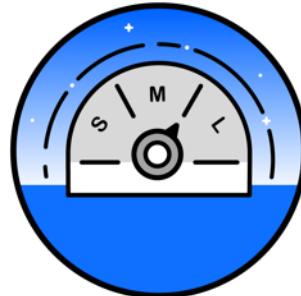
Kubernetes



Intelligent Scheduling



Self-healing



Horizontal scaling



Service discovery & load balancing



Automated rollouts and rollbacks



Secret and configuration management

Helm & Helm Chart

- Helm
 - Package manager for Kubernetes
 - Used to manage Kubernetes applications
- Helm Chart
 - Used to define, install, and upgrade complex Kubernetes applications
 - Easy to create, version, share and publish
 - Expressed in “Yet Another Markup Language” (YAML) files



Challenges with Microservices

Security

Canary deployments

A/B testing

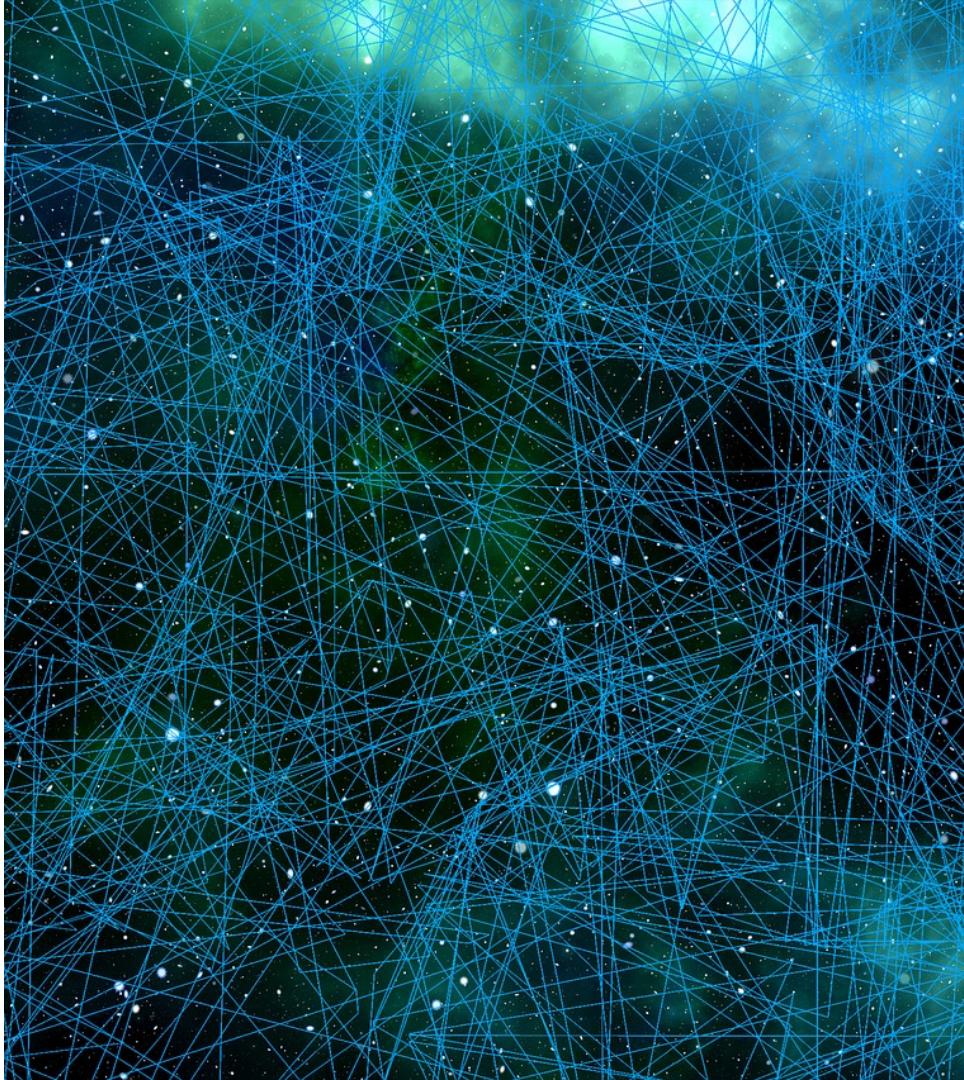
Retries and Circuit breaking

Rate limiting

Fault injection

Policy management

Telemetry



What is a ‘Service Mesh’ ?

A network for services, not bytes

- Observability
- Resiliency
- Traffic Control
- Security
- Policy Enforcement



Istio



An open platform to connect, manage, and secure microservices

<http://istio.io>

Istio community partners

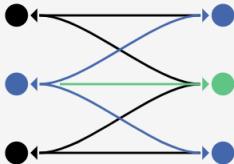
- RedHat
- Pivotal
- WeaveWorks
- Tigera
- Datawire
- Scytale (SPIFFE)
- Microsoft
- Uber (Jaeger)





Istio

Connect, secure, control, and observe services.



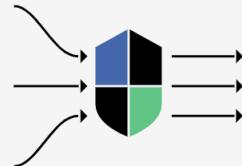
Connect

Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments.



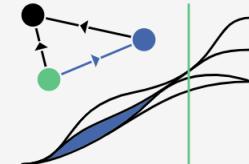
Secure

Automatically secure your services through managed authentication, authorization, and encryption of communication between services.



Control

Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers.

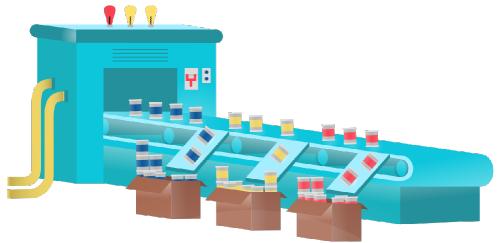


Observe

See what's happening with rich automatic tracing, monitoring, and logging of all your services.

WHERE ISTIO FINDS ITS PLACE

Intelligent
Routing and
Load Balancing



Resiliency
across
Languages and
Platforms

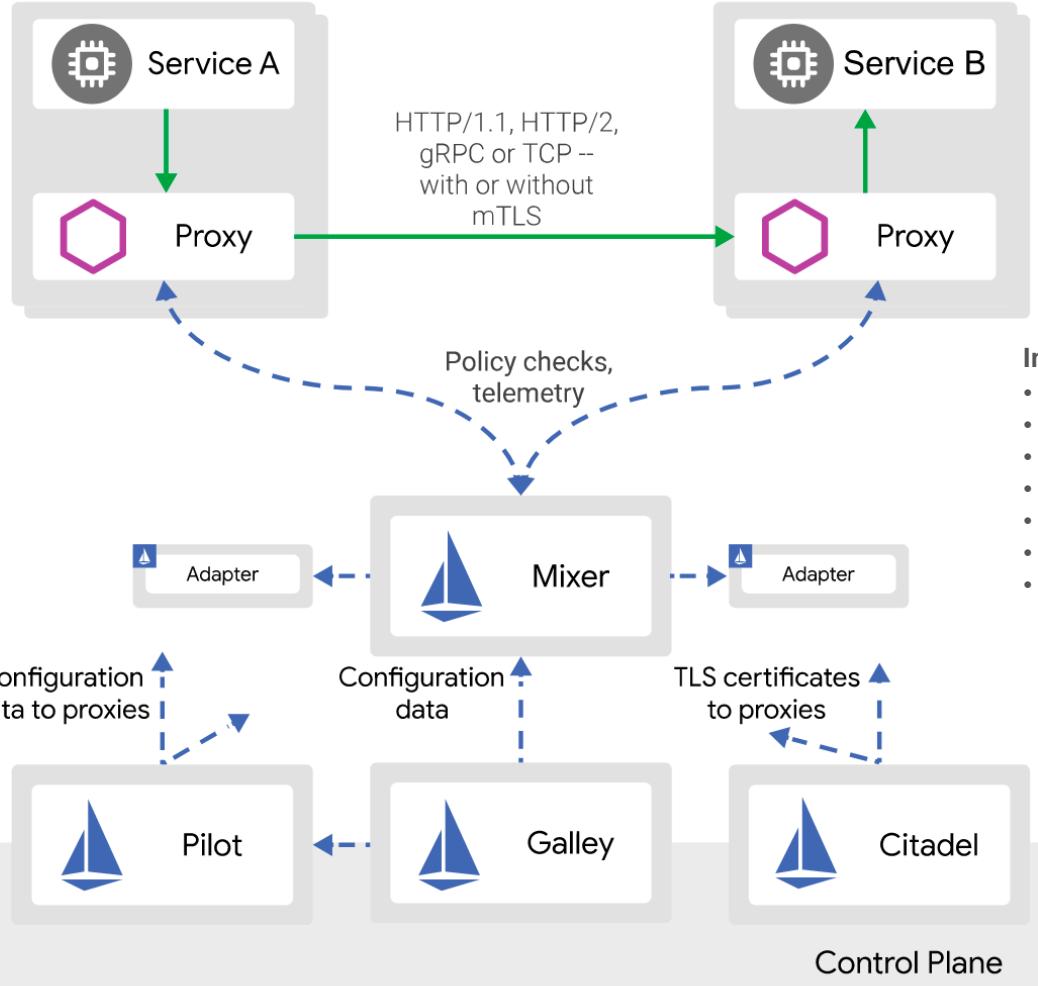


Fleet Wide
Policy
Enforcement

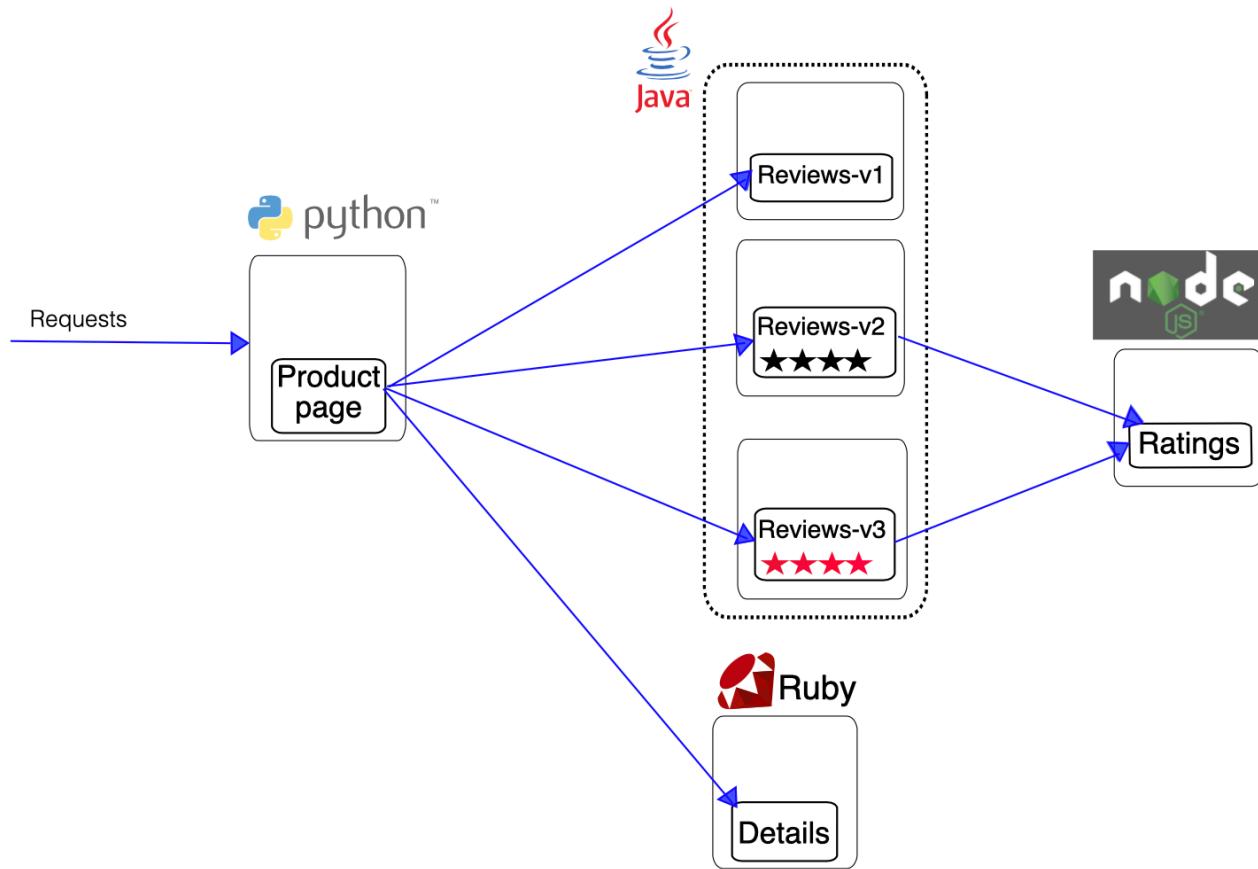


In-Depth
Telemetry and
Reporting



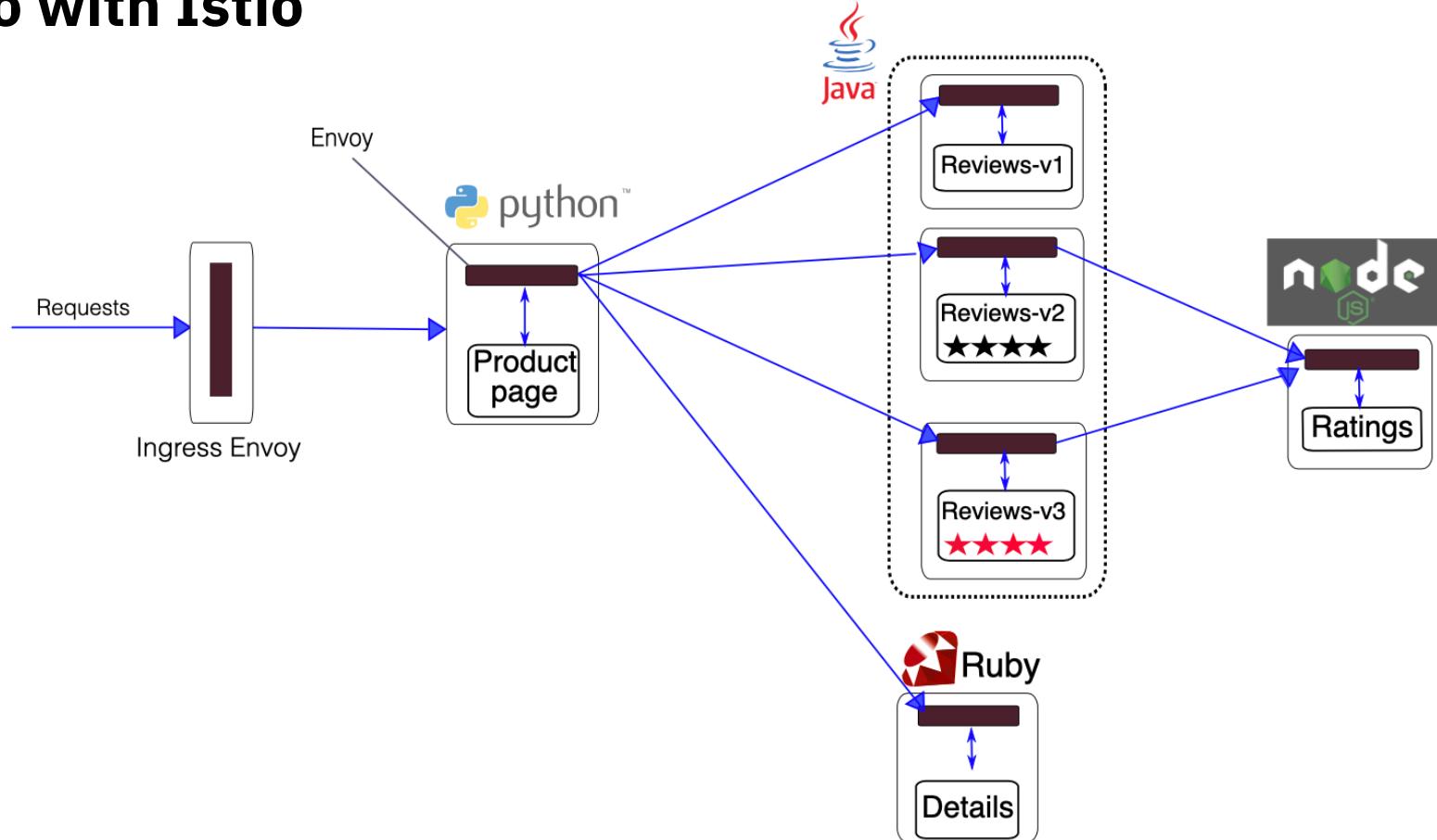


Bookinfo



Bookinfo Application without Istio

Bookinfo with Istio



Bookinfo Application

ORIGIN COMMUNITY DISTRIBUTION OF KUBERNETES



The Origin Community Distribution of Kubernetes that powers Red Hat OpenShift.

Built around a core of OCI container packaging and Kubernetes container cluster management, OKD is also augmented by application lifecycle management functionality and DevOps tooling. OKD provides a complete open source container application platform.

Standardization through Containerization

Standards are powerful forces in the software industry. They can drive technology forward by bringing together the combined efforts of multiple developers, different communities, and even competing vendors.



kubernetes
Google

Open source container orchestration and cluster management at scale.

[Learn more](#)



docker

Standardized Linux container packaging for applications and their dependencies.

[Learn more](#)



Core OS

A container-focused OS that's designed for painless management in large clusters.

[Learn more](#)



**OPERATOR
FRAMEWORK**

An open source project that provides developer and runtime Kubernetes tools, enabling you to accelerate the development of an Operator.

[Learn more](#)



cri-o

A lightweight container runtime for Kubernetes.

[Learn more](#)

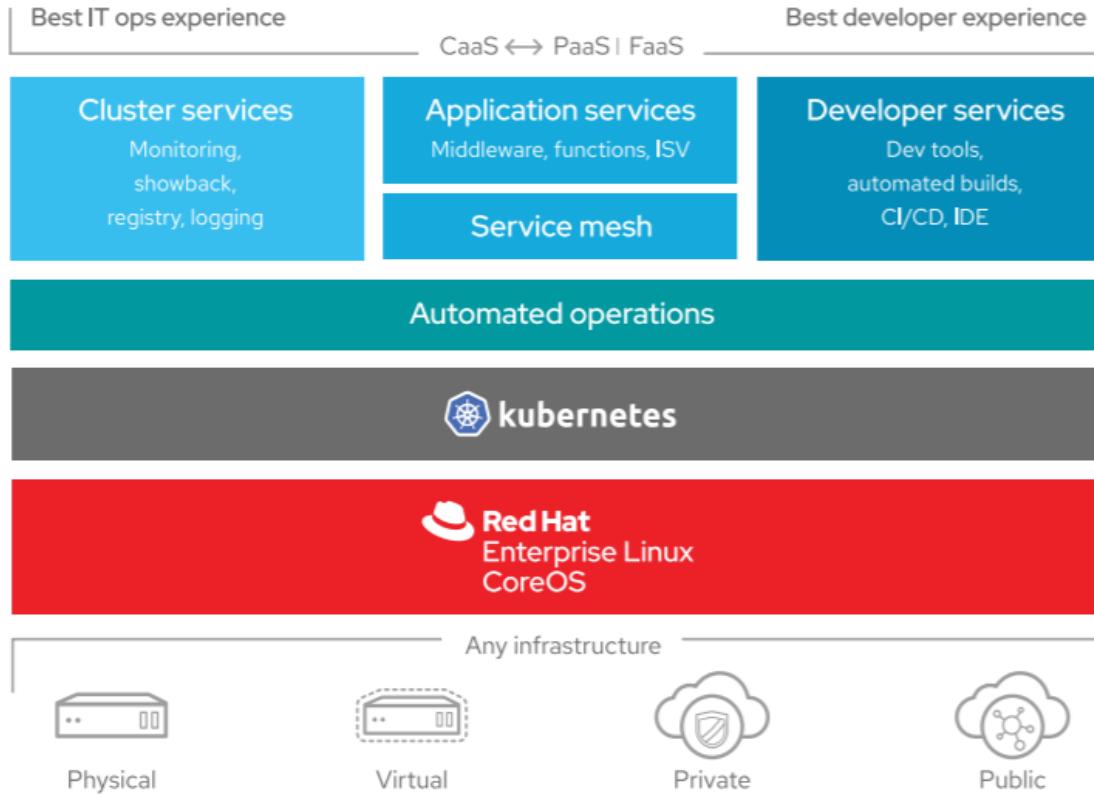


Prometheus

Prometheus is a systems and service monitoring toolkit that collects metrics from configured targets at given intervals, evaluates rule expressions, displays the results, and can trigger alerts if some condition is observed to be true.

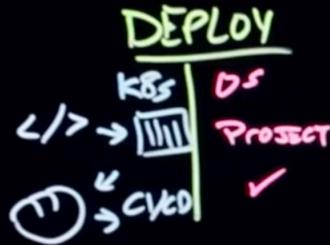
[Learn more](#)

Open Source | Secure | Fully Certified | Portable



#IBMDveloper

KUBERNETES & OPENSHIFT



LAB with Red Hat – something like this:

- <https://medium.com/@blumareks/running-a-microservice-containers-on-kubernetes-straight-from-a-github-ff73047e877b>

Thank you!

Marek Sadowski

Developer Advocate

