



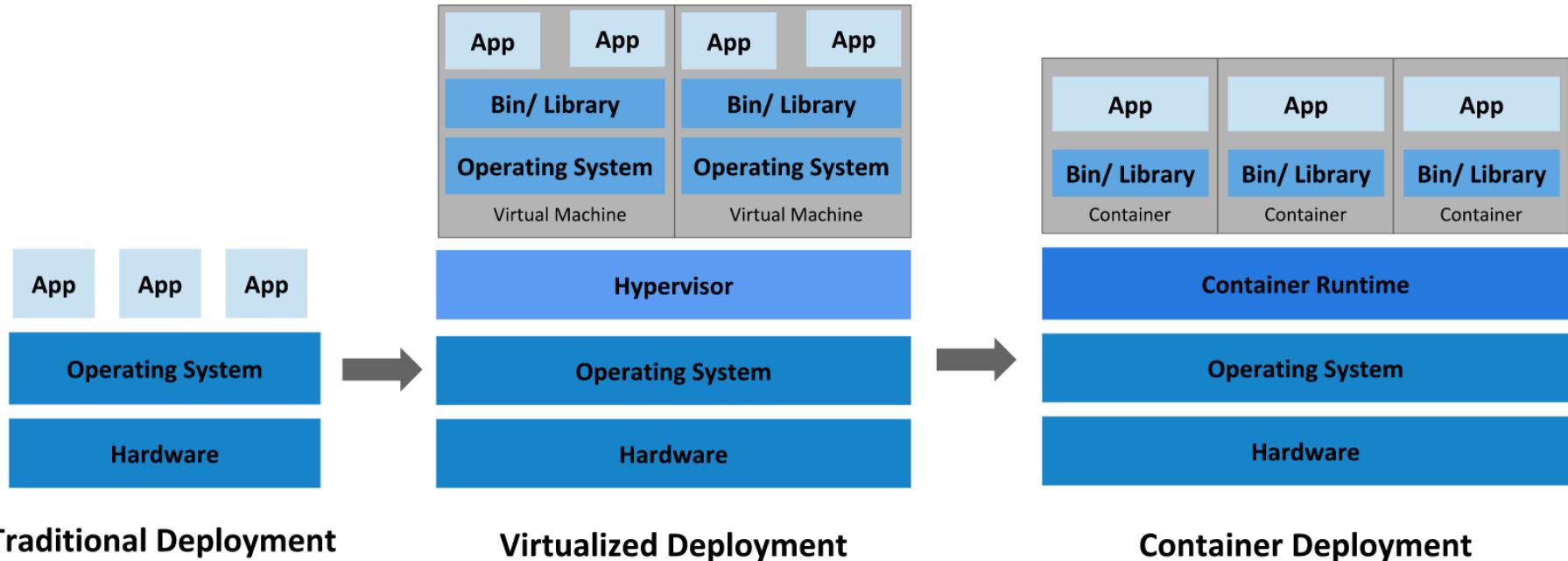
redhat.[®]



↳ Why Containers?



Why Containers?



Containers overview

- Environment isolation
- Demand growth
- New Cloud-Native Apps
- Modernize existing apps
- Dev vs Ops

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.

Containers: Dev vs Ops

Code	Logging
Libraries	Remote Access
Config	Network Config
Runtime	Monitoring
OS	



Why Containers?

- Agile
- Continuous Deployment
- Separation of Concerns
- Observability
- Consistency
- Management
- Microservices
- Resource Isolation
- Resource Utilization



Other High-Level Benefits

- Portable
- Easy to manage
- Containers provide “just enough” isolation
- Immutable



↳ Microservice Architectures



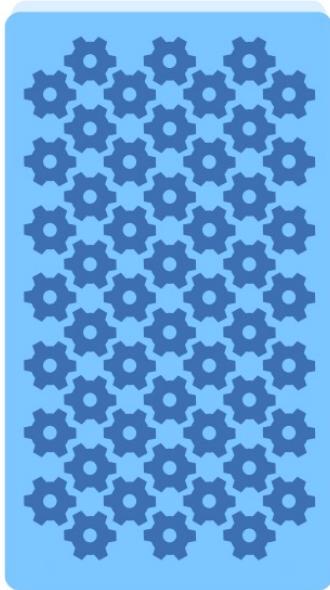
Microservices Defined

Martin Fowler: Microservices

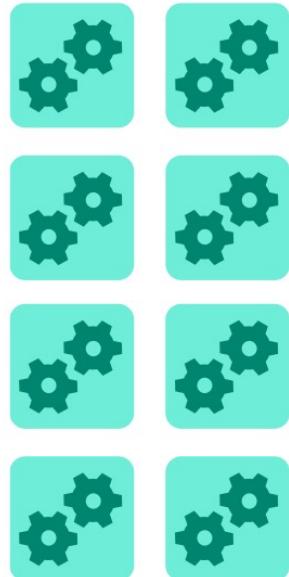
“In short, the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies. “



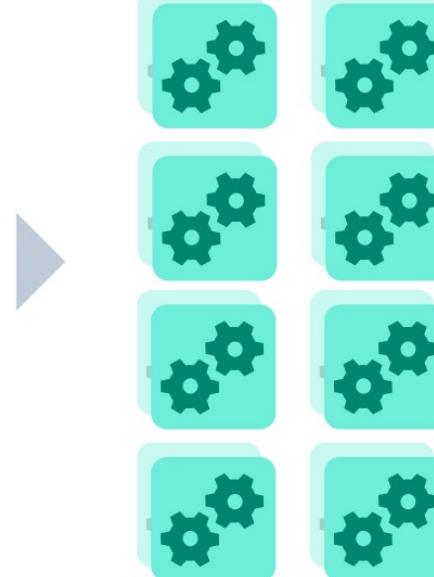
Microservices



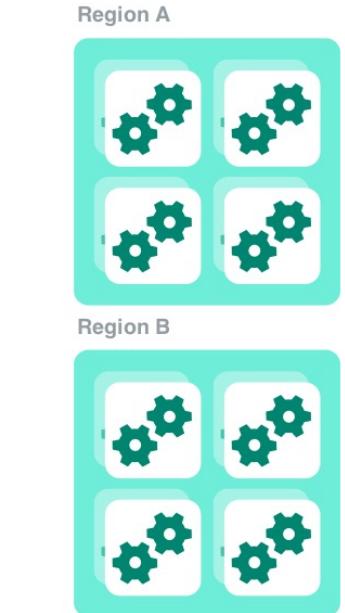
Monolithic Application



Break-down into microservices



Make each microservice HA



Protect against regional outage

↳ Orchestration

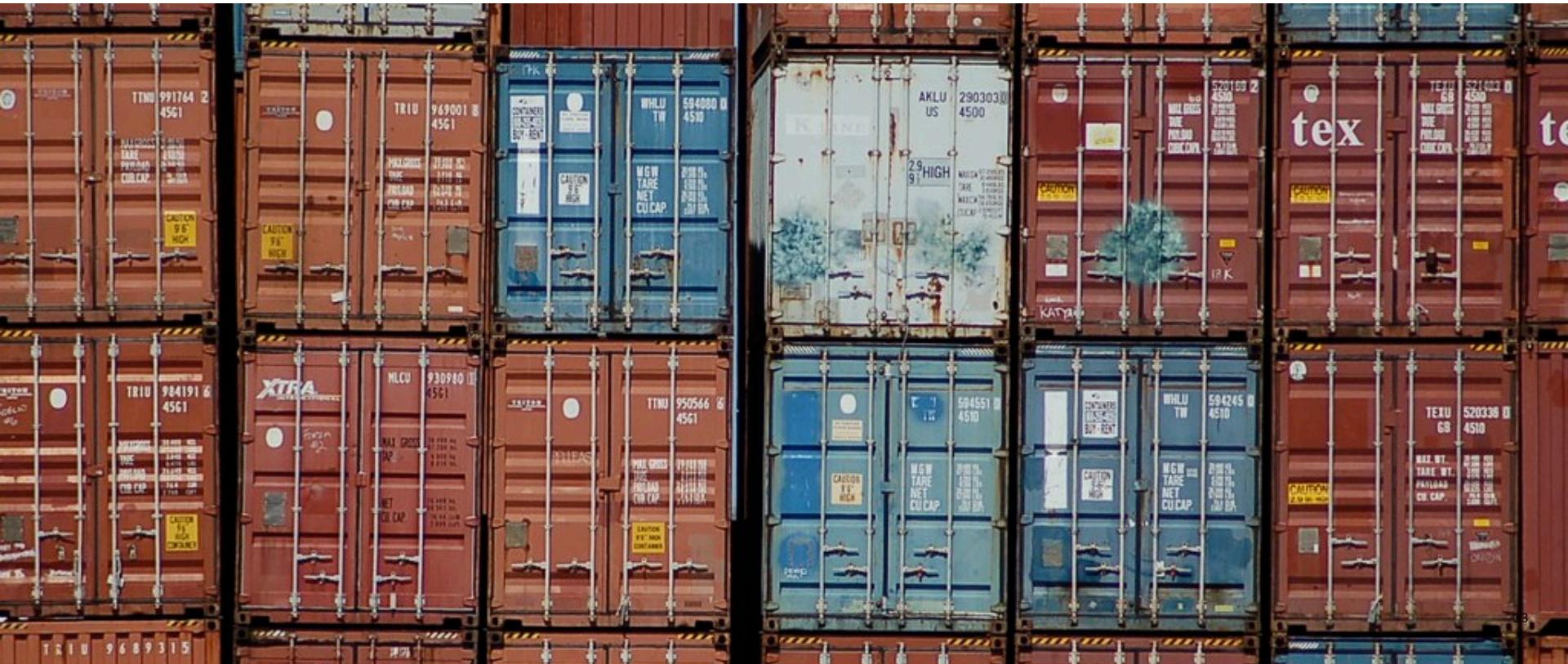


Orchestration

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



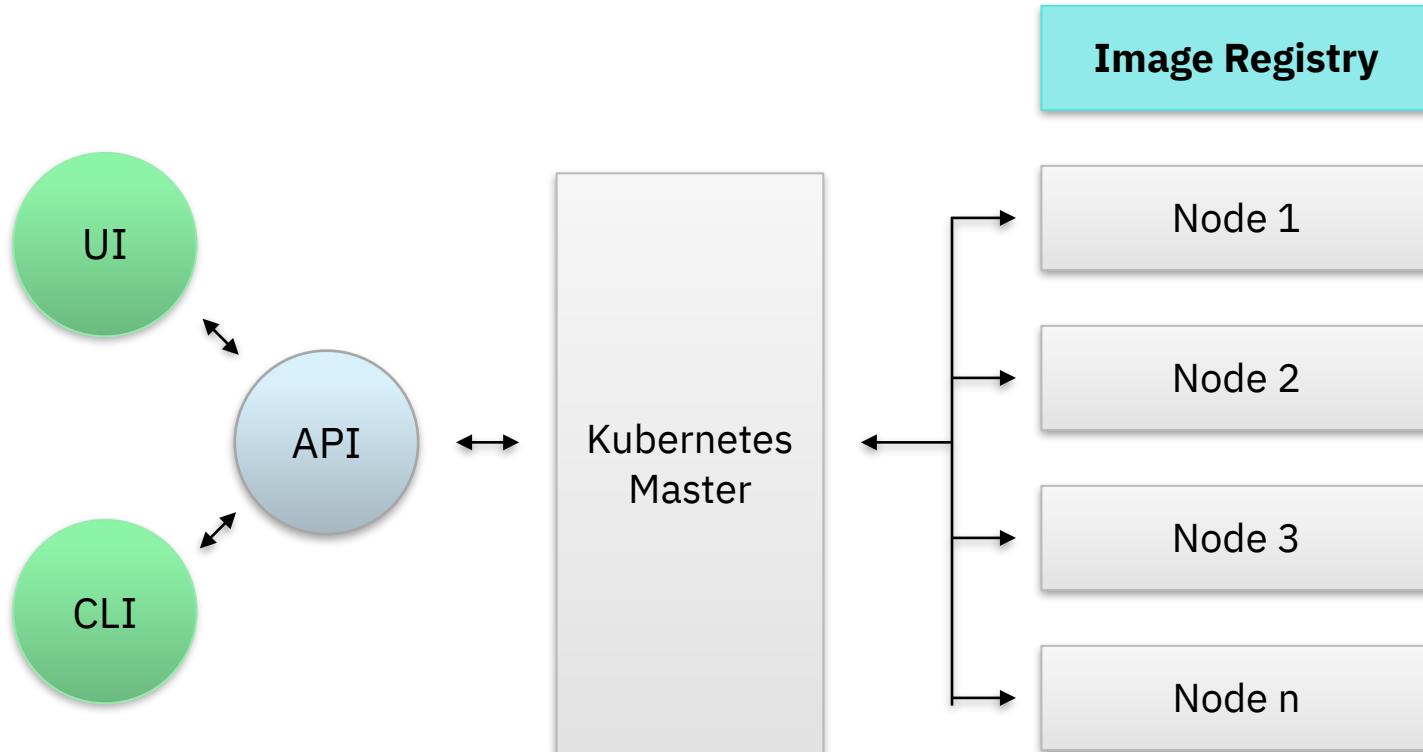
↳ Kubernetes





kubernetes

Kubernetes Architecture



Why Kubernetes?

- Service Discovery
- Storage Orchestration
- Rollouts/Rollbacks
- Automatic Bin Packing
- Self-Healing
- Secret/Config Management



What Doesn't Kubernetes Do?

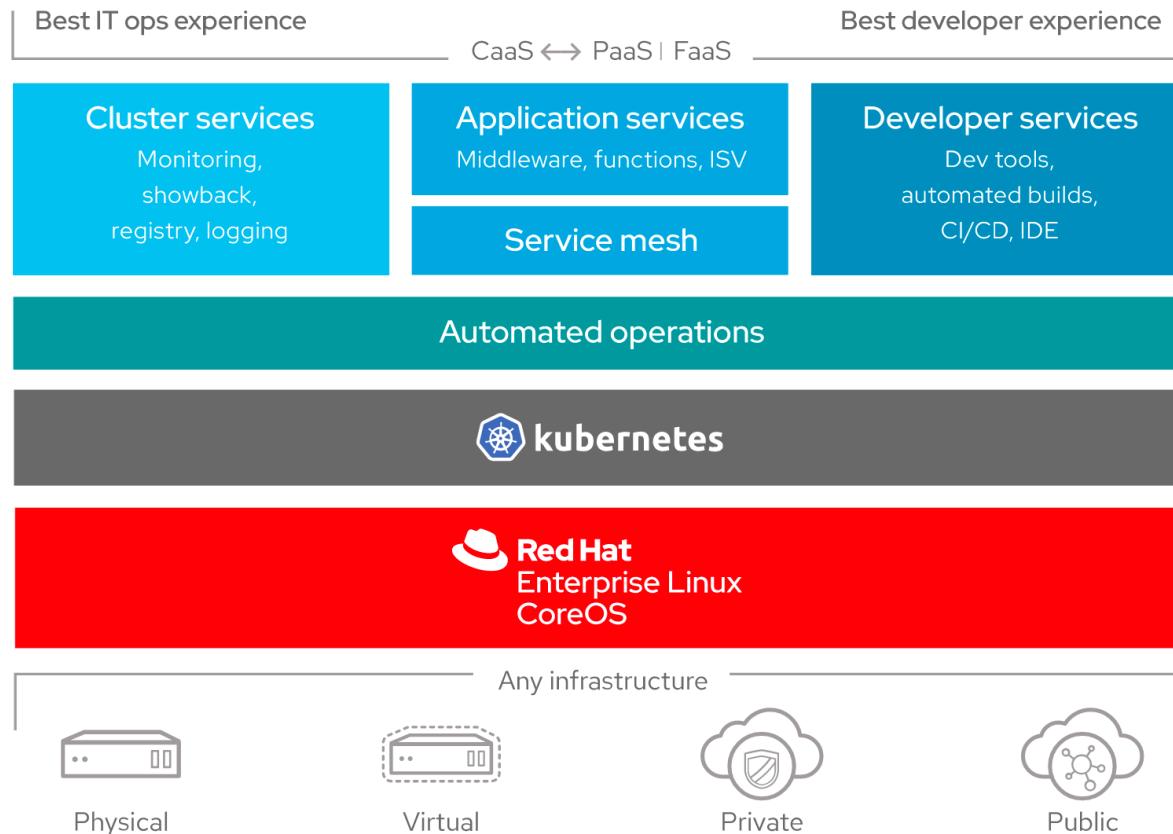
- Define Application Types
- Deploy Code
- Application-Level Services
- Logging/Monitoring/Alerting
- Config
- Machine Management



↳ OpenShift



↳ OpenShift Architectural Overview



OpenShift Overview

- Container Host & Runtime
- Enterprise Kubernetes
- Validated Integrations
- Integrated Container Registry
- Developer Workflows
- Access to Services

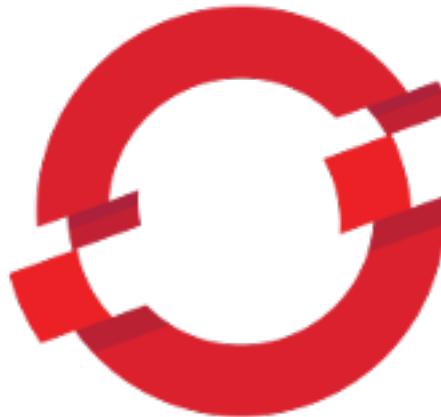


Why OpenShift 4 instead of 3?

- Not Just an Upgrade
- Immutable RHEL CoreOS
- OpenShift Services Mesh
- Operator Framework
- Knative Framework
- CodeReady Containers
- Simplified Update Process



↳ Kubernetes vs Red Hat OpenShift



kubernetes OPENSHIFT

K8s vs Red Hat OpenShift

- Product vs Project
- Security
- Management
- Integrations
- Support
- Catalog



↳ Conclusion & Lab



