

# Building OpenShift Applications with Quarkus and Red Hat Business Process Automation

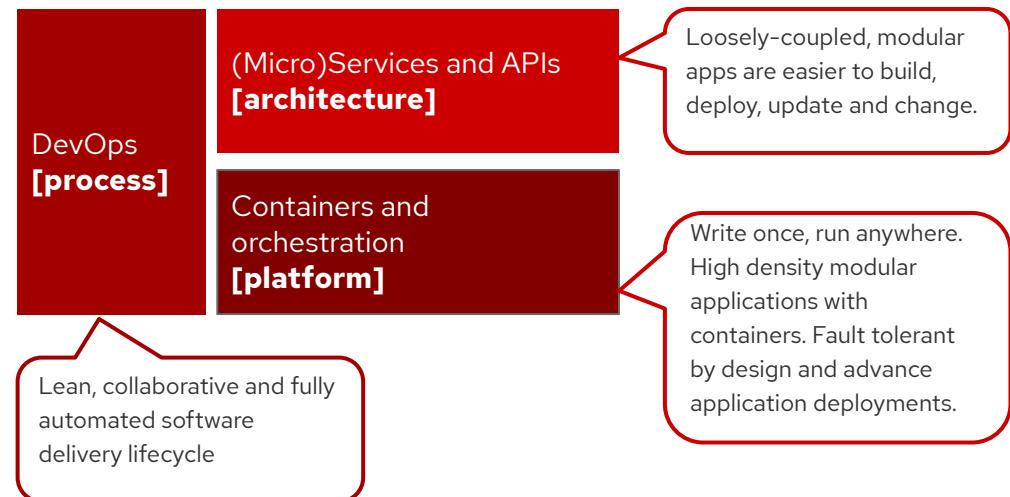
Partner Tech Days

Feb. 6 2020

# The way we innovate is rapidly changing

## Cloud Native Development

**Applications** adopting the principles of **DevSecOps CICD** practices to create **(Micro)Services** packaged as **Containers** orchestrated by **Platforms** running on top of **Cloud** infrastructure



# CONTAINER CHALLENGES

## Container security

Image scanning, patching, and compliance

## Day 2 management

Installations, upgrades, and maintenance  
Integration of existing enterprise technology

## Application delivery

Monitoring, metering, and management  
Integration of existing developer tools



## Trusted enterprise Kubernetes

Continuous security, world-class support and services, and deep expertise to confidently run any application

## A cloud-like experience, everywhere

Full-stack automated operations on a consistent foundation across on-premises or hybrid cloud infrastructure

## Empowerment for developers to innovate

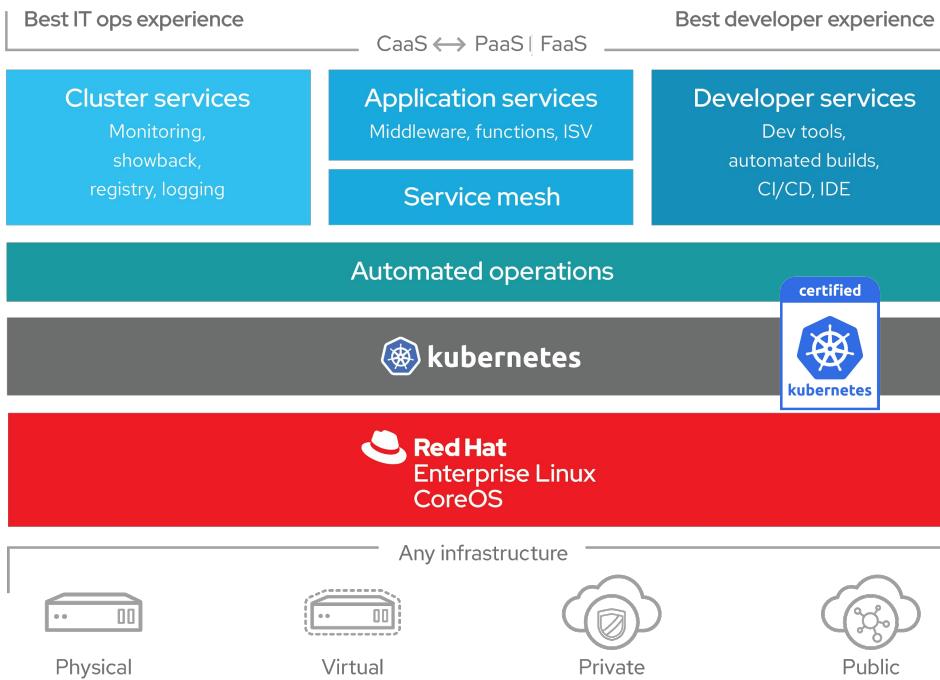
Ability to get applications to production sooner with a wide range of technologies and streamlined workflows

# Why customers choose Red Hat OpenShift

The diagram consists of four main sections, each enclosed in a dashed border:

- Trusted enterprise Kubernetes**: Contains an icon of a server with a gear inside, representing a cluster of nodes.
- Cloud-like experience everywhere**: Contains the OpenShift logo (a red circular arrow) above two clouds, one containing a network icon and the other a shield icon.
- Empowering developers to innovate**: Contains icons for Spring (green leaf), Node.js (green hexagon with 'node'), Docker (yellow blob), a smartphone displaying various app icons, Kafka (blue hexagon with 'kafka'), and Kubernetes (blue hexagon with 'K' and 'n').
- Open source innovation**: Contains icons for Helm (blue hexagon with a steering wheel), OpenShift (blue sailboat), Rook (red flame), CoreOS (red lightning bolt), Ceph (blue hexagon with a red 'd'), etcd (blue hexagon with a gear), and Kubernetes (blue hexagon with a snowflake).

# OpenShift 4 - A smarter Kubernetes platform



**Automated, full-stack installation** from the container host to application services

**Seamless Kubernetes deployment** to any cloud or on-premises environment

**Autoscaling** of cloud resources

**One-click updates** for platform, services, and applications

# Kubernetes adoption phases

## 1. Stateless apps

ReplicaSets

Deployments

## 2. Stateful apps

StatefulSets

Storage/CSI

## 3. Distributed systems

Data rebalancing

Autoscaling

Seamless upgrades

# Automated container operations

FULLY AUTOMATED DAY-1 AND DAY-2 OPERATIONS

INSTALL	DEPLOY	HARDEN	OPERATE
<b>AUTOMATED OPERATIONS</b>			
Infra provisioning	Full-stack deployment	Secure defaults	Multicloud aware
Embedded OS	On-premises and cloud	Network isolation	Monitoring and alerts
	Unified experience	Audit and logs	Full-stack patch & upgrade
		Signing and policies	Zero-downtime upgrades
			Vulnerability scanning

# Kubernetes-native day 2 management



	Flexible app architectures		No reinvention of core concepts
	Uniform deploy and debug		Truly hybrid

Operators codify operational knowledge and workflows to automate life-cycle management of containerized applications with Kubernetes

# Red Hat Certified Operators

## DEVOPS



## APM



## DATA SERVICES



## DATABASE



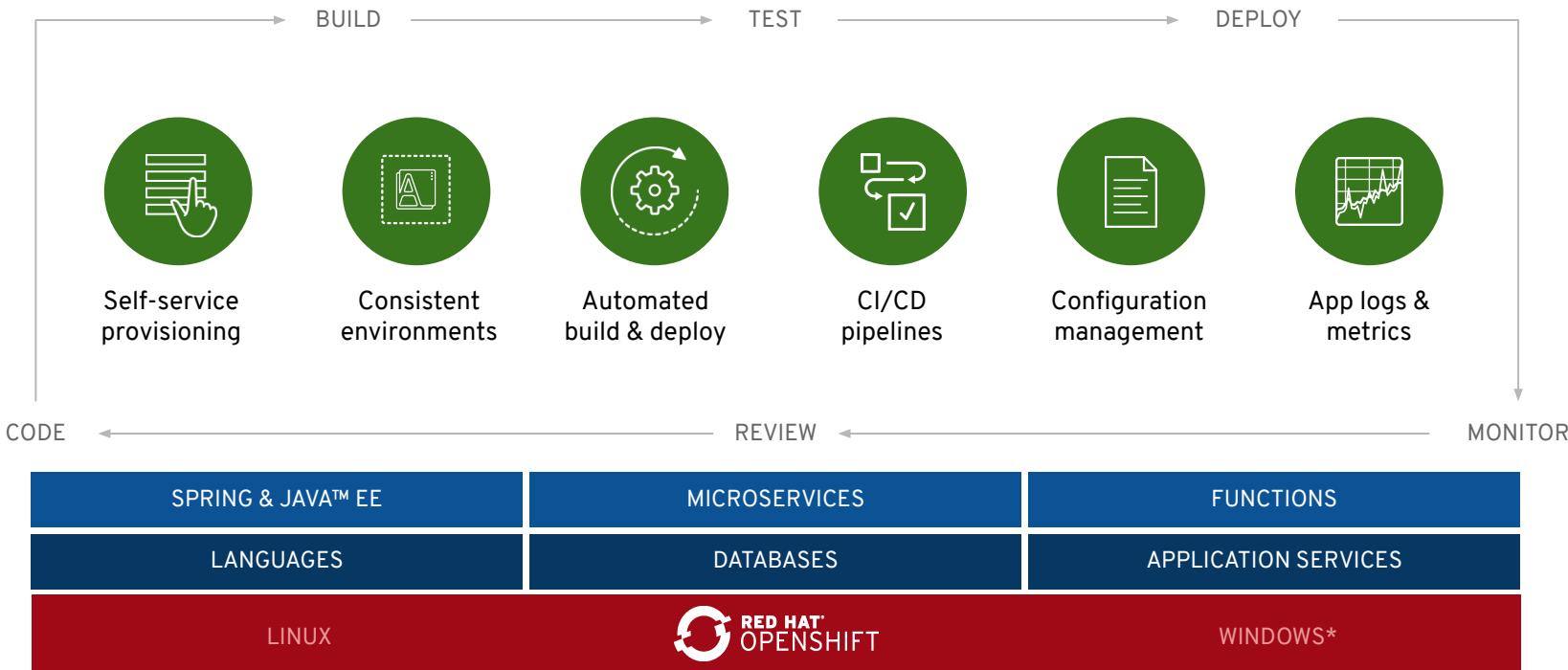
## SECURITY



## STORAGE



# OpenShift enables developer productivity



\* coming soon



# Enabling greater developer productivity

## CodeReady Workspaces

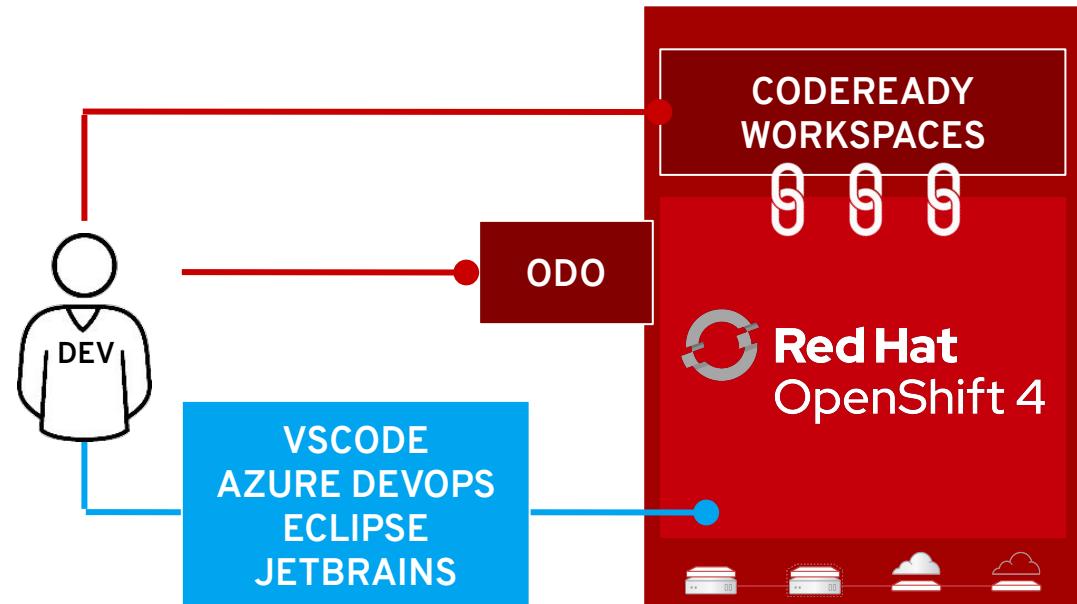
Web-Based IDE (Eclipse Che),  
Collaborative Development,  
integrated with CI/CD.

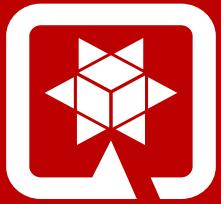
## OpenShift ODO

Advanced developer CLI

## OpenShift Plugins

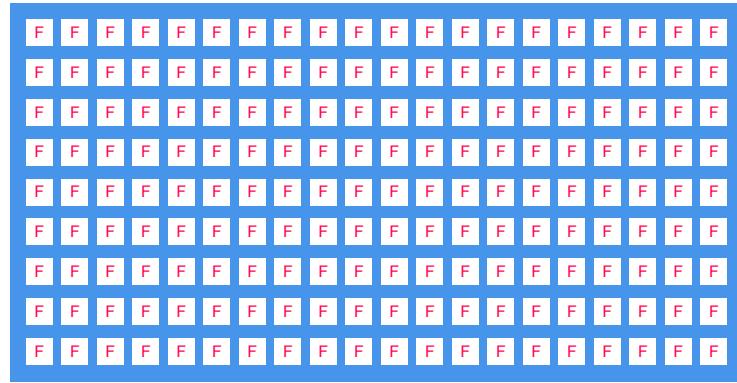
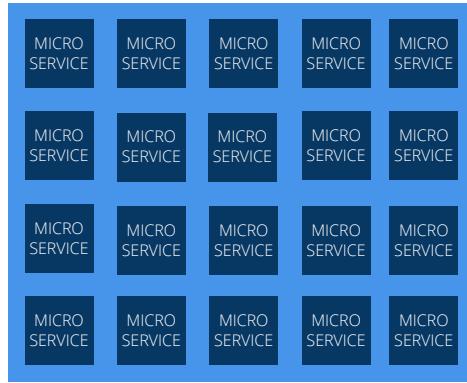
Integration plugins - VScode, Azure DevOps, Eclipse IDE, JetBrains



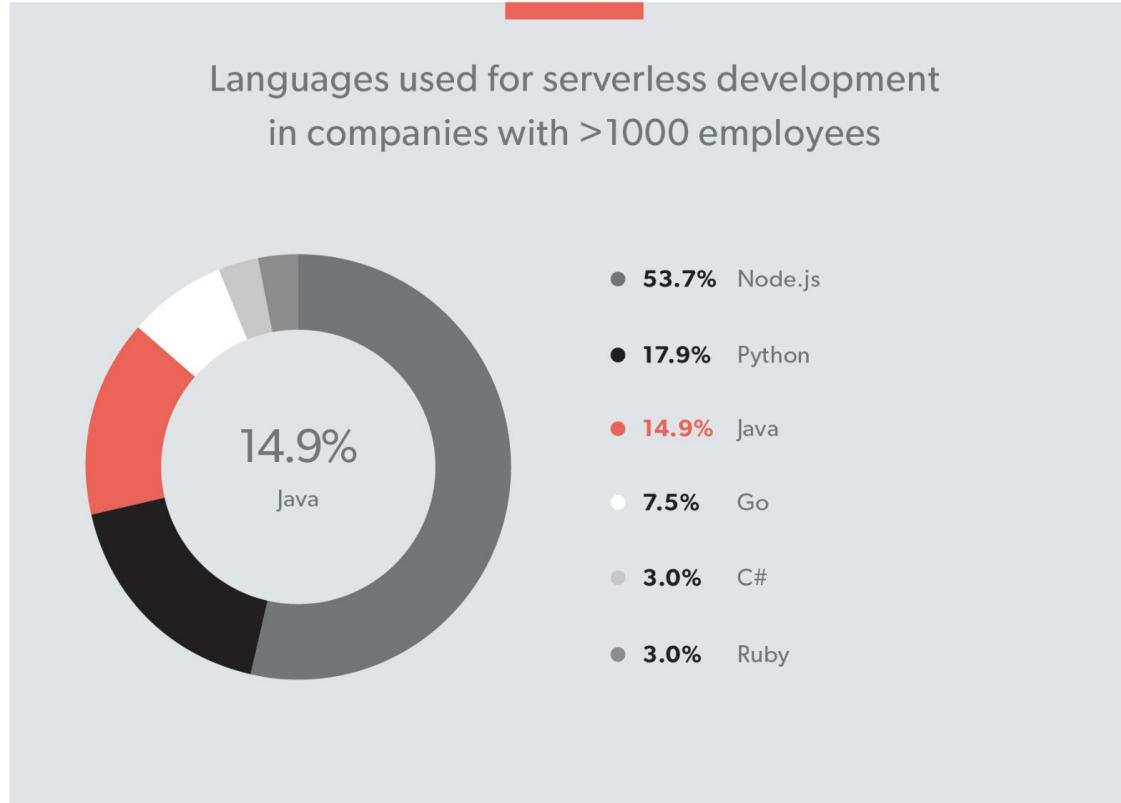


# Developing Cloud-Native Apps with Quarkus

From monolith to...



- 1 monolith  $\approx$  20 microservices  $\approx$  200 functions
- Scale to 1 vs scale to 0
- Decrease startup time

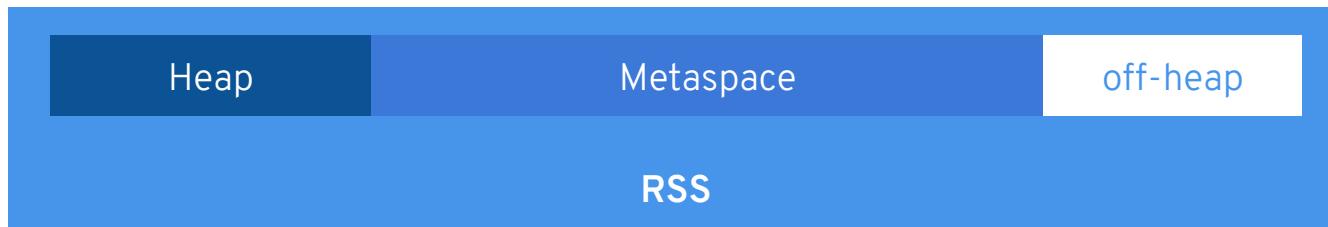


<https://serverless.com/blog/2018-serverless-community-survey-huge-growth-usage/>

# The problem with running Java in containers

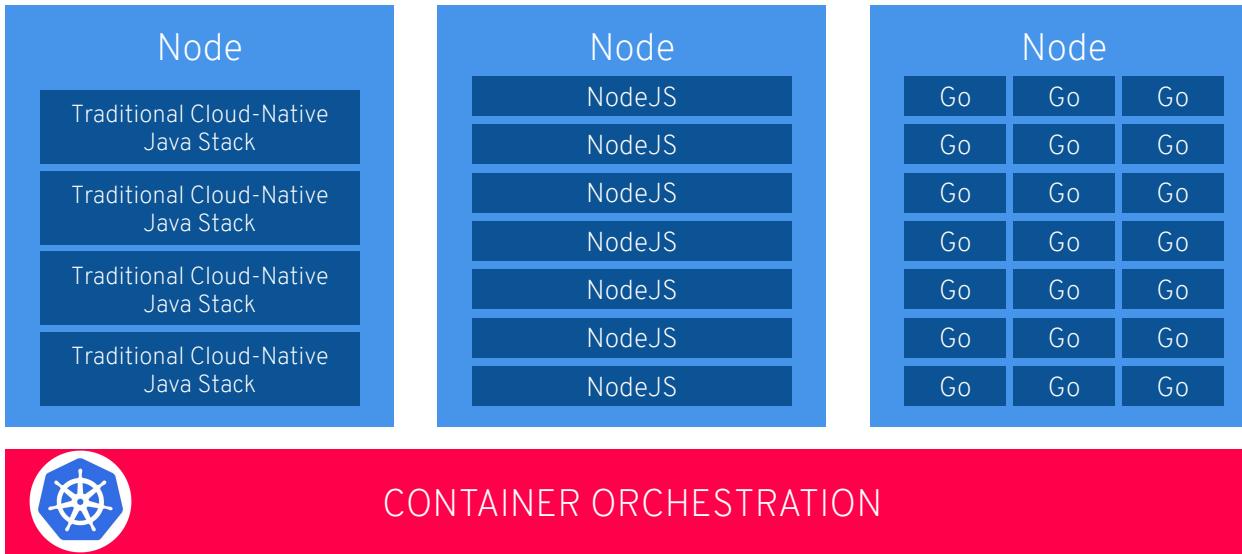
Startup Overhead: # of classes, bytecode, JIT

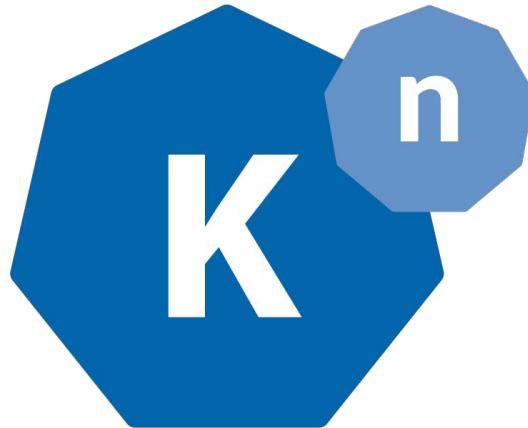
Memory Overhead: # of classes, metadata, compilation



<https://developers.redhat.com/blog/2017/03/14/java-inside-docker/>

# Low App Density with Java - Memory is a bottleneck







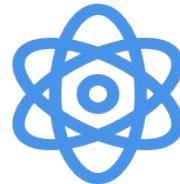
## Quarkus - Kubernetes Native Java



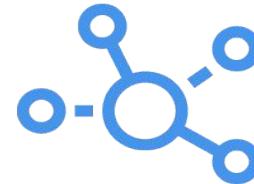
Monolith



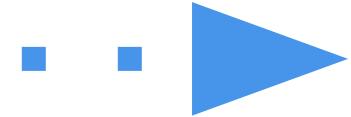
Cloud Native



Microservices



Serverless



Event-Driven  
Architecture



**kubernetes**



**Istio**



**Knative**

# Experts from cloud-native Java OS projects

**VERT.X**



Eclipse Vert.x



Hibernate



RESTEasy



Eclipse MicroProfile



WildFly

**OpenJDK™**

OpenJDK



# QUARKUS

# Time to first response

REST

Quarkus + GraalVM **0.014 Seconds**

Quarkus + OpenJDK **0.75 Seconds**

Traditional Cloud-Native Stack **4.3 Seconds**

---

REST + CRUD

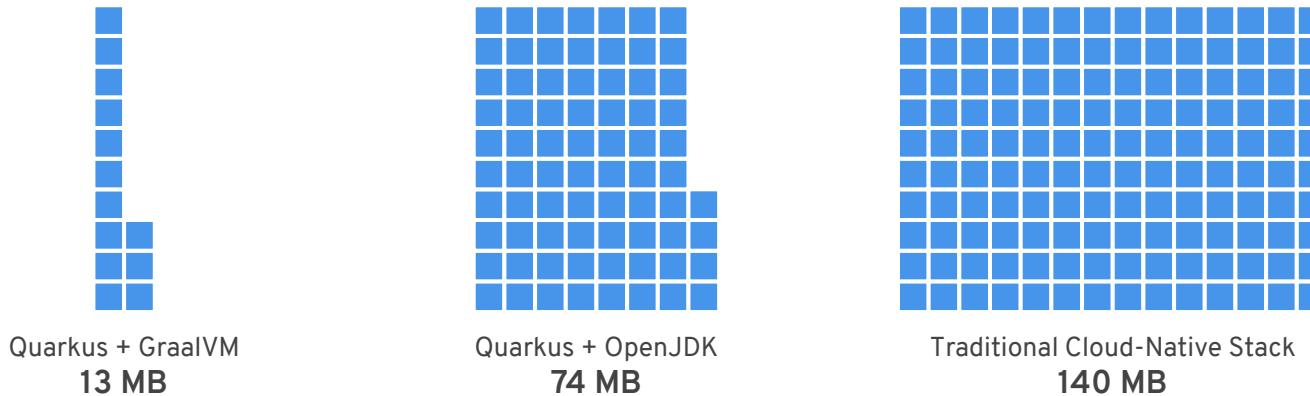
Quarkus + GraalVM **0.055 Seconds**

Quarkus + OpenJDK **2.5 Seconds**

Traditional Cloud-Native Stack **9.5 Seconds**

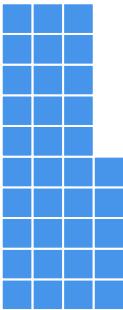
# Memory (RSS)

REST

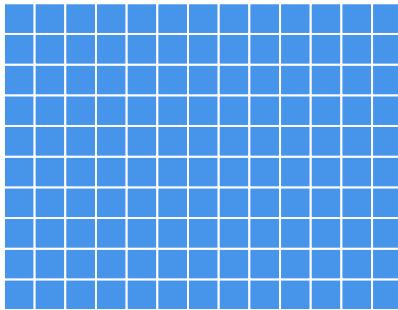


# Memory (RSS)

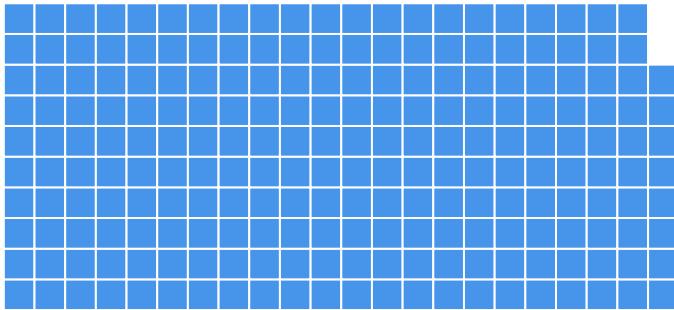
REST + CRUD



Quarkus + GraalVM  
**35 MB**



Quarkus + OpenJDK  
**130 MB**



Traditional Cloud-Native Stack  
**218 MB**

<https://quarkus.io/guides/performance-measure>

# Quarkus Business Value



## Cost Savings

Low memory, fast startup, cloud efficiency, low learning curve, serverless, high density on K8s & cloud



## Faster time to market

Through low learning curve, developer efficiency, extension ecosystem



## Remain competitive

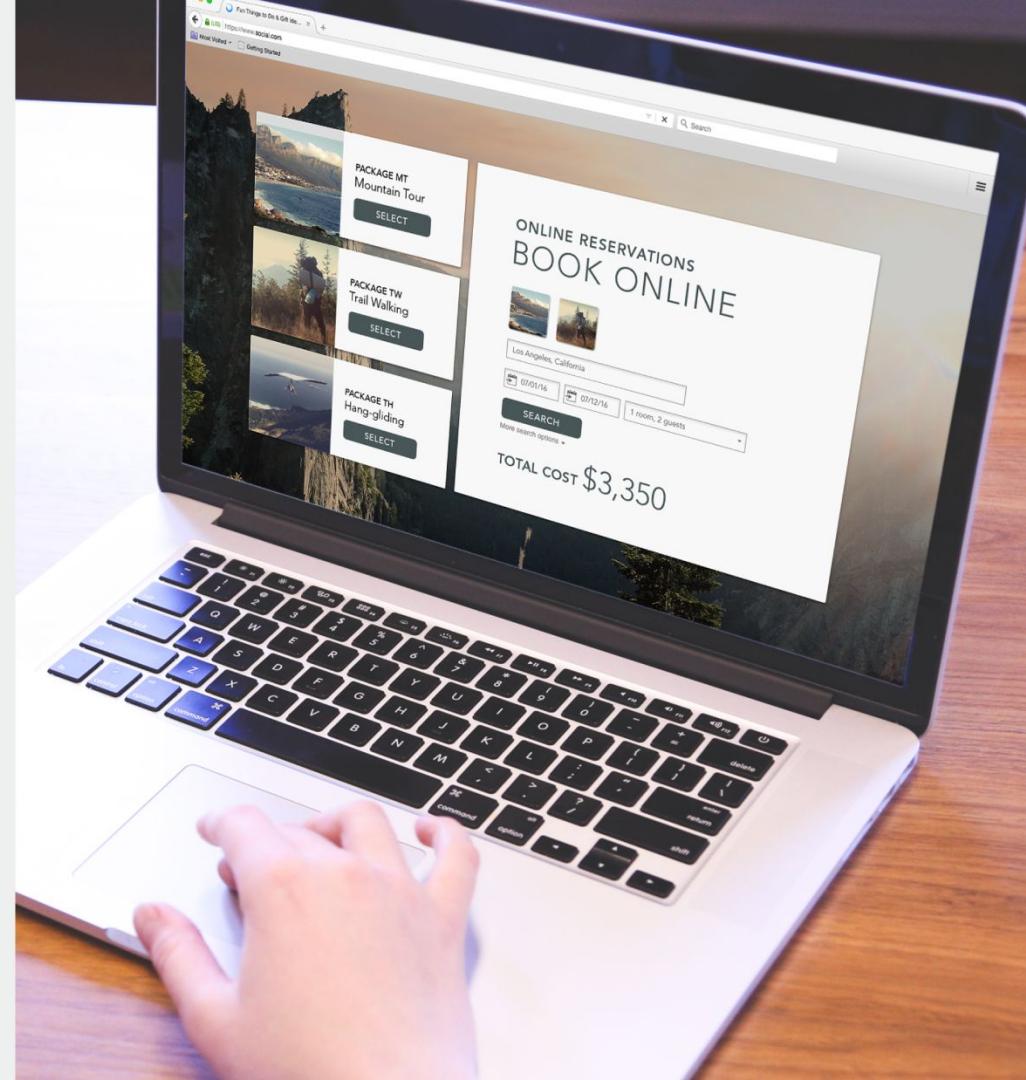
Modern software stack focused on containers and kube that enables fast experimentation and rapid delivery of changes in production



# Cloud-Native Business Process Automation

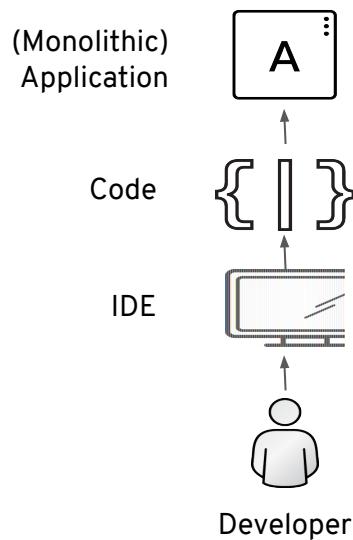
**EVERY ORGANIZATION  
MAKES OPERATIONAL  
DECISIONS**

**OPERATIONAL DECISIONS  
ARE GUIDED BY  
BUSINESS RULES**

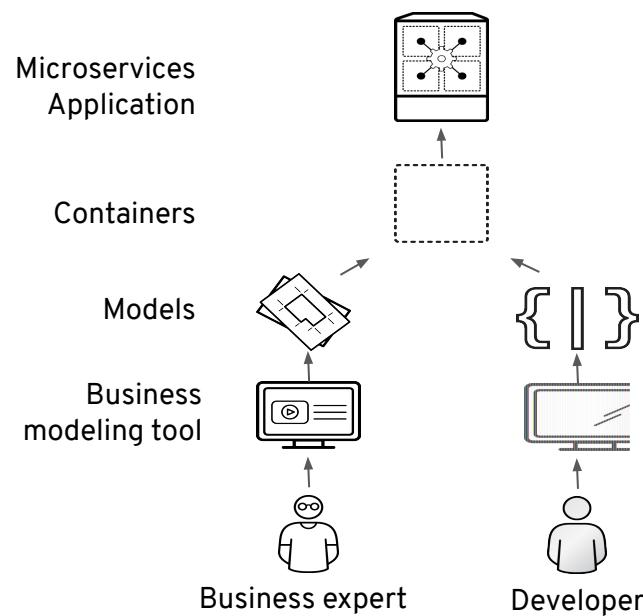


# Application Development Must Change

**THEN:**

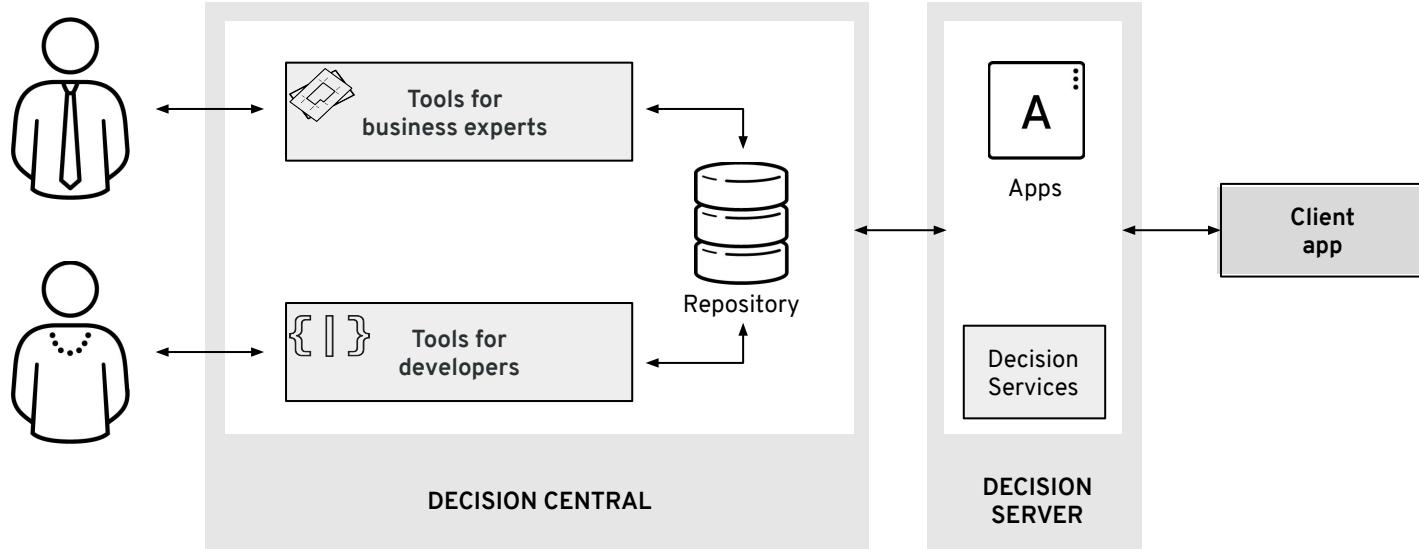


**NOW:**

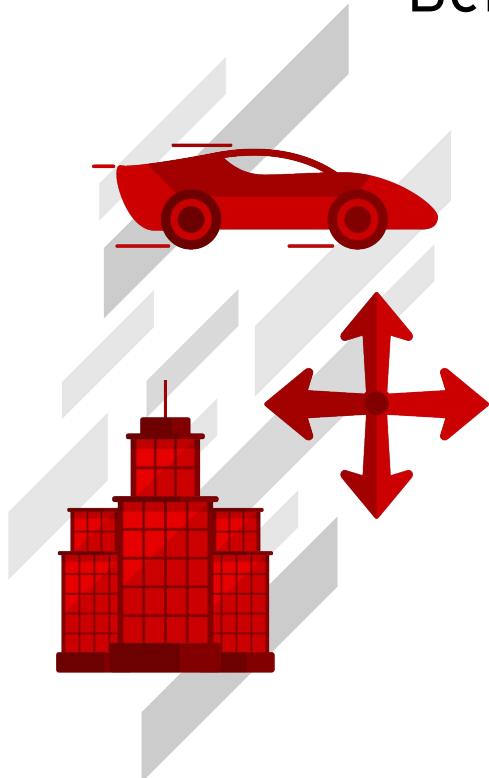


# RED HAT DECISION MANAGER

A platform for creating and running decision services



# Benefits of Red Hat's approach



## INCLUSIVE

- Build microservices directly from decision models
- Business and IT contribute as equal partners
- Business directly accountable for the business logic

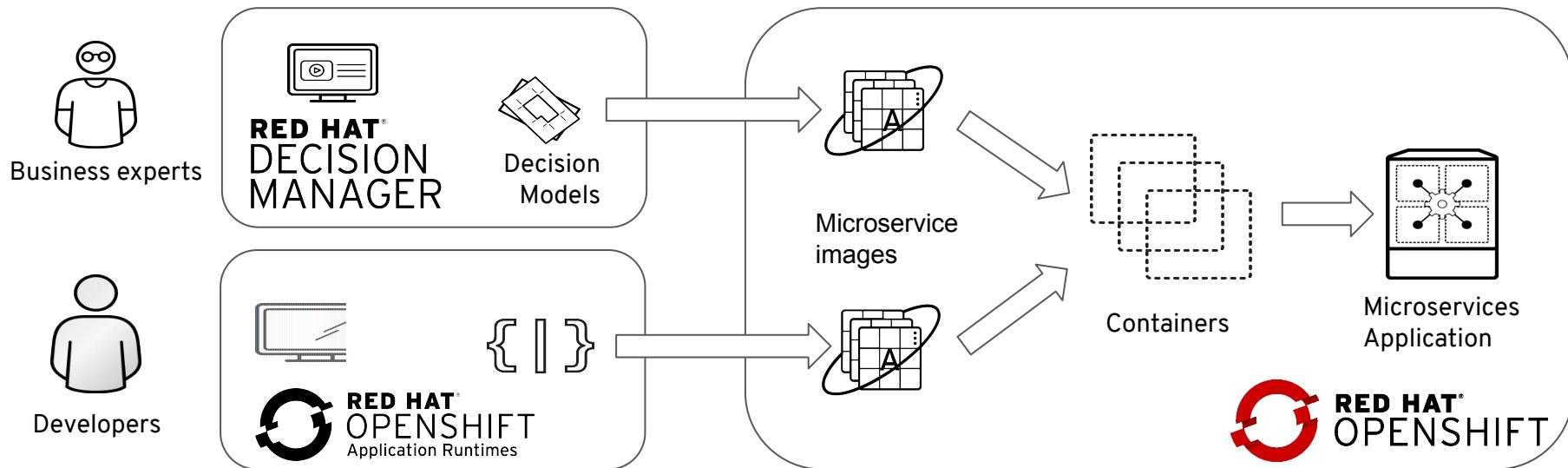
## OPEN

- Uses standard repositories and tools that integrate with existing software development environments
- Complies with standard IT development practices

## MODERN

- Supports development of containerized microservices applications
- Deploy on public, private, or hybrid clouds

# HERE'S HOW IT WORKS





# References

OpenShift:

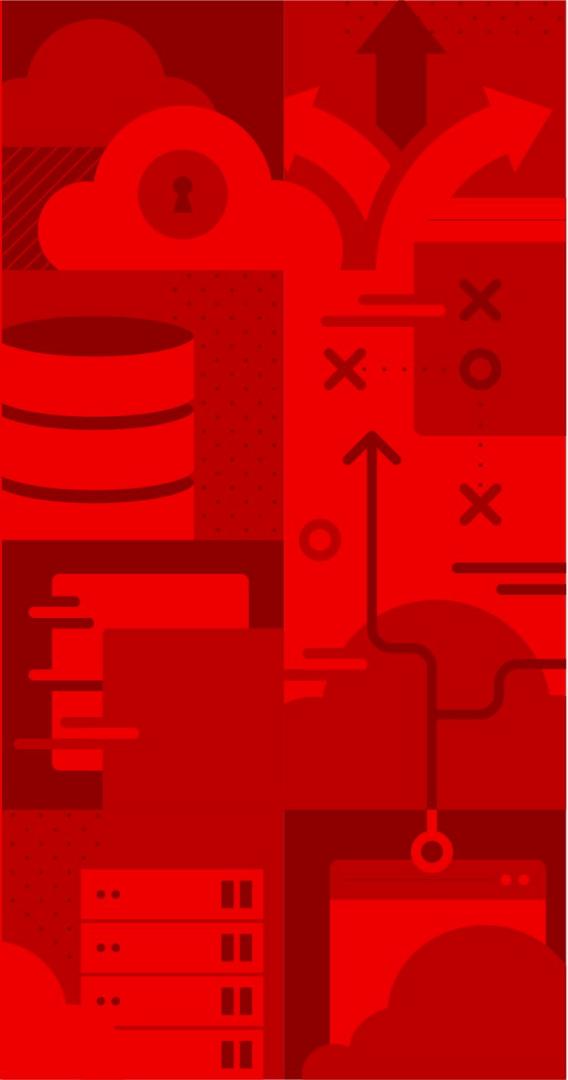
- [learn.openshift.com](https://learn.openshift.com)

Quarkus:

- [Quarkus Website](#)
- [DZone Quarkus Cheatsheet](#)

PAM / DM

- [Decision Manager Product Website](#)



# Thank you

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

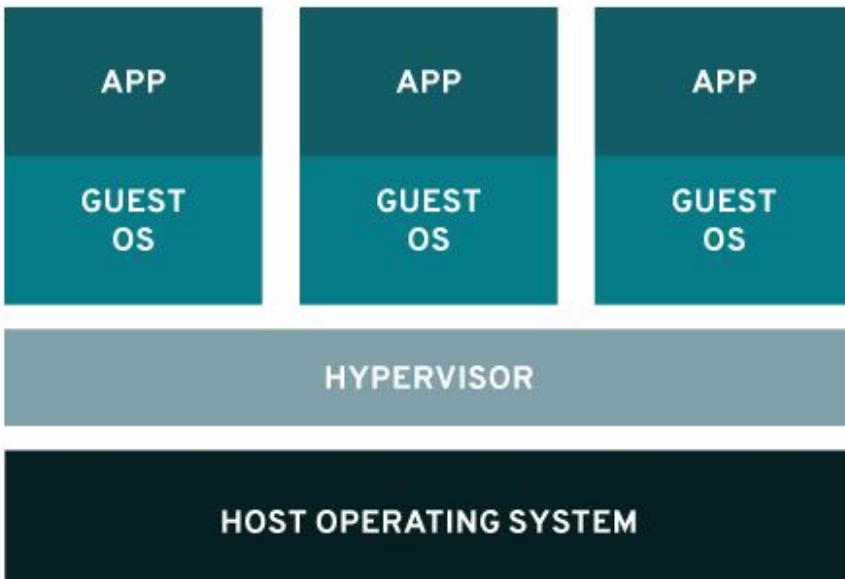
 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

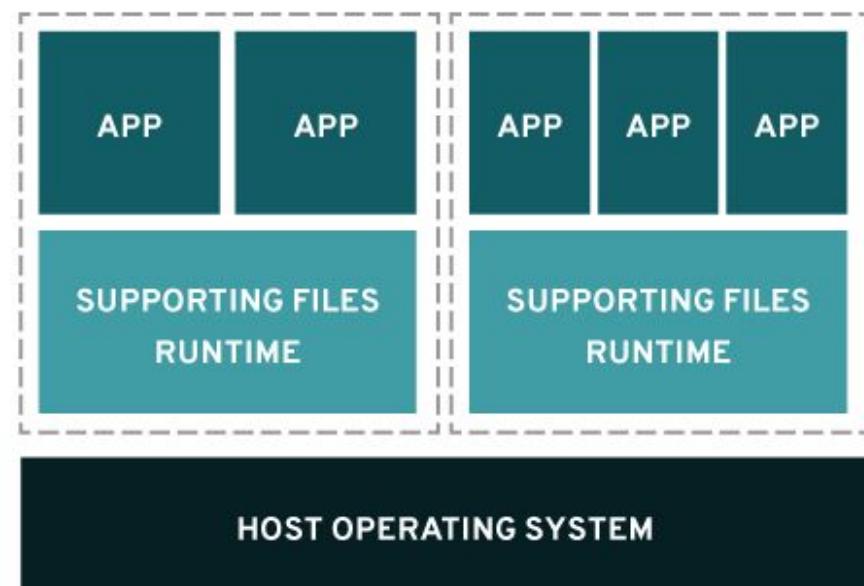
 [twitter.com/RedHat](https://twitter.com/RedHat)

# Virtualization VS Containers

## VIRTUALIZATION



## CONTAINERS



VS.