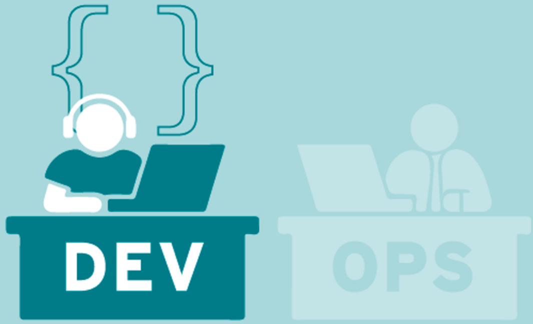


The Containers and Cloud-Native Roadshow Developer Track Lab Guide

A hands-on experience for Ops and Dev professionals

DEVELOPER TRACK MODULES

1

OPTIMIZING EXISTING APPLICATIONS

Migrate an existing monolithic Java application from a legacy platform to Red Hat.

Modernize by incrementally refactoring to microservices architecture and modern Java platform

2

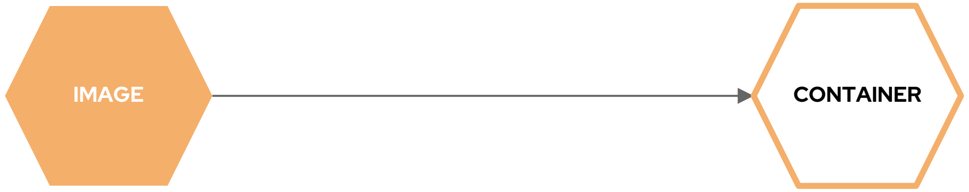
ADVANCED CLOUD NATIVE WITH EVENT-DRIVEN SERVERLESS

Dynamically respond to events and scale applications using powerful Kubernetes constructs

OpenShift Concepts

a container is the smallest compute unit

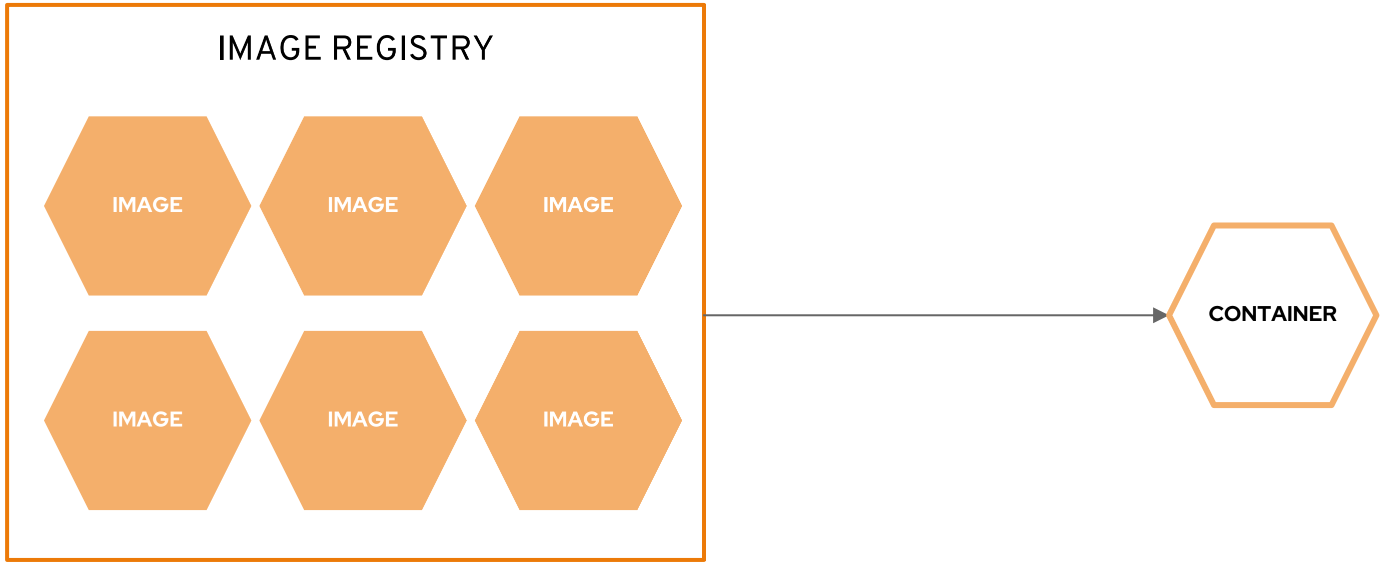
containers are created from container images



BINARY

RUNTIME

container images are stored in an image registry



an image repository contains all versions of an image in the image registry

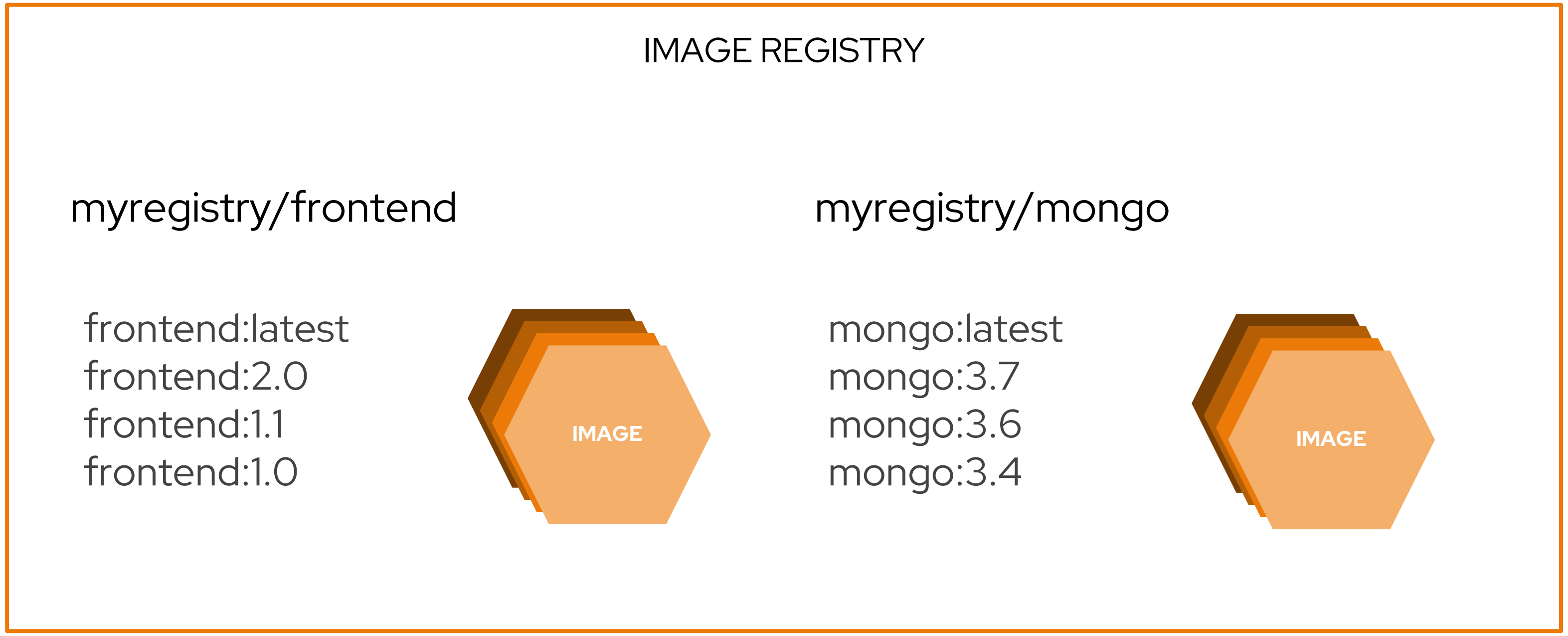


IMAGE REGISTRY

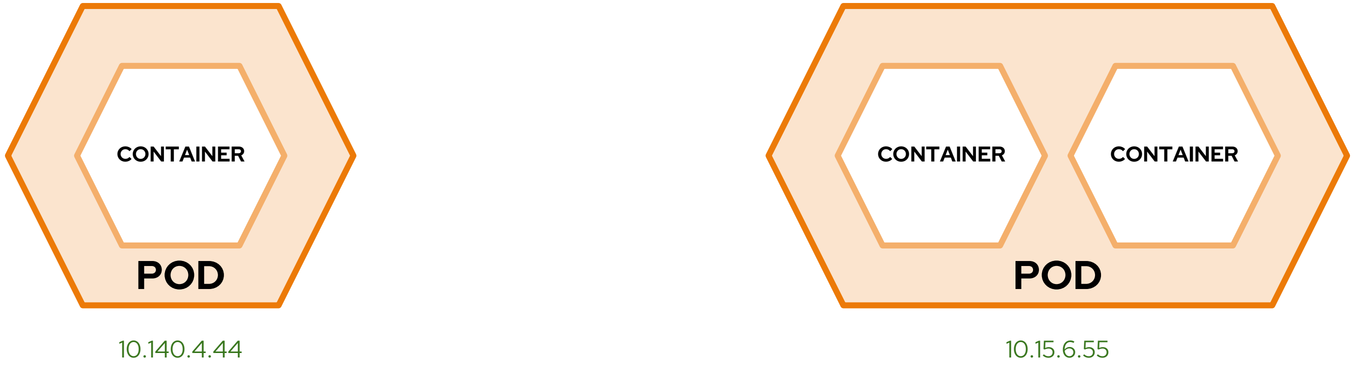
myregistry/frontend

frontend:latest
frontend:2.0
frontend:1.1
frontend:1.0

myregistry/mongo

mongo:latest
mongo:3.7
mongo:3.6
mongo:3.4

containers are wrapped in pods which are units of deployment and management



ReplicationControllers & ReplicaSets ensure a specified number of pods are running at any given time

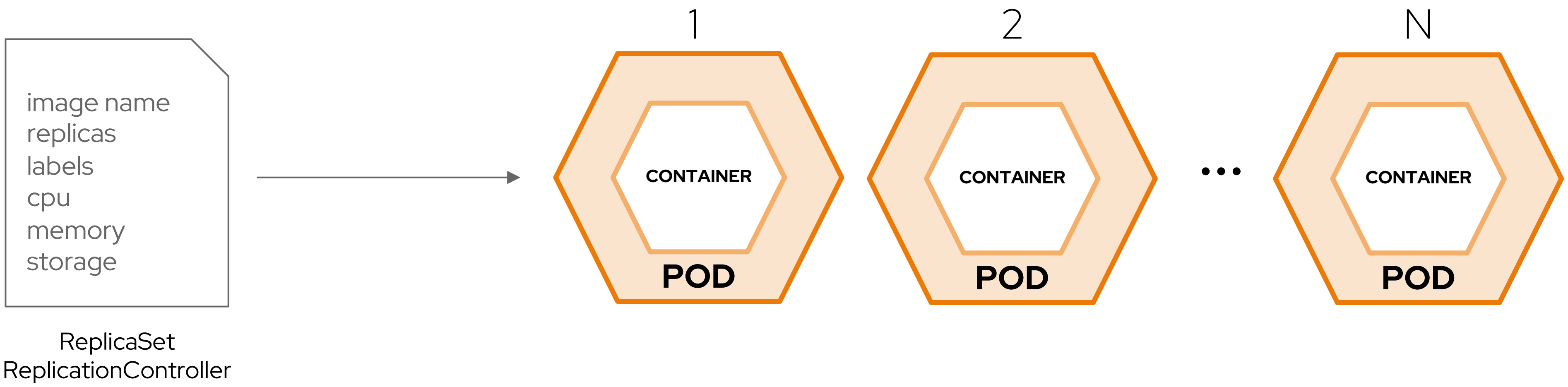
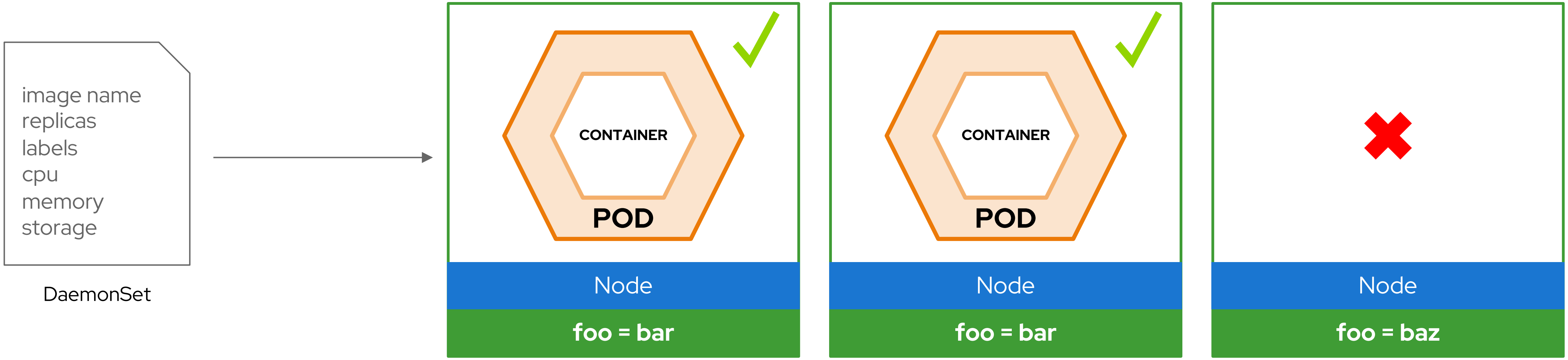


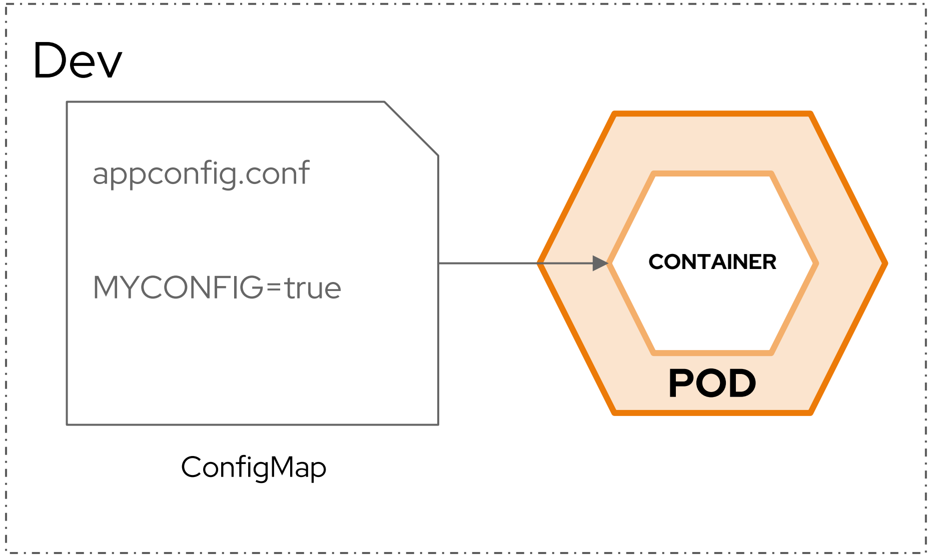
image name
replicas
labels
cpu
memory
storage

ReplicaSet
ReplicationController

a daemonset ensures that all (or some) nodes run a copy of a pod



configmaps **allow you to decouple configuration artifacts from image content**



Dev

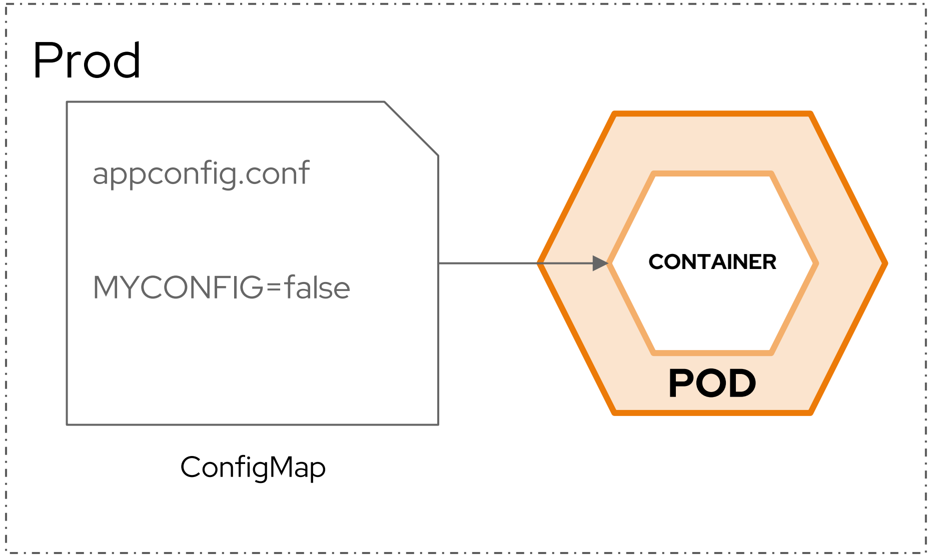
appconfig.conf

MYCONFIG=true

ConfigMap

CONTAINER

POD



Prod

appconfig.conf

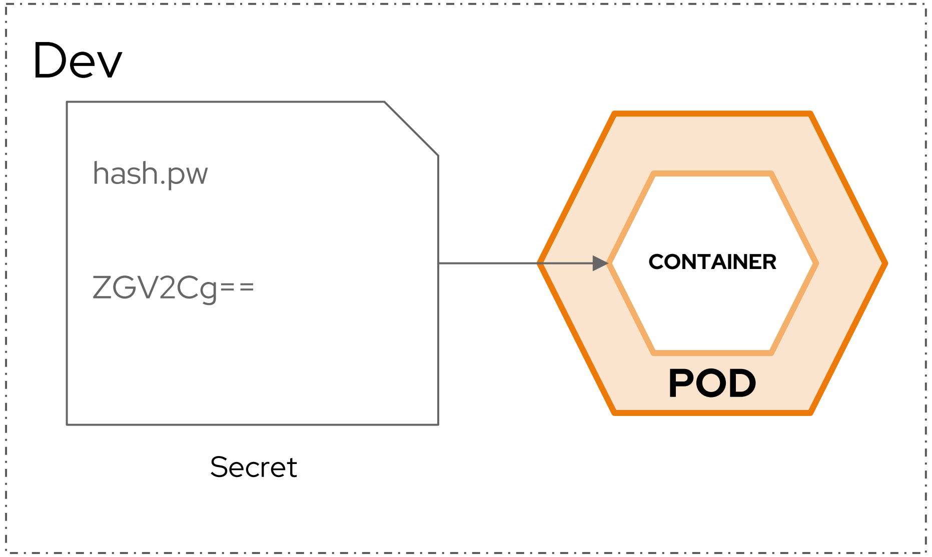
MYCONFIG=false

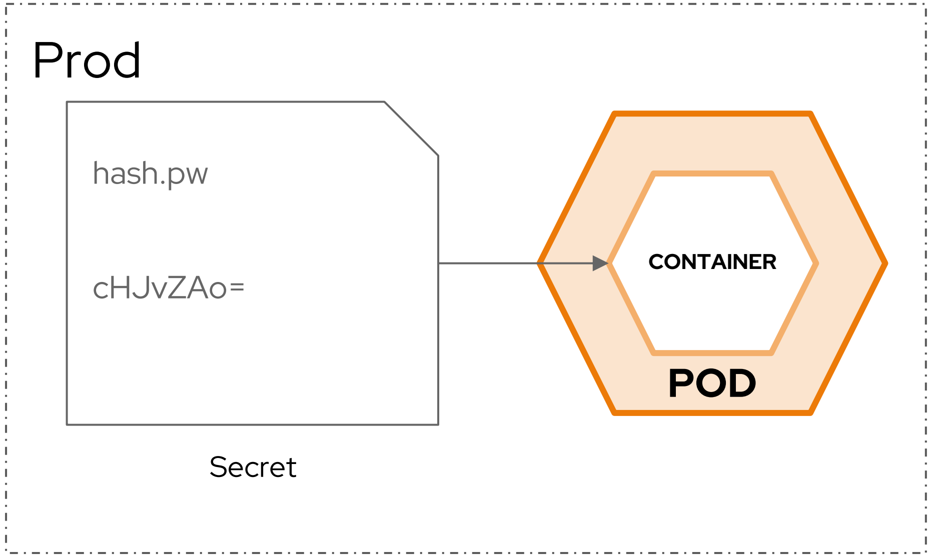
ConfigMap

CONTAINER

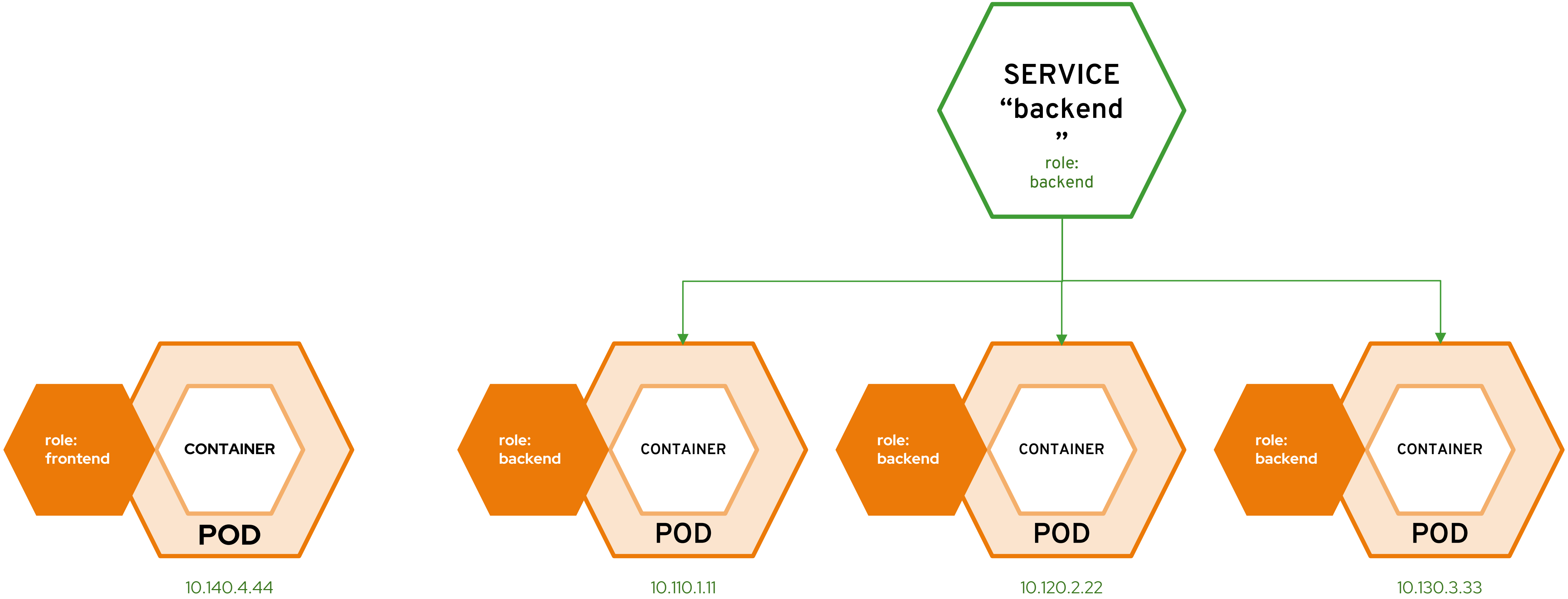
POD

secrets **provide a mechanism to hold sensitive information such as passwords**

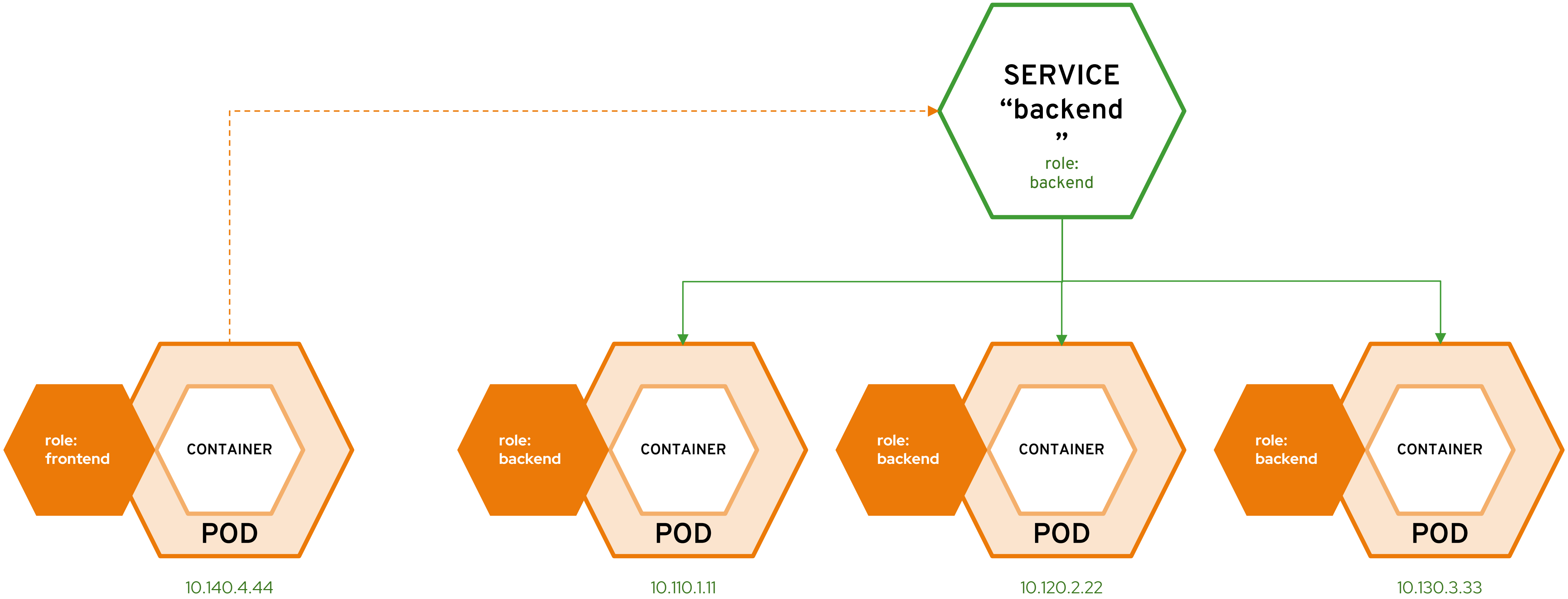




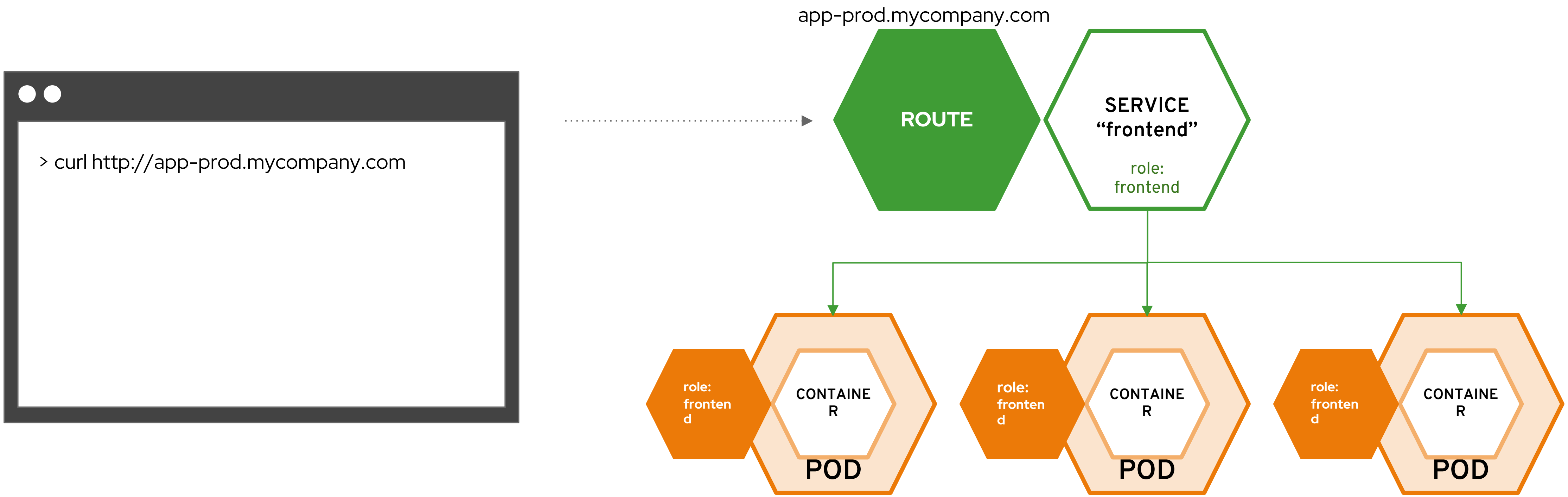
services provide internal load-balancing and service discovery across pods



apps can talk to each other via services

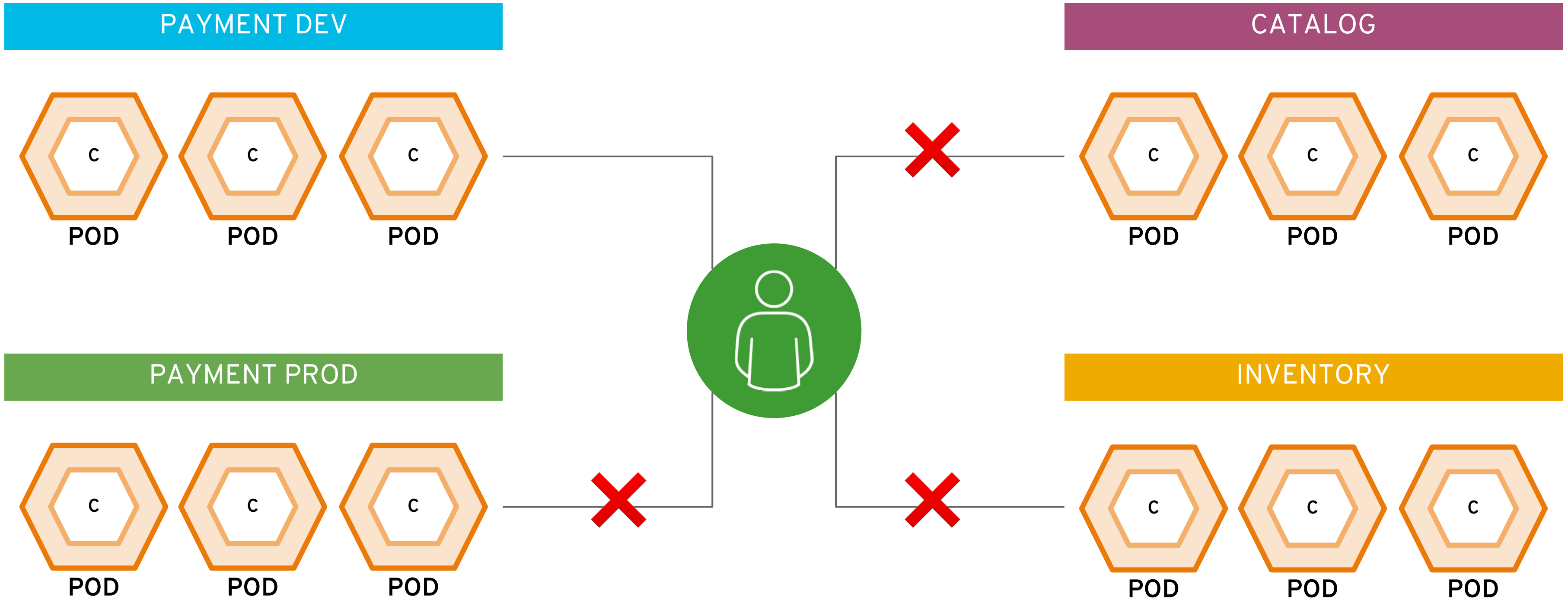


routes **make services accessible to clients outside the environment via real-world urls**



```
> curl http://app-prod.mycompany.com
```

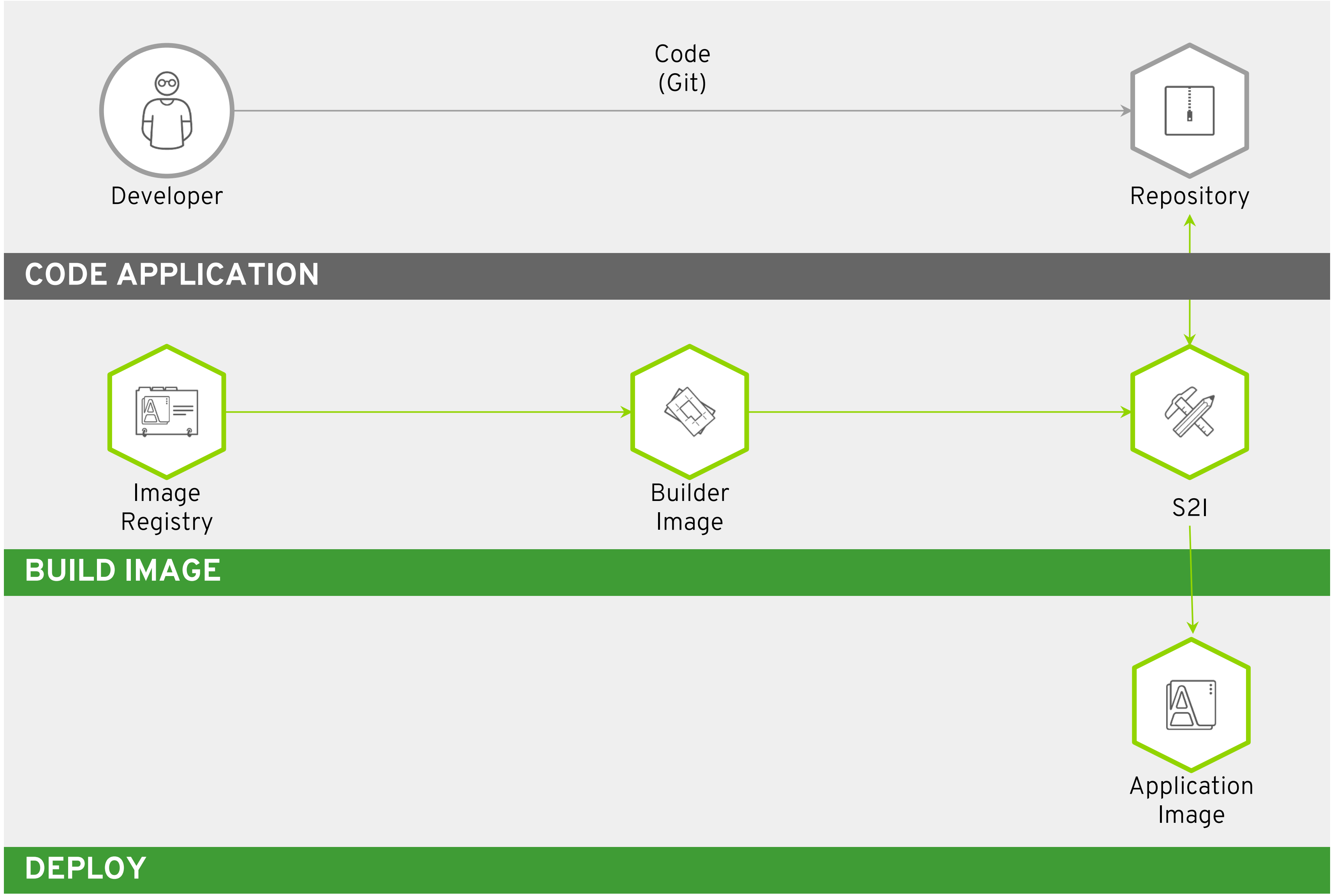
projects isolate apps across environments, teams, groups and departments



DEPLOY YOUR SOURCE CODE

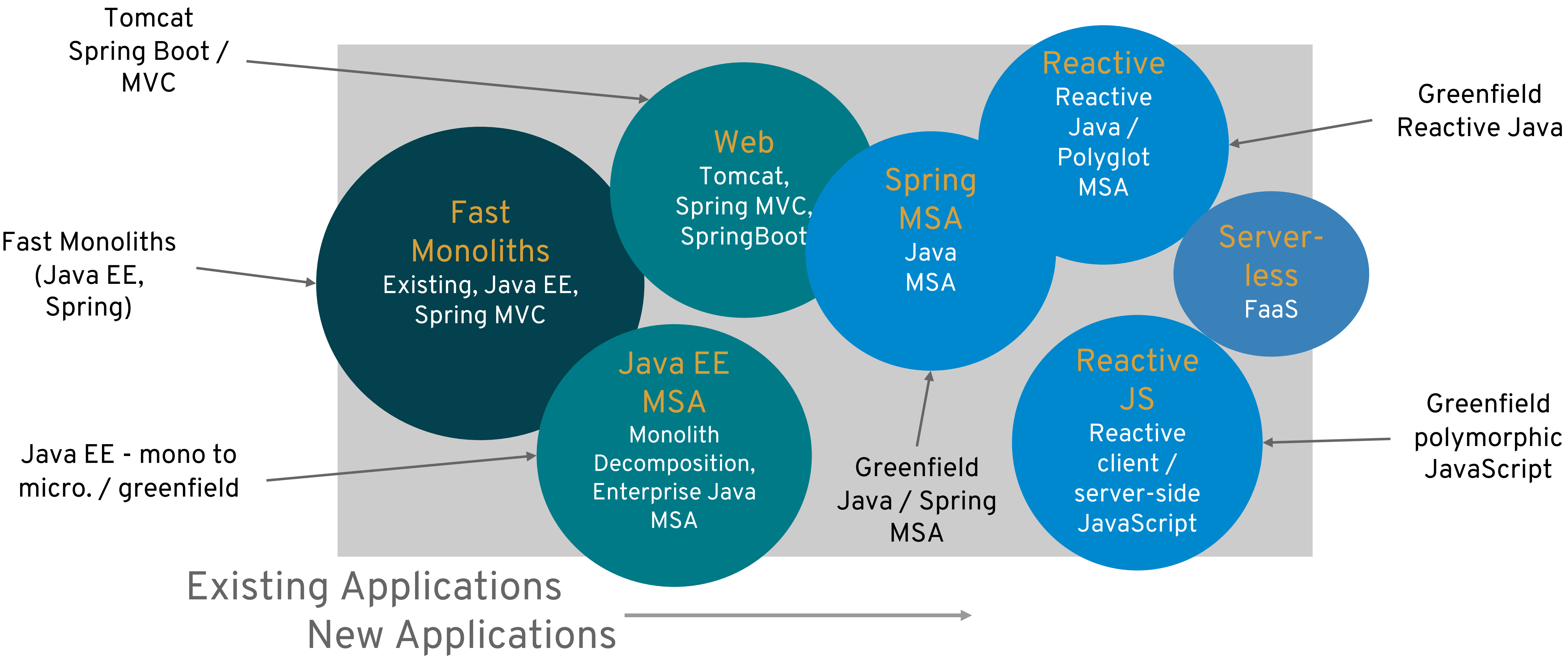
DEPLOY YOUR APP BINARY

DEPLOY YOUR CONTAINER IMAGE


Code
(Git)
Developer
Repository
CODE APPLICATION
Image
Registry
Builder
Image
S2I
BUILD IMAGE
Application
Image
DEPLOY

Module: Optimizing Existing Applications

THE SPECTRUM OF ENTERPRISE APPS



Tomcat
Spring Boot / MVC

Fast Monoliths (Java EE, Spring)

Java EE - mono to micro. / greenfield

Fast Monoliths
Existing, Java EE, Spring MVC

Web
Tomcat, Spring MVC, SpringBoot

Java EE MSA
Monolith Decomposition, Enterprise Java MSA

Spring MSA
Java MSA

Greenfield
Java / Spring MSA

Reactive
Reactive Java / Polyglot MSA

Server-less
FaaS

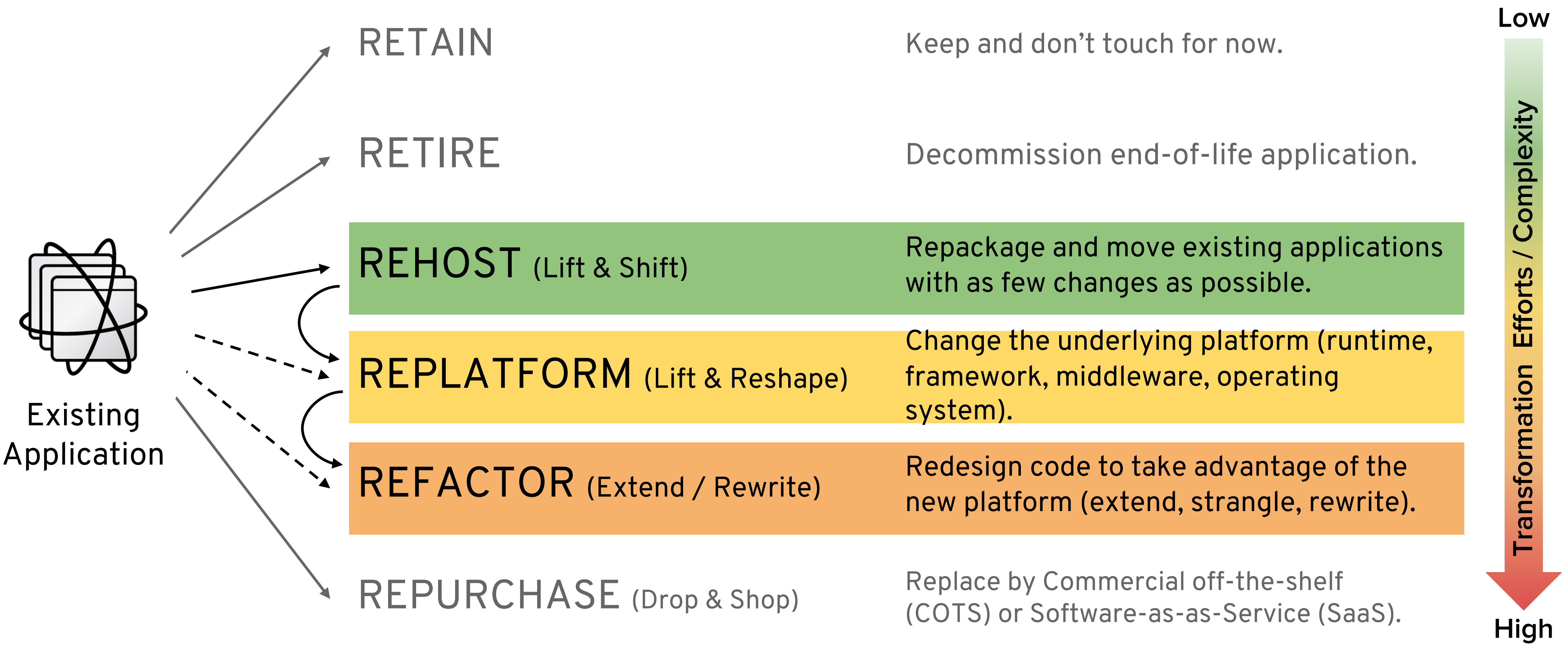
Reactive JS
Reactive client / server-side JavaScript

Greenfield
Reactive Java

Greenfield
polymorphic JavaScript

Existing Applications
New Applications

OPTIONS FOR CONTAINERIZATION



Existing Application

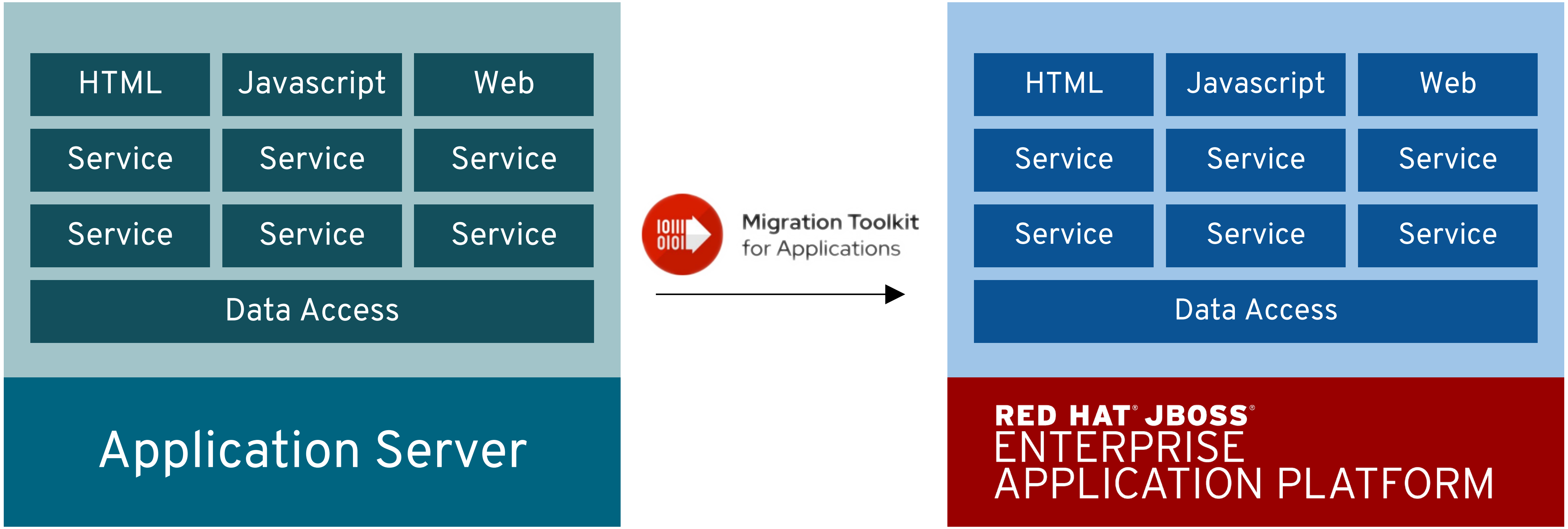
RETAIN	Keep and don’t touch for now.
RETIRE	Decommission end-of-life application.
REHOST (Lift & Shift)	Repackage and move existing applications with as few changes as possible.
REPLATFORM (Lift & Reshape)	Change the underlying platform (runtime, framework, middleware, operating system).
REFACTOR (Extend / Rewrite)	Redesign code to take advantage of the new platform (extend, strangle, rewrite).
REPURCHASE (Drop & Shop)	Replace by Commercial off-the-shelf (COTS) or Software-as-as-Service (SaaS).

Low

Transformation Efforts / Complexity

High

LIFT-AND-SHIFT MONOLITH TO CLOUD



HTML | Javascript | Web

Service | Service | Service

Service | Service | Service

Data Access

Application Server

Migration Toolkit for Applications

HTML | Javascript | Web

Service | Service | Service

Service | Service | Service

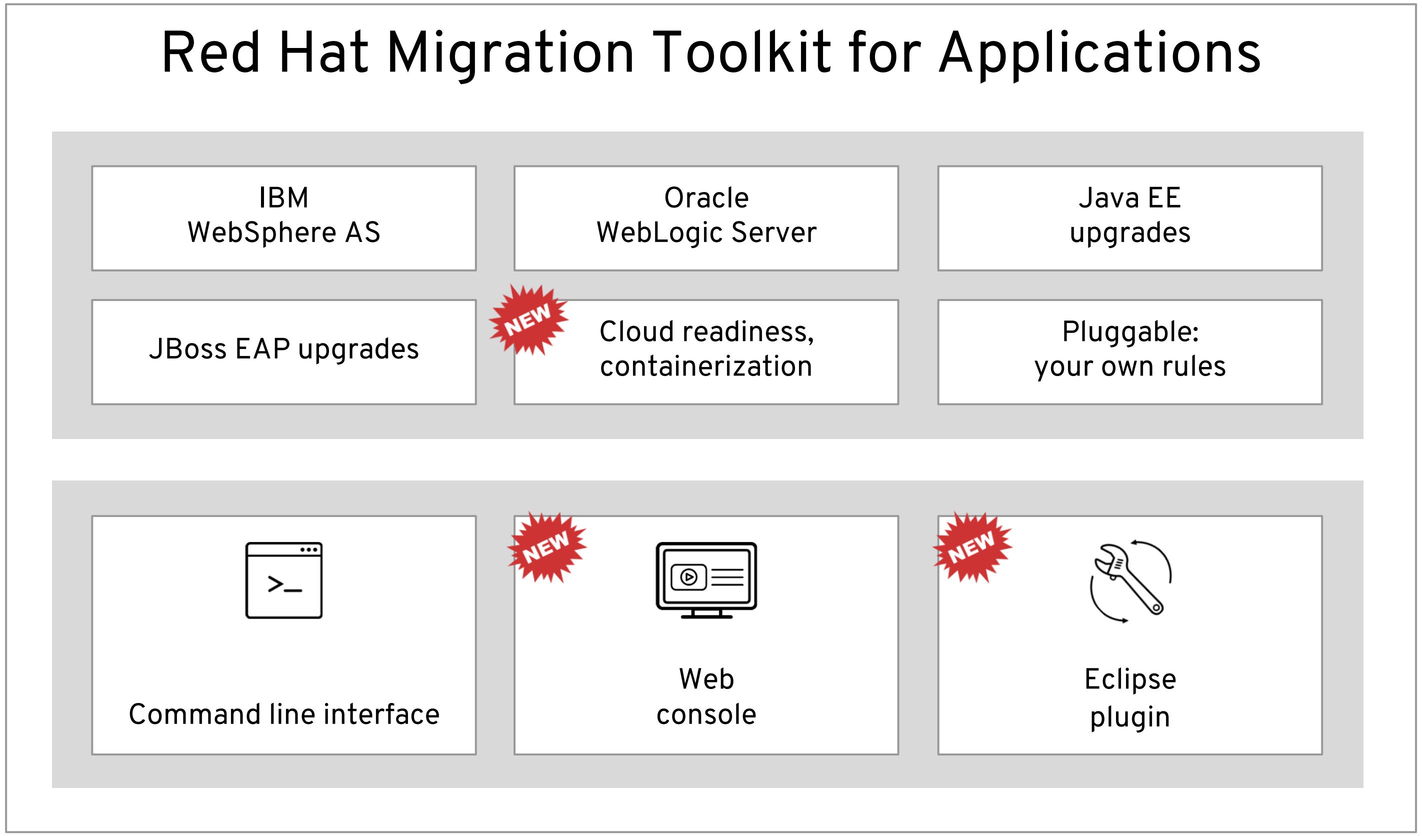
Data Access

RED HAT® JBOSS® ENTERPRISE APPLICATION PLATFORM

Migration Toolkit for Applications

Catalyze large scale application modernizations and migrations

- Automate analysis
- Support effort estimation
- Accelerate code migration
- Free & Open Source

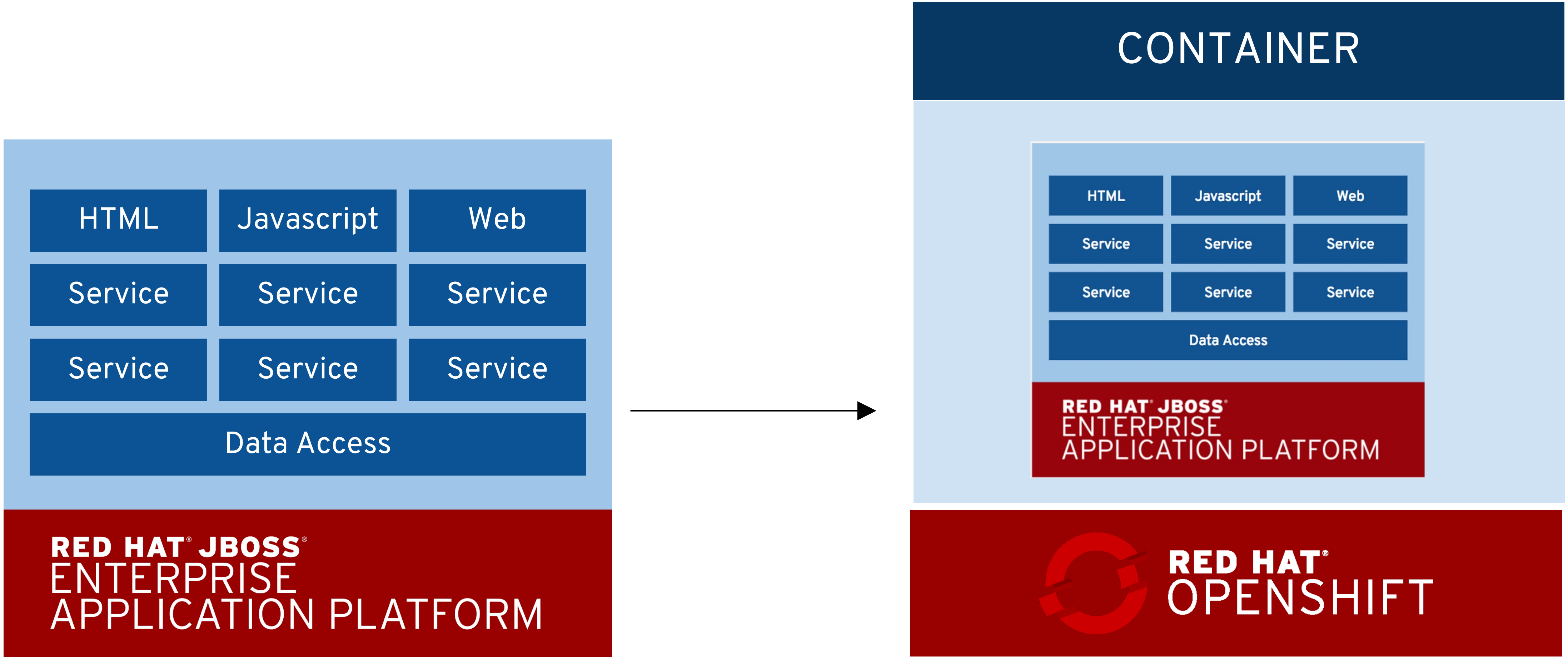


Red Hat Migration Toolkit for Applications

IBM WebSphere AS	Oracle WebLogic Server	Java EE upgrades
JBoss EAP upgrades	NEW Cloud readiness, containerization	Pluggable: your own rules

Command line interface	NEW Web console	NEW Eclipse plugin

LIFT-AND-SHIFT MONOLITH TO CLOUD



HTML | Javascript | Web

Service | Service | Service

Service | Service | Service

Data Access

RED HAT® JBOSS®
ENTERPRISE
APPLICATION PLATFORM

CONTAINER

HTML | Javascript | Web

Service | Service | Service

Service | Service | Service

Data Access

RED HAT® JBOSS®
ENTERPRISE
APPLICATION PLATFORM

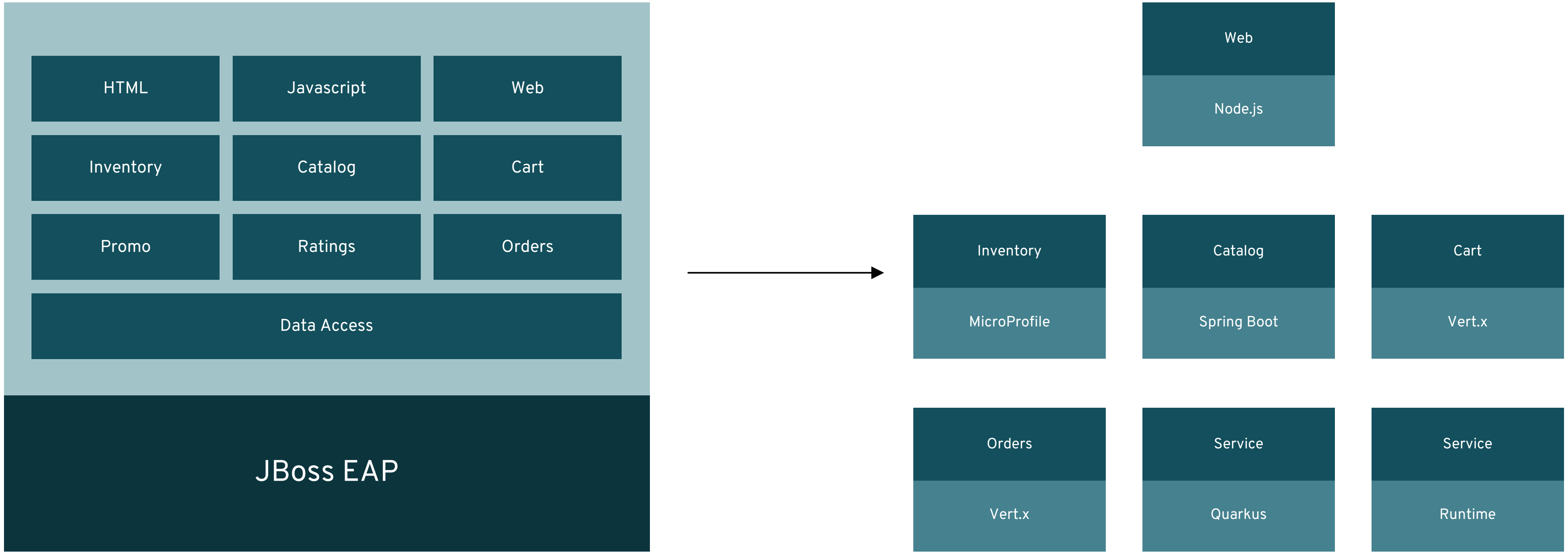
RED HAT®
OPENSHIFT

FAST-MOVING MONOLITHS

- Large organizations have a tremendous amount of resources **invested in existing** monolith applications
- Looking for a **sane way** to capture the benefits of containers and orchestration **without having to complete rewrite**
- **OpenShift** provides the platform for their existing investment with the benefit of a **path forward** for microservice based apps in the future

STRANGLING THE MONOLITH

- In this lab, you will begin to ‘strangle’ the coolstore monolith by implementing its services as external microservices, split along business boundaries
- As functionality is replaced, “dead” parts of monolith can be removed/retired.



CLOUD-NATIVE RUNTIMES

QUARKUS

spring

VERT.X

node JS®

QUARKUS

“MODERN” JAVA APP STACK

MONOLITHS, MICROSERVICES

SINGLE APP

DAYS OF LIFE

100s MB RAM

SECONDS TO START

QUARKUS

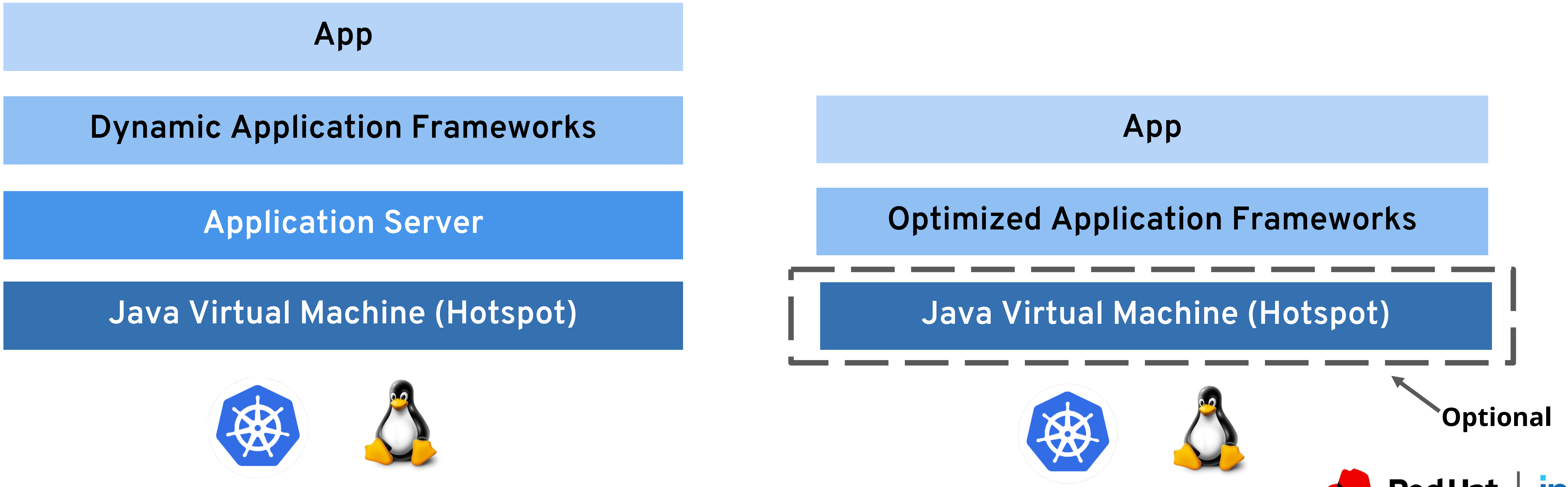
MICROSERVICES, **SERVERLESS**

SINGLE APP

SECONDS TO DAYS OF LIFE

10s MB RAM

MILLISECONDS TO START


App
Dynamic Application Frameworks
Application Server
Java Virtual Machine (Hotspot)
App
Optimized Application Frameworks
Java Virtual Machine (Hotspot)
Optional

spring

- Microservices for Developers using Spring Framework
  - Spring Boot (2.2.6), Spring Core, Spring Data, Spring Web, Spring Security, etc
- An opinionated approach to building Spring applications
- Red Hat Certified with
  - OpenShift Java Runtime
  - JBoss Web Server (Tomcat) embedded web container
- **Can also use Spring APIs in Quarkus**

GOAL FOR LAB

In this lab you will learn:

- How to use lab environment for today
- How to migrate an existing legacy Java EE application (CoolStore) from Weblogic to **JBoss EAP** using **Red Hat Application Migration Toolkit**
- How to deploy the result to **OpenShift container platform** to create a *Fast Moving Monolith*
- Begin modernization journey by breaking subset of monolith into microservices using **Spring Boot** and **Quarkus**

LAB INSTRUCTIONS

- **Everything is done in browser** - no local commands or installs needed on your laptop
- Tested with **Chrome 81.0.4044.138 or later**, **Firefox 60.8.0esr or later**. → **Safari 12.x does not work!**
- If things get weird, just reload browser page
- **Turn off VPN** (we use websockets extensively), **pause AdBlock** for the lab domain (there are no ads)
- To recreate the lab locally, visit `https://github.com/redhat-cop/agnosticd/tree/development/ansible/roles/ocp4-workload-ccnrd`
- Everyone should have their own **unique logins,** e.g.: `user45` / `r3dh4t1!`

Get Started at `red.ht/sea-ccn1`

Credentials: `userXX` / `r3dh4t1!`

If you get stuck, raise hand

LAB INSTRUCTIONS

- Get assigned username at red.ht/sea-ccn1
- Password: **r3dh4t1!**

Summary

- Migrated Java EE app from Weblogic to JBoss EAP
- Built Microservices using Quarkus & Spring Boot
- Strangled the monolith

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

- in linkedin.com/company/red-hat
- ▶ youtube.com/user/RedHatVideos
- f facebook.com/redhatinc
- twitter.com/RedHat