Red Hat
OpenShift

CoreOS and CRI-O & Operator Framework

CoreOS

Red Hat Enterprise Linux

	RED HAT® ENTERPRISE LINUX®	RED HAT® ENTERPRISE LINUX CoreOS
	General Purpose OS	**Immutable container host**
BENEFITS	• 10+ year enterprise life cycle • Industry standard security • High performance on any infrastructure • Customizable and compatible with wide ecosystem of partner solutions	• Self-managing, over-the-air updates • Immutable and tightly integrated with OpenShift • Host isolation is enforced via Containers • Optimized performance on popular infrastructure
WHEN TO USE	When customization and integration with additional solutions is required	When cloud-native, hands-free operations are a top priority

Immutable Operating System

Red Hat Enterprise Linux CoreOS is versioned with OpenShift

CoreOS is tested and shipped in conjunction with the platform. Red Hat runs thousands of tests against these configurations.

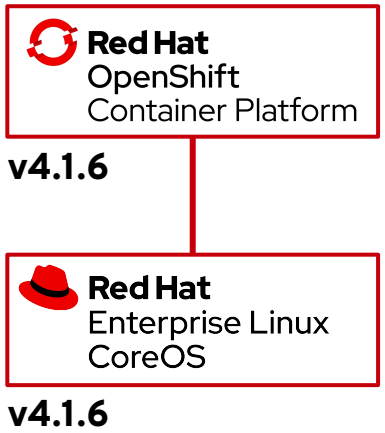
Red Hat Enterprise Linux CoreOS is managed by the cluster

The Operating system is operated as part of the cluster, with the config for components managed by Machine Config Operator:

- CRI-O config
- Kubelet config
- Authorized registries
- SSH config

RHEL CoreOS admins are responsible for:

Nothing. 😀🙌



cri-o

A lightweight, OCI-compliant container runtime

Minimal and Secure Architecture	Optimized for Kubernetes	Runs any OCI-compliant image (including docker)

CRI-O Support in OpenShift

CRI-O tracks and versions identical to Kubernetes, simplifying support permutations

CRI-O 1.12  Kubernetes 1.12  OpenShift 4.0 

CRI-O 1.13  Kubernetes 1.13  OpenShift 4.1 

CRI-O 1.14  Kubernetes 1.14  OpenShift 4.2 

Red Hat

podman

podman

A docker-compatible CLI for containers

- Remote management API via Varlink
- Image/container tagging
- Advanced namespace isolation

buildah

buildah

Secure & flexible OCI container builds

- Integrated into OCP build pods
- Performance improvements for knative enablement
- Image signing improvements

HandsOn Part two

“Build an app with Buildah”

Welcome!

Building Container Images with Buildah

Difficulty: Beginner

Estimated Time: 10 minutes

buildah

Buildah - a tool that facilitates building OCI container images

Buildah creates OCI container images without requiring a Docker Daemon.

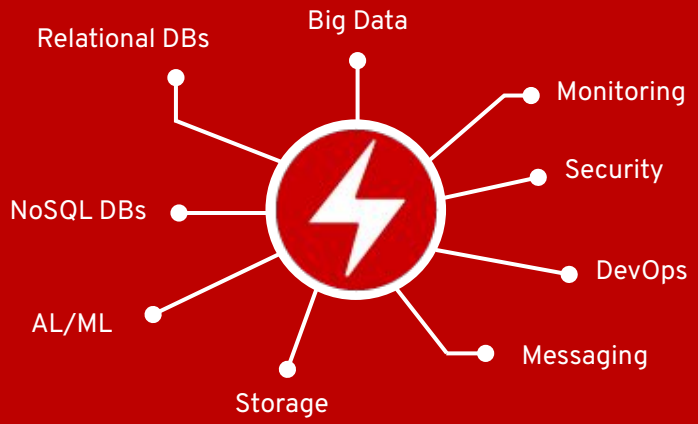
In this scenario you will learn how to build images based on existing Dockerfiles using Buildah. You will also learn how to build images from bash scripts allowing for different syntax and approaches to image creation.

The scenario also introduces skopeo, a tool for inspecting images and container registries

Red Hat

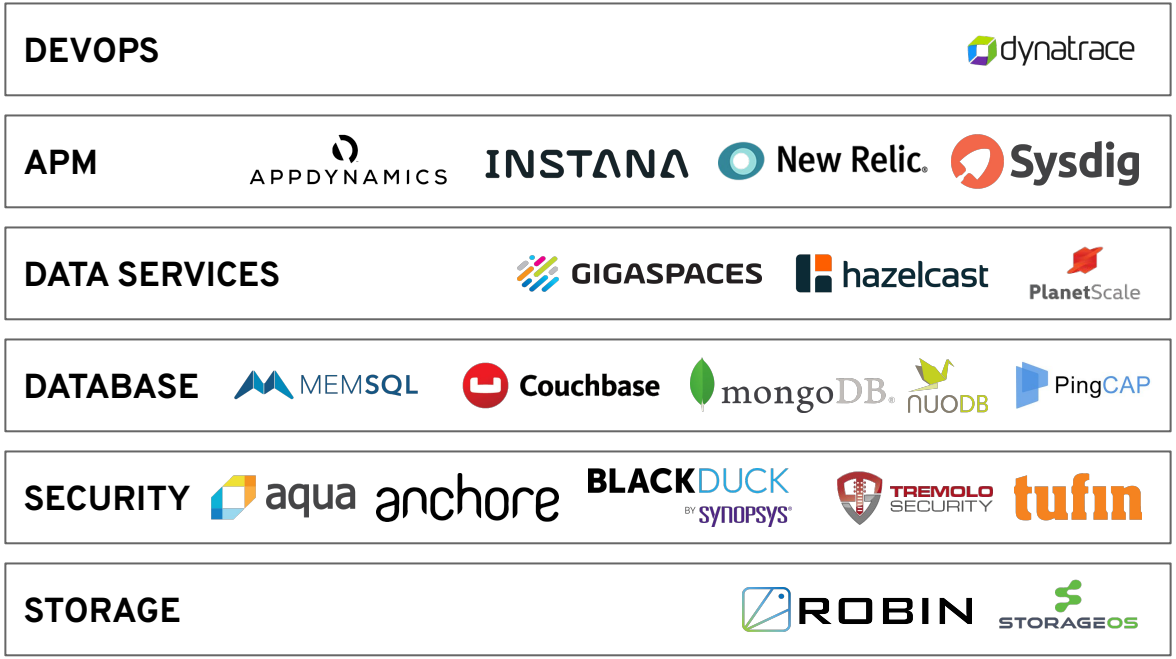
A broad ecosystem of workloads

Operator-backed services allow for a
SaaS experience on your own infrastructure



Red Hat

Red Hat Certified Operators



OperatorHub data sources

Requires an online cluster

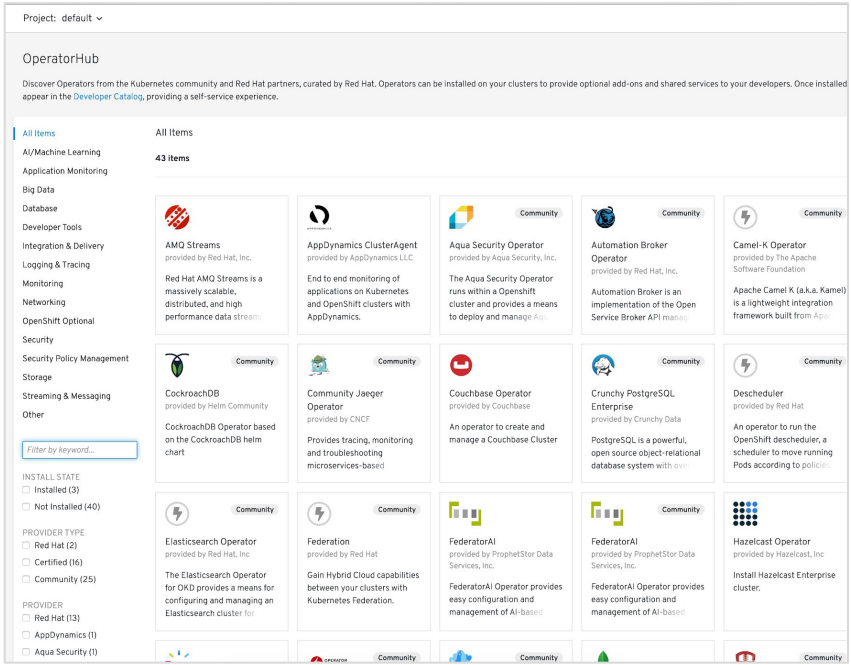
- For 4.1, the cluster must have connectivity to the internet
- Later 4.x releases will add offline capabilities

Operator Metadata

- Stored in quay.io
- Fetches channels and available versions for each Operator

Container Images

- Red Hat products and certified partners come from RHCC
- Community content comes from a variety of registries



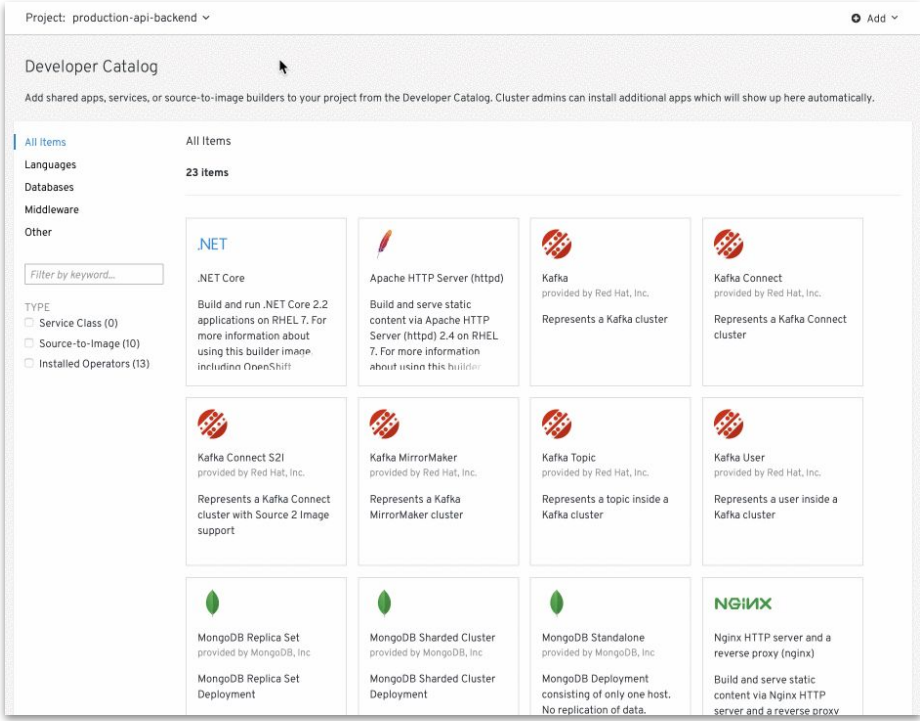
Services ready for your developers

New Developer Catalog aggregates apps

- Blended view of Operators, Templates and Broker backed services
- Operators can expose multiple CRDs. Example:
  - MongoDBReplicaSet
  - MongoDBSharded Cluster
  - MongoDBStandalone
- Developers can't see any of the admin screens

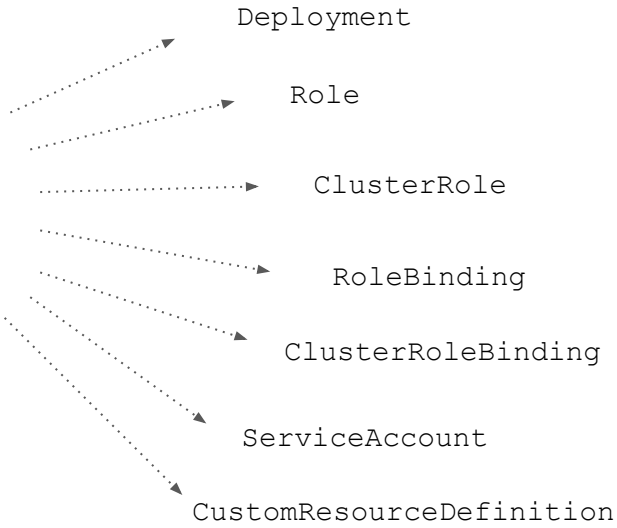
Self-service is key for productivity

- Developers with access can change settings and test out new services at any time



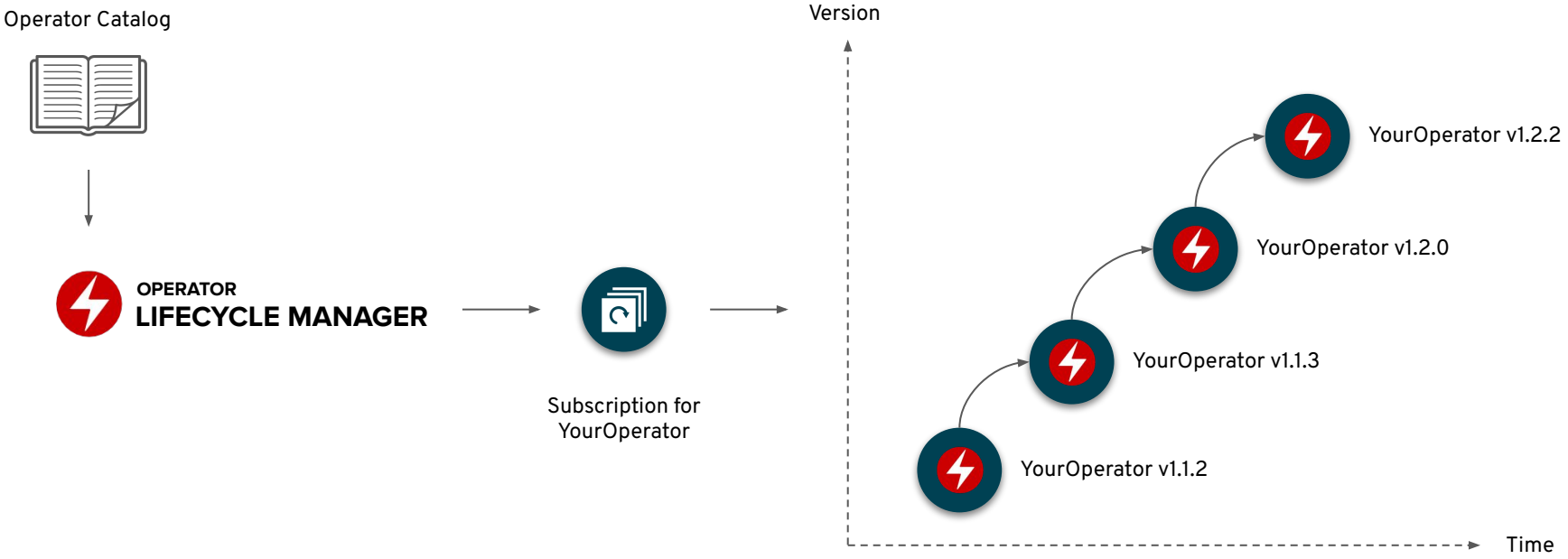
Operators as a First-Class Citizen

YourOperator v1.1.2
Bundle

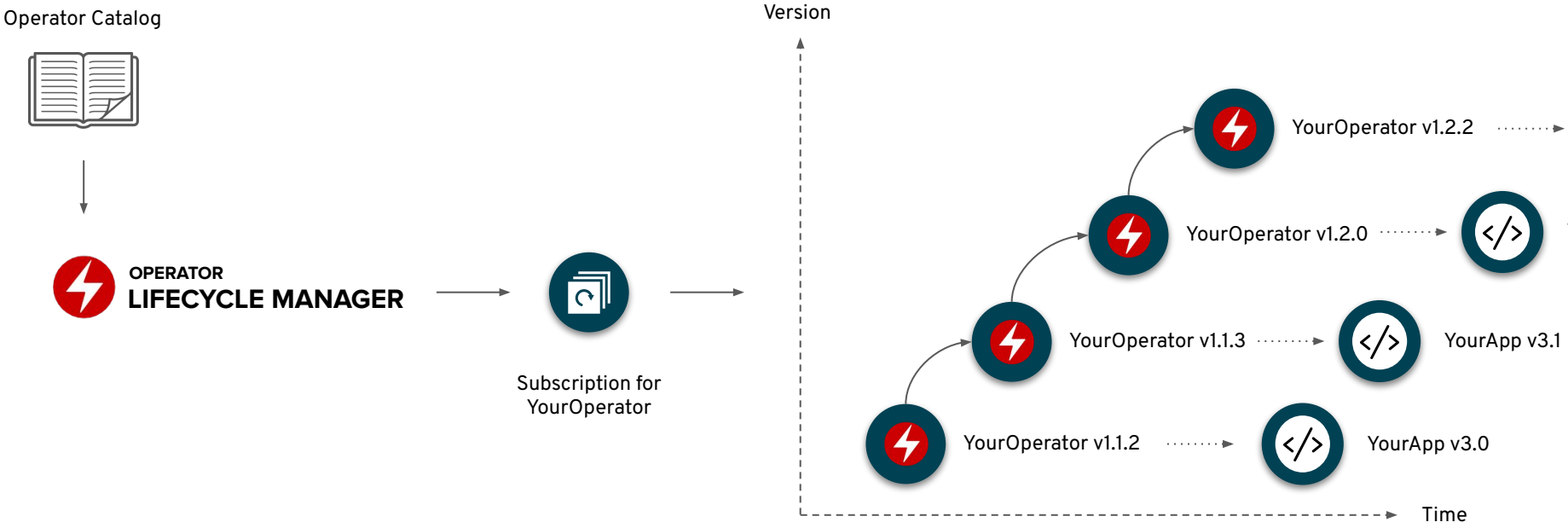


Operator Deployment
Custom Resource
Definitions
RBAC
API Dependencies
Update Path
Metadata

Operator Lifecycle Management



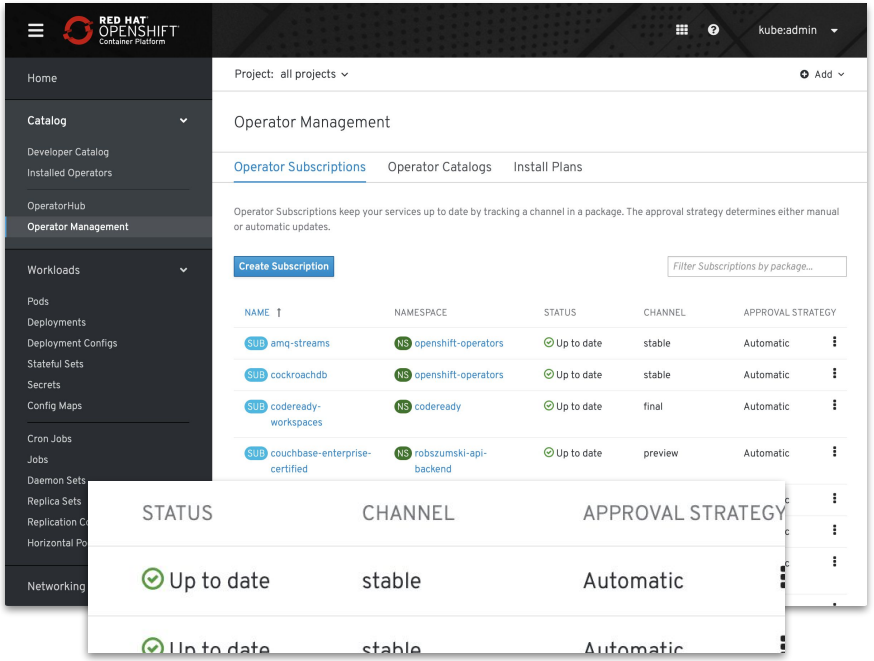
Operator Lifecycle Management



Operator Upgrade in Detail

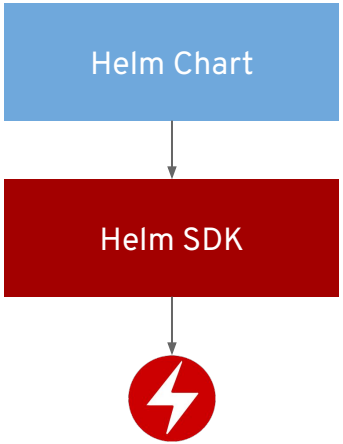
OperatorHub facilitates upgrades of installed Operators

- Manual or automatic modes can be chosen per Operator
- The Operator itself is upgraded by OLM via Deployment and a regular rolling upgrade
- The objects managed by the Operator use built in mechanisms to maintain HA
  - Deployments/StatefulSets
  - affinity/anti-affinity
  - taints/tolerations
  - PodDisruptionBudgets
- Behavior is dependent on the maturity of the Operator
- Optional cluster components like Cluster Logging are well behaved during upgrades

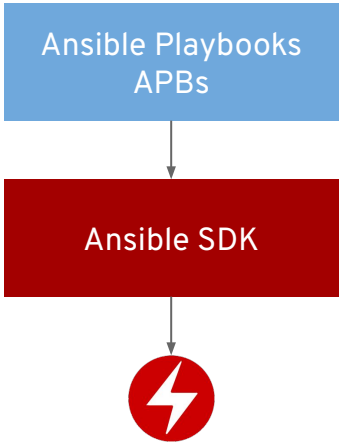


OPERATOR SDK

Build Operators for your apps



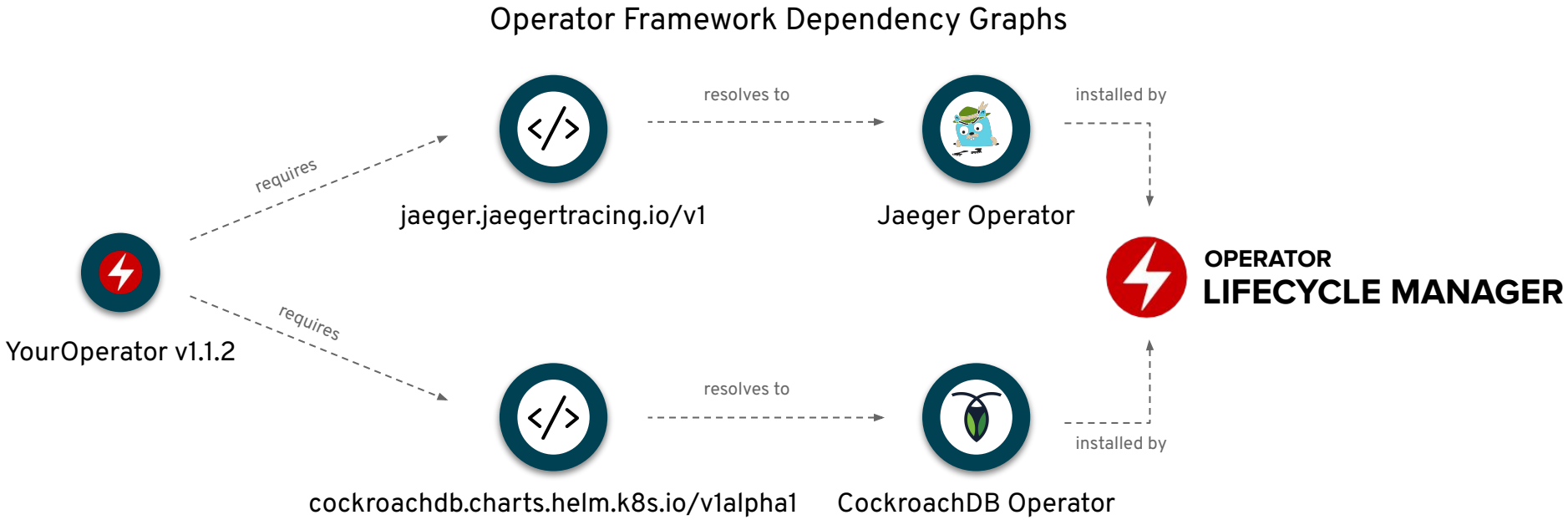
Build operators from Helm chart, without any coding



Build operators from Ansible playbooks and APBs

Build advanced operators for full lifecycle management

Depend on other Operators



Red Hat Middleware

Same experience as 3.x for developers

- Admins install Service Brokers via OperatorHub
- Devs consume via Developer Catalog

Transitioning to Operators

- First Operators are out
  - AMQ Streams (Kafka)
  - Fuse Online
  - CodeReady Workspaces
  - Business Automation (Tech Preview)
  - Data Grid
- More to follow in 2019
  - Red Hat Integration - July:
    - AMQ Interconnect, AMQ Broker
    - 3scale API Management
    - Apicurio API Designer
  - Business Automation - July (GA)
  - Red Hat Application Runtimes
    - MW Component Operator - July

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat