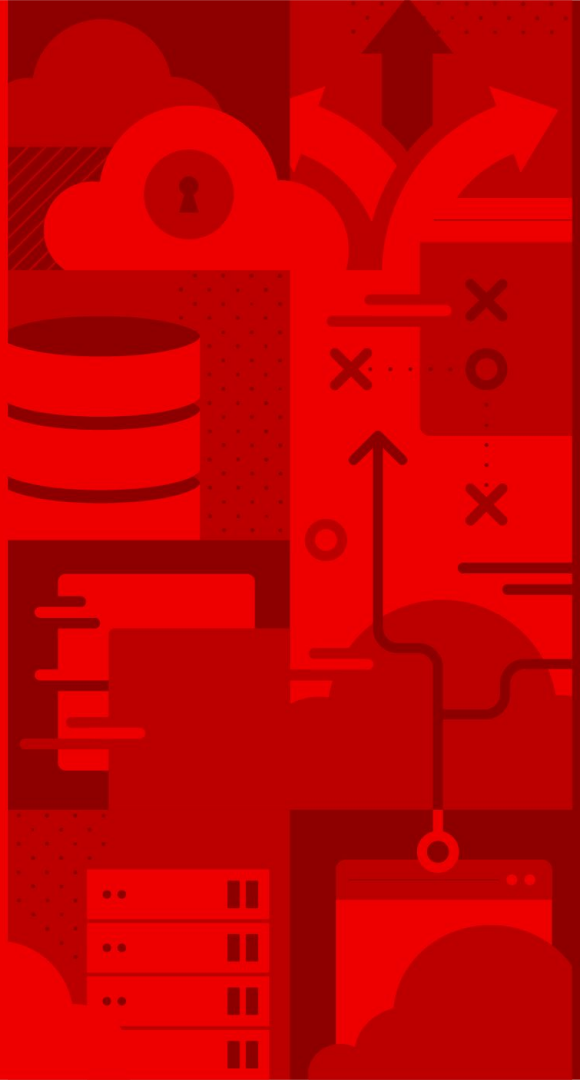


OpenShift Introduction

Red Hat

Hi! I'm Laine Vyvyan.

- I'm a Channel Solutions Architect at Red Hat. I cover all of the Great Lakes states.
- I live in Lansing, MI.
- My favorite color is *glitter*.

To get in touch with me:

lvyvyan@redhat.com

Twitter, etc: lainie_ftw

Red Hat

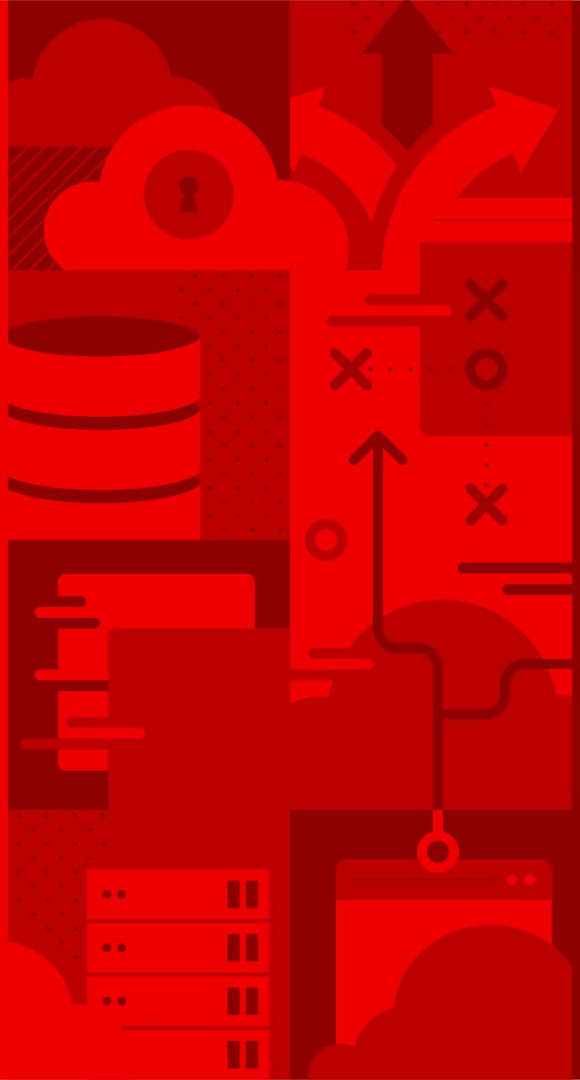
AGENDA

Overview of Containers

Overview of OpenShift Architecture

Developer Enablement with OpenShift

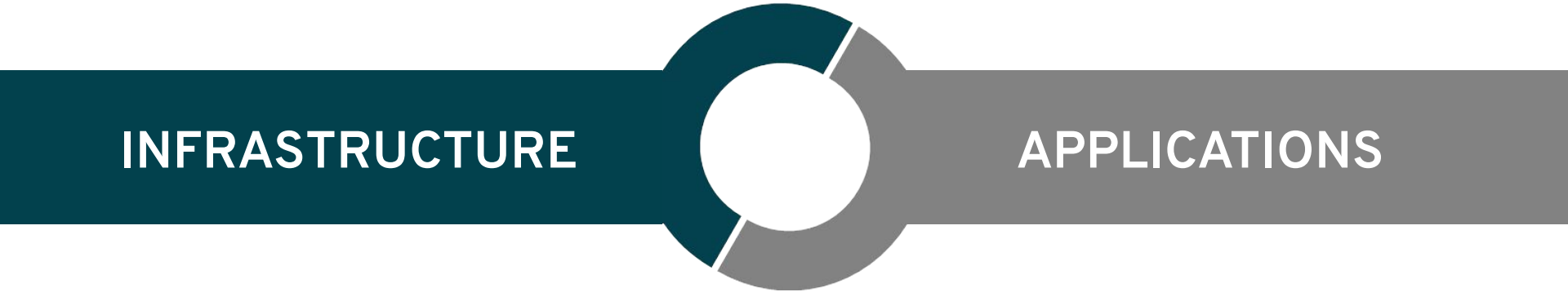
Demo!



Overview of Containers

WHAT ARE CONTAINERS?

Well...it depends on who you ask...



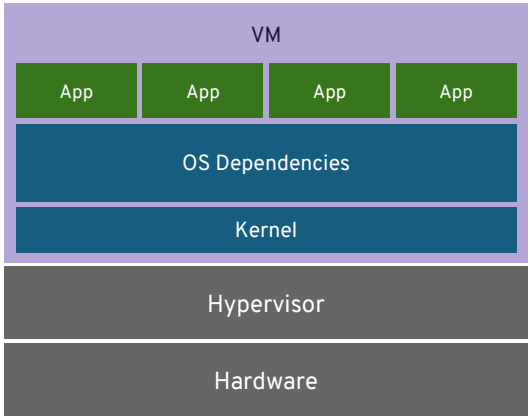
- Application processes on a shared kernel
- Simpler, lighter, and denser than VMs
- Portable across different environments

- Package apps with all dependencies
- Deploy to any environment in seconds
- Easily accessed and shared

Turns out...both are correct!

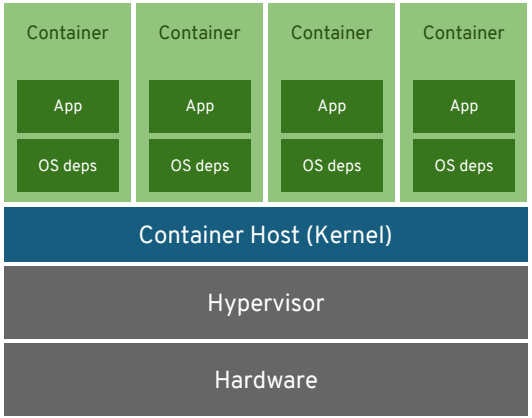
VIRTUAL MACHINES VS. CONTAINERS

VIRTUAL MACHINES



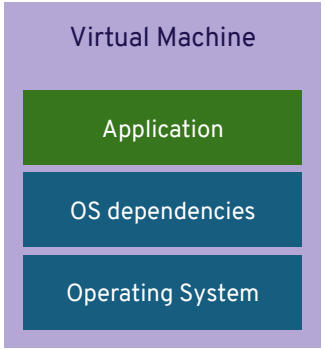
VM abstracts the hardware

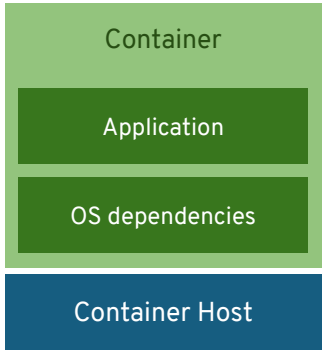
CONTAINERS



Container isolates processes:
abstracts OS *and* hardware

VIRTUAL MACHINES VS. CONTAINERS



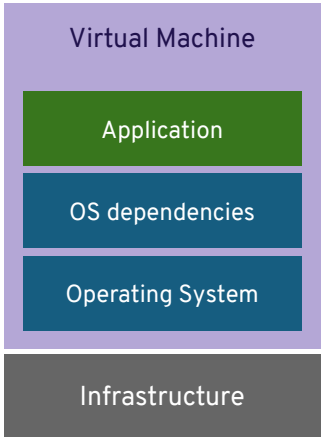


- ✚ VM Isolation
- ▬ Complete OS
- ▬ Static Compute
- ▬ Static Memory
- ▬ High Resource Usage
- ▬ VM Drift

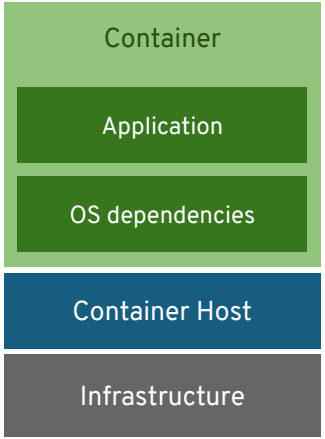
- ✚ Container Isolation
- ✚ Shared Kernel
- ✚ Burstable Compute
- ✚ Burstable Memory
- ✚ Low Resource Usage
- ✚ Build Once, Deploy Anywhere

VIRTUAL MACHINES VS. CONTAINERS

IT Ops
(and Dev, sort of)



Clear ownership boundary between Dev and IT Ops drives DevOps adoption and fosters agility



Dev

IT Ops

Optimized for stability
Optimized for agility

Container Workloads

	Easy
Code	Completely Isolated (single process)
Configuration	One Configuration File
Data	Data Saved in Single Place
Secrets	Static Files
Network	HTTP, HTTPS
Installation	Packages, Source
Licensing	Open Source

Container Workloads

	Easy	Moderate
Code	Completely Isolated (single process)	Somewhat Isolated (multiple processes)
Configuration	One Configuration File	Several Configuration Files
Data	Data Saved in Single Place	Data Saved in Several Places
Secrets	Static Files	Network
Network	HTTP, HTTPS	TCP, UDP
Installation	Packages, Source	Installers (install.sh) and Understood Configuration
Licensing	Open Source	Proprietary

Container Workloads

	Easy	Moderate	Difficult
Code	Completely Isolated (single process)	Somewhat Isolated (multiple processes)	Self Modifying (e.g. Actor Model)
Configuration	One Configuration File	Several Configuration Files	Configuration Anywhere in Filesystem
Data	Data Saved in Single Place	Data Saved in Several Places	Data Anywhere in Filesystem
Secrets	Static Files	Network	Dynamic Generation of Certificates
Network	HTTP, HTTPS	TCP, UDP	IPSEC, Highly Isolated
Installation	Packages, Source	Installers (install.sh) and Understood Configuration	Installers (install.sh)
Licensing	Open Source	Proprietary	Restrictive & Proprietary

Container Workloads

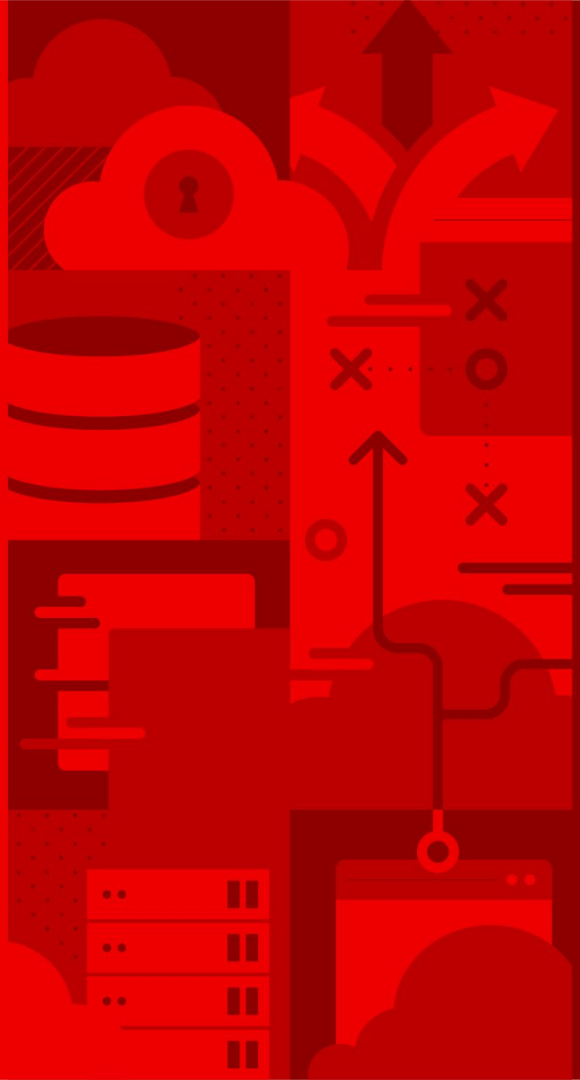
	Greenfield	Moderate	Difficult
Code	Completely Isolated (single process)	Somewhat Isolated (multiple processes)	Self Modifying (e.g. Actor Model)
Configuration	One Configuration File	Several Configuration Files	Configuration Anywhere in Filesystem
Data	Data Saved in Single Place	Data Saved in Several Places	Data Anywhere in Filesystem
Secrets	Static Files	Network	Dynamic Generation of Certificates
Network	HTTP, HTTPS	TCP, UDP	IPSEC, Highly Isolated
Installation	Packages, Source	Installers (install.sh) and Understood Configuration	Installers (install.sh)
Licensing	Open Source	Proprietary	Restrictive & Proprietary

Container Workloads

	Greenfield	Monolith Modernization	Difficult
Code	Completely Isolated (single process)	Somewhat Isolated (multiple processes)	Self Modifying (e.g. Actor Model)
Configuration	One Configuration File	Several Configuration Files	Configuration Anywhere in Filesystem
Data	Data Saved in Single Place	Data Saved in Several Places	Data Anywhere in Filesystem
Secrets	Static Files	Network	Dynamic Generation of Certificates
Network	HTTP, HTTPS	TCP, UDP	IPSEC, Highly Isolated
Installation	Packages, Source	Installers (install.sh) and Understood Configuration	Installers (install.sh)
Licensing	Open Source	Proprietary	Restrictive & Proprietary

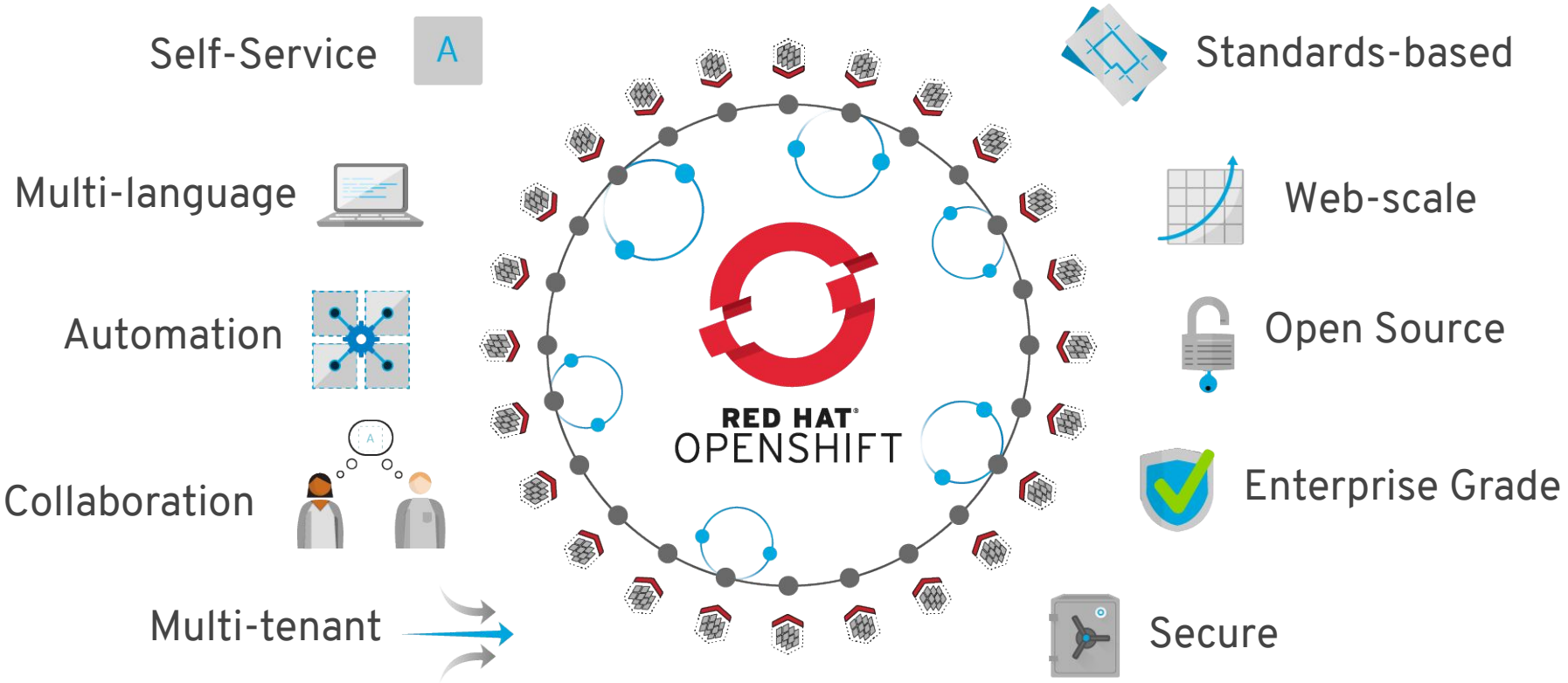
Container Workloads

	Greenfield	Monolith Modernization	COTS or Lift and Shift
Code	Completely Isolated (single process)	Somewhat Isolated (multiple processes)	Self Modifying (e.g. Actor Model)
Configuration	One Configuration File	Several Configuration Files	Configuration Anywhere in Filesystem
Data	Data Saved in Single Place	Data Saved in Several Places	Data Anywhere in Filesystem
Secrets	Static Files	Network	Dynamic Generation of Certificates
Network	HTTP, HTTPS	TCP, UDP	IPSEC, Highly Isolated
Installation	Packages, Source	Installers (install.sh) and Understood Configuration	Installers (install.sh)
Licensing	Open Source	Proprietary	Restrictive & Proprietary



Overview of OpenShift Architecture

Red Hat


Self-Service
Multi-language
Automation
Collaboration
Multi-tenant
RED HAT
OPENSHIFT
Standards-based
Web-scale
Open Source
Enterprise Grade
Secure

Red Hat

CATHAY PACIFIC

10x

Increased application development throughput from 20 to 200 changes a day

OpenShift on AWS & private cloud

Hilton

Months → Days

Improved time to market by accelerating development time

OpenShift on AWS

Schiphol Amsterdam Airport

50%

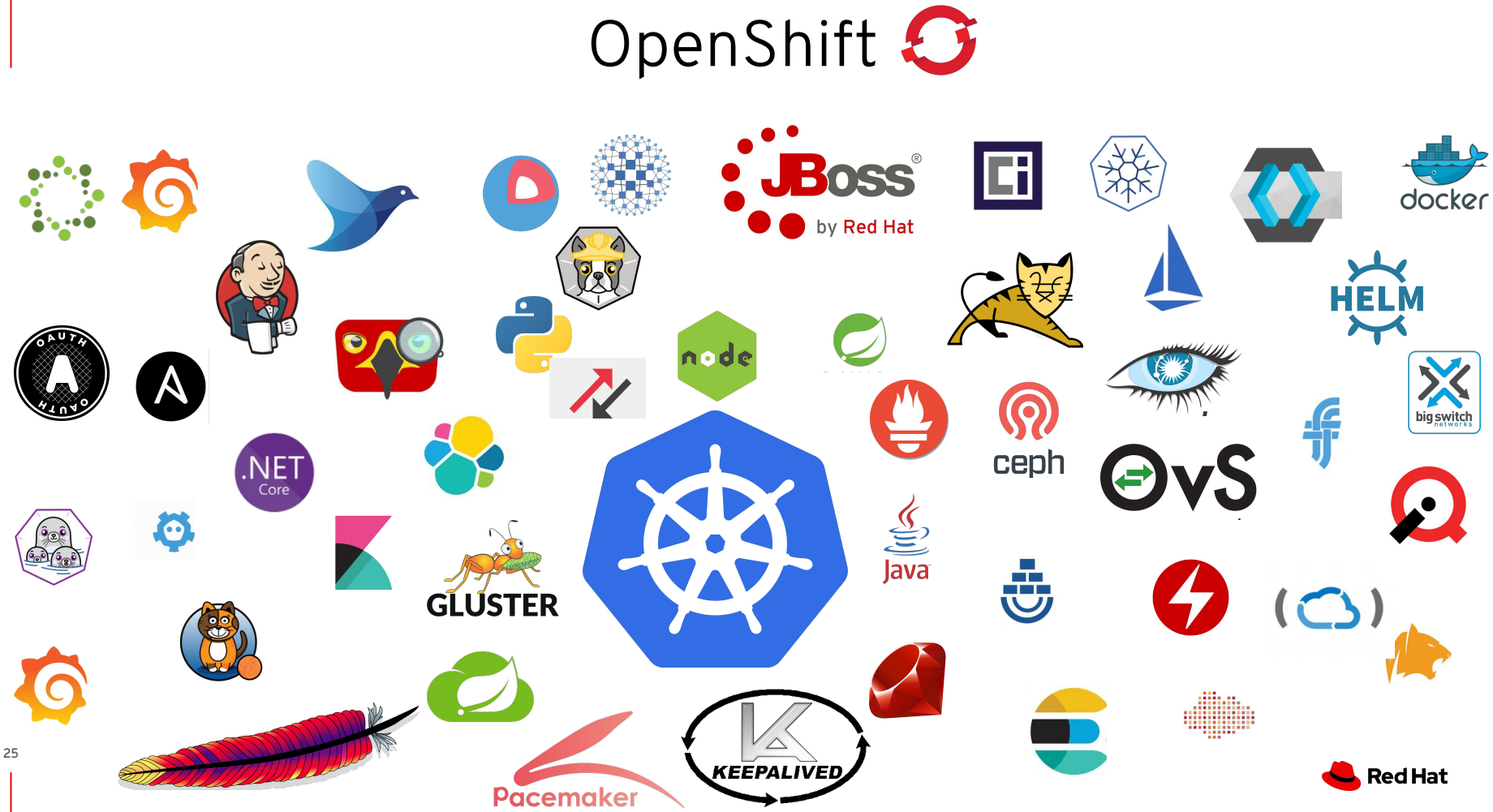
Reduction in development time for new services and APIs. Launched a new cloud platform in **10 days**

OpenShift on AWS, Azure, & private cloud

Source:
Cathay Pacific: Red Hat press release, Cathay Pacific Takes Customer Experiences to New Heights with Red Hat's Hybrid Cloud Technologies, May 2018.
Hilton: Red Hat case study, Hilton enhances digital guest experience with Red Hat container and automation technology, October 2018.
Schiphol: Red Hat case study, Amsterdam Airport Schiphol builds agile cloud with Red Hat, August 2017.

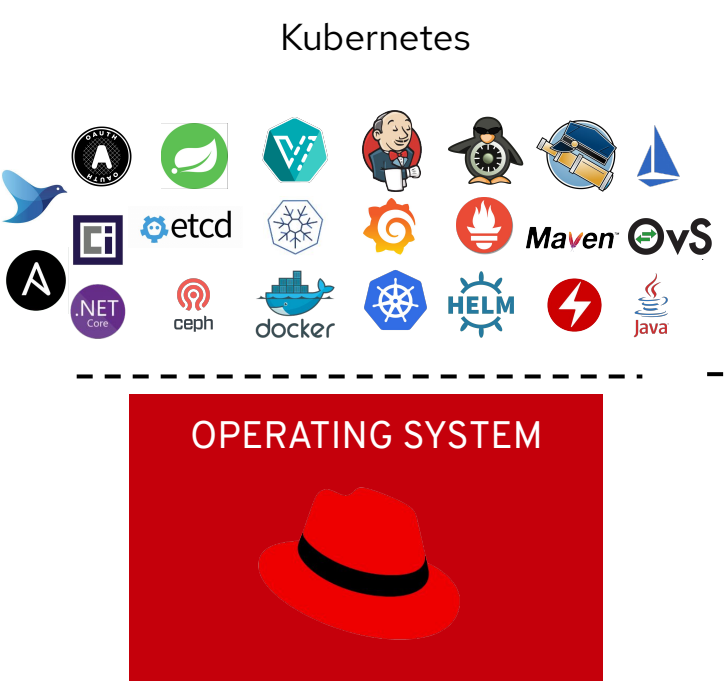
Red Hat

OpenShift



Red Hat

Kubernetes, OpenShift 3, Openshift 4



Single-Image Platform Packaging

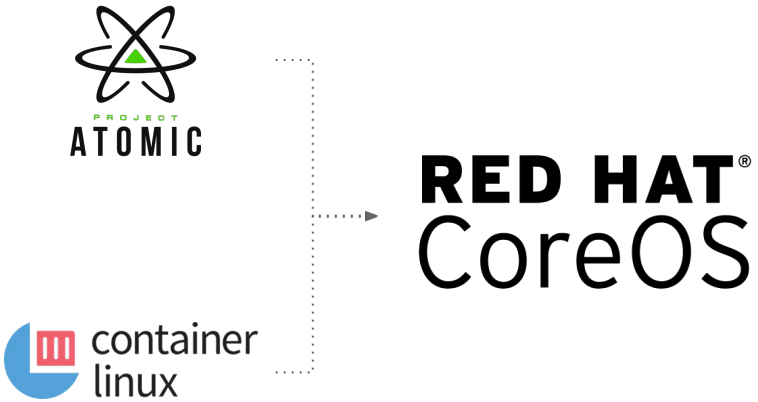
Red Hat

Red Hat Enterprise Linux CoreOS:

The OpenShift Operating System

Red Hat

IMMUTABLE INFRASTRUCTURE WITH RED HAT COREOS



- Minimal Linux distribution
- Optimized for running containers
- Decreased attack surface
- Over-the-air automated updates
- Immutable foundation for OpenShift
- Bare-metal and cloud host configuration

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.

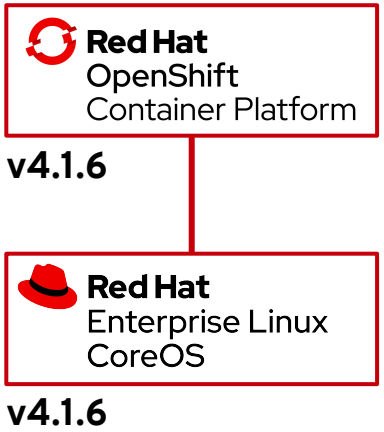
Red Hat Enterprise Linux CoreOS is managed by the *cluster*

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. 😀🙌



Red Hat

Red Hat Enterprise Linux vs RHEL CoreOS

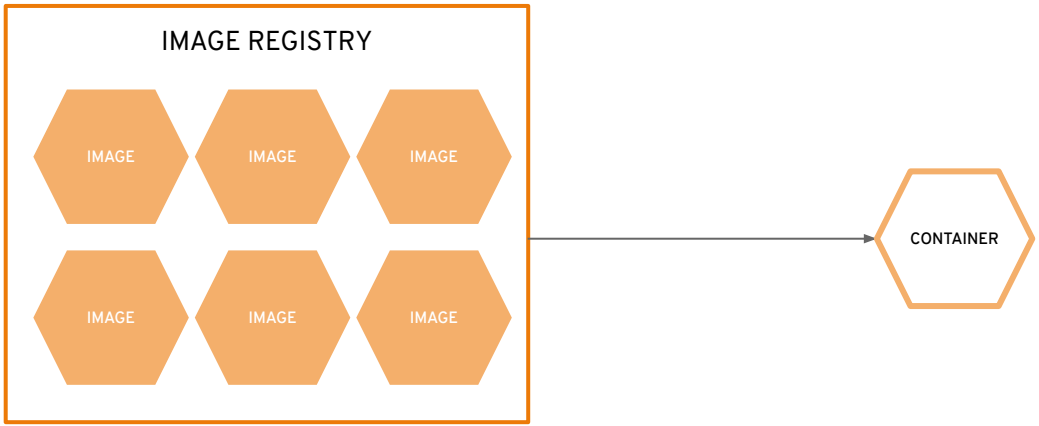
	RED HAT® ENTERPRISE LINUX®	RED HAT® ENTERPRISE LINUX CoreOS
	General Purpose OS	**Immutable container host**
BENEFITS	• 10+ year enterprise life cycle • Industry standard security • High performance on any infrastructure • Customizable and compatible with wide ecosystem of partner solutions	• Self-managing, 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

Red Hat

a container is the smallest compute unit

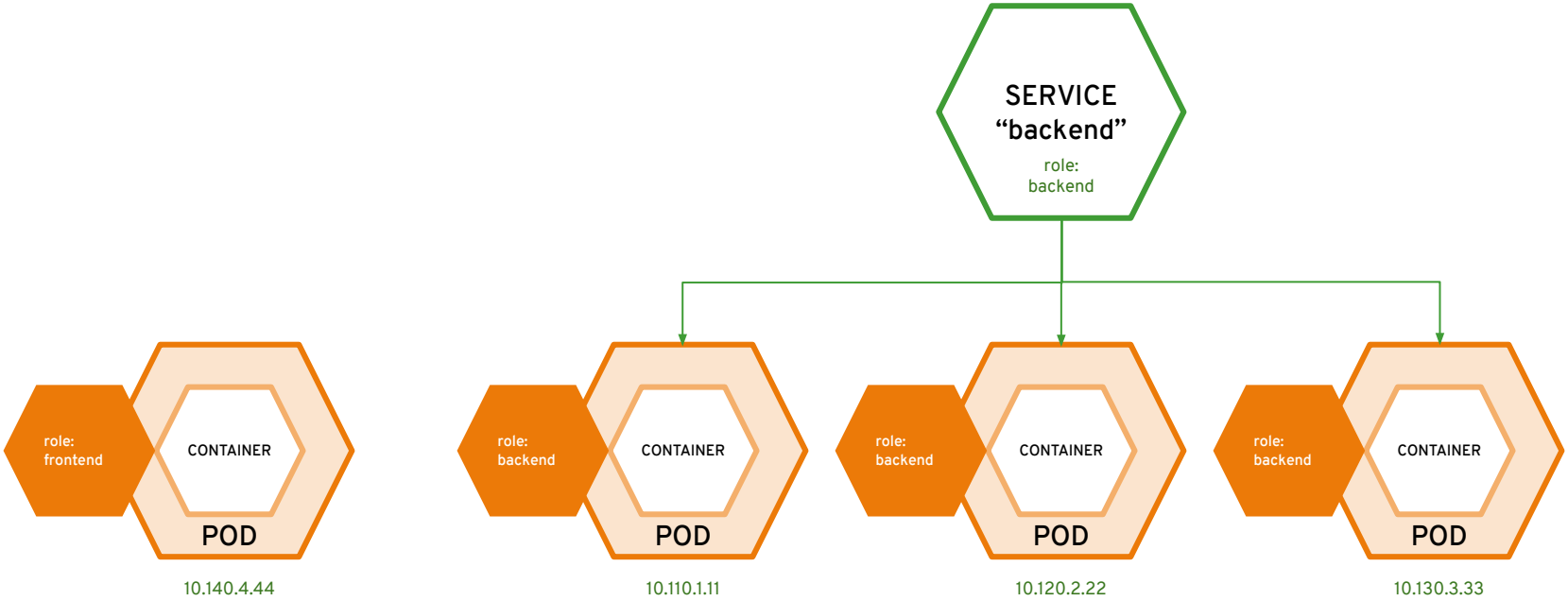
containers are created from container `images`

container `images` are stored in an `image registry`

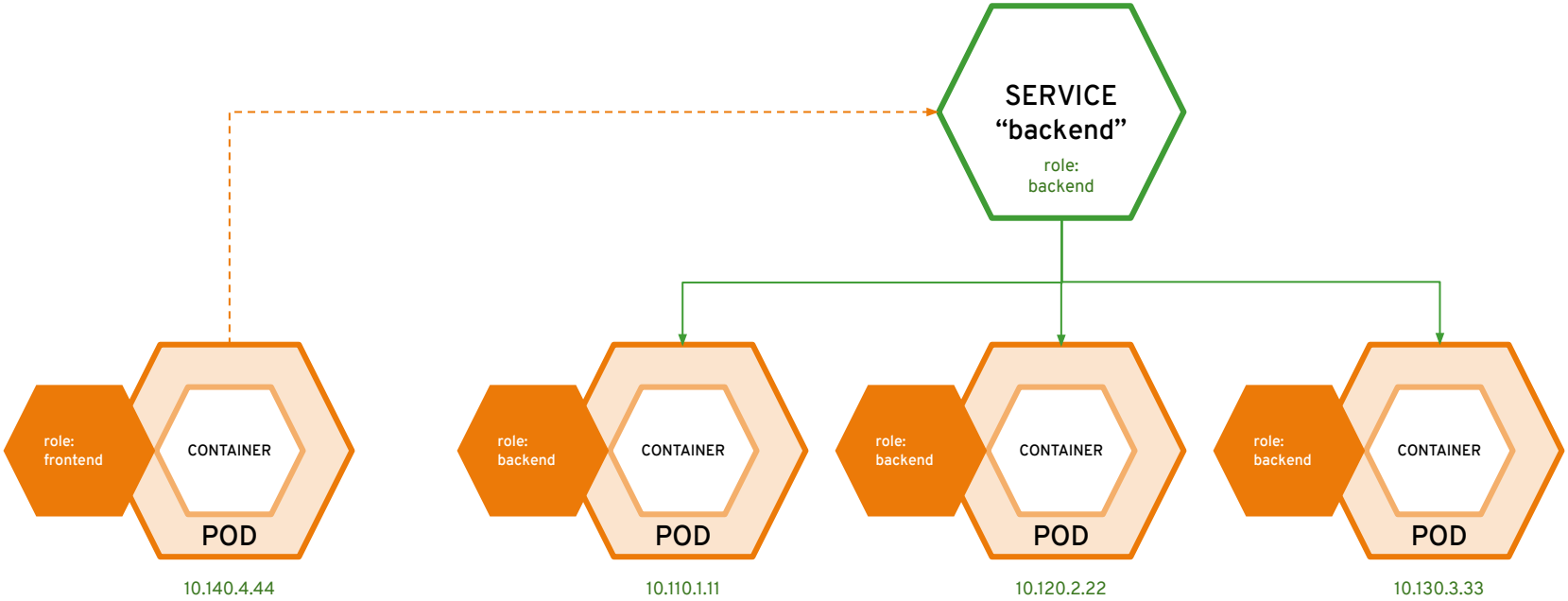


containers are wrapped in `pods` which are units of deployment and management

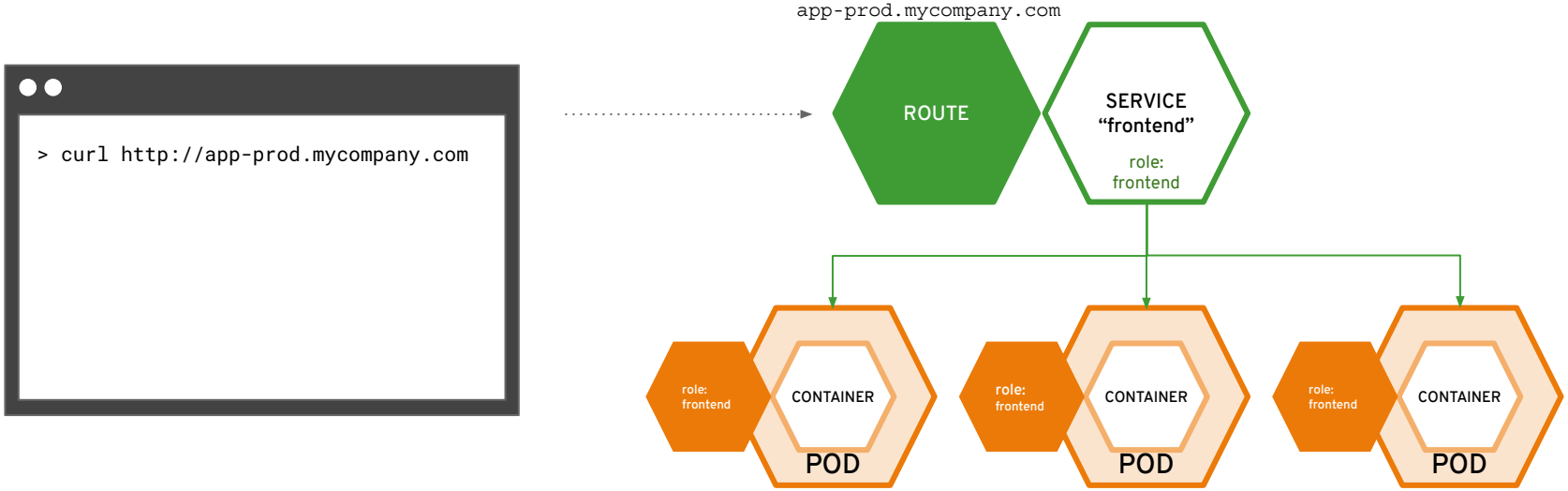
`services` provide internal load-balancing and service discovery across pods



also, apps can talk to *each other* via `services`

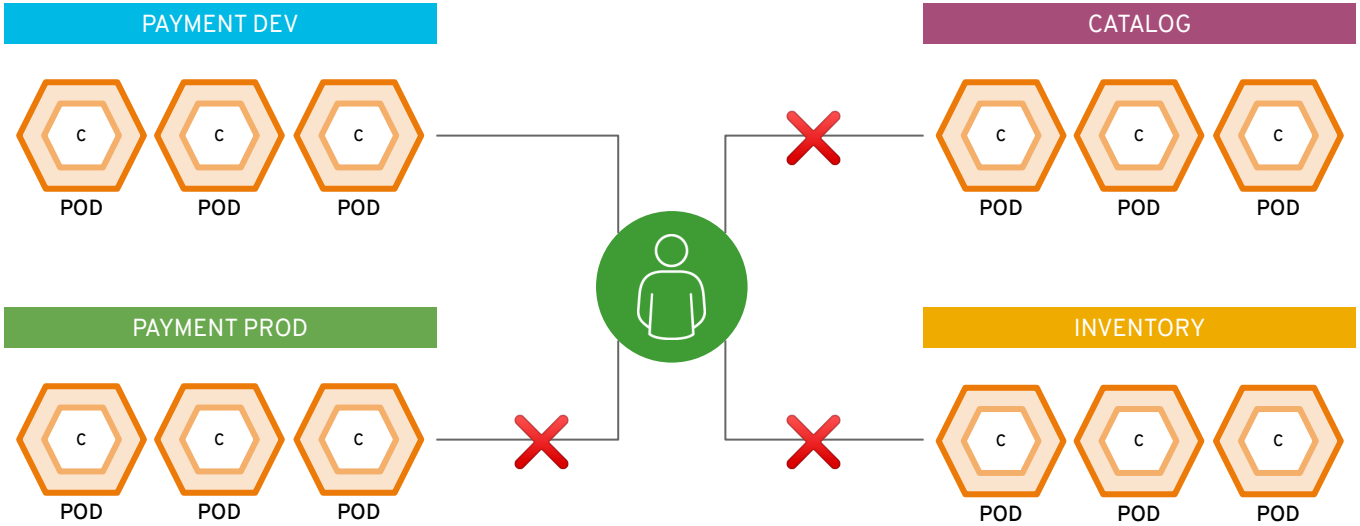


routes make services accessible to clients outside the environment via real-world urls



Red Hat

`projects` isolate **apps** across environments, teams, groups and departments



your choice of infrastructure

COMPUTE

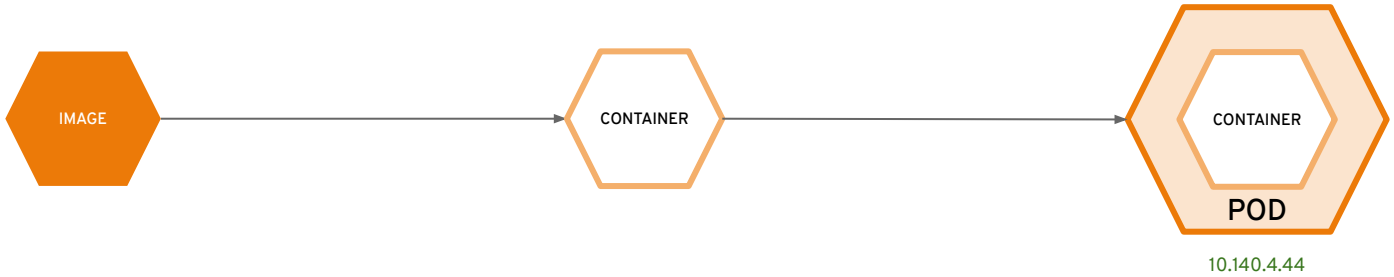
NETWORK

STORAGE

workers run workloads

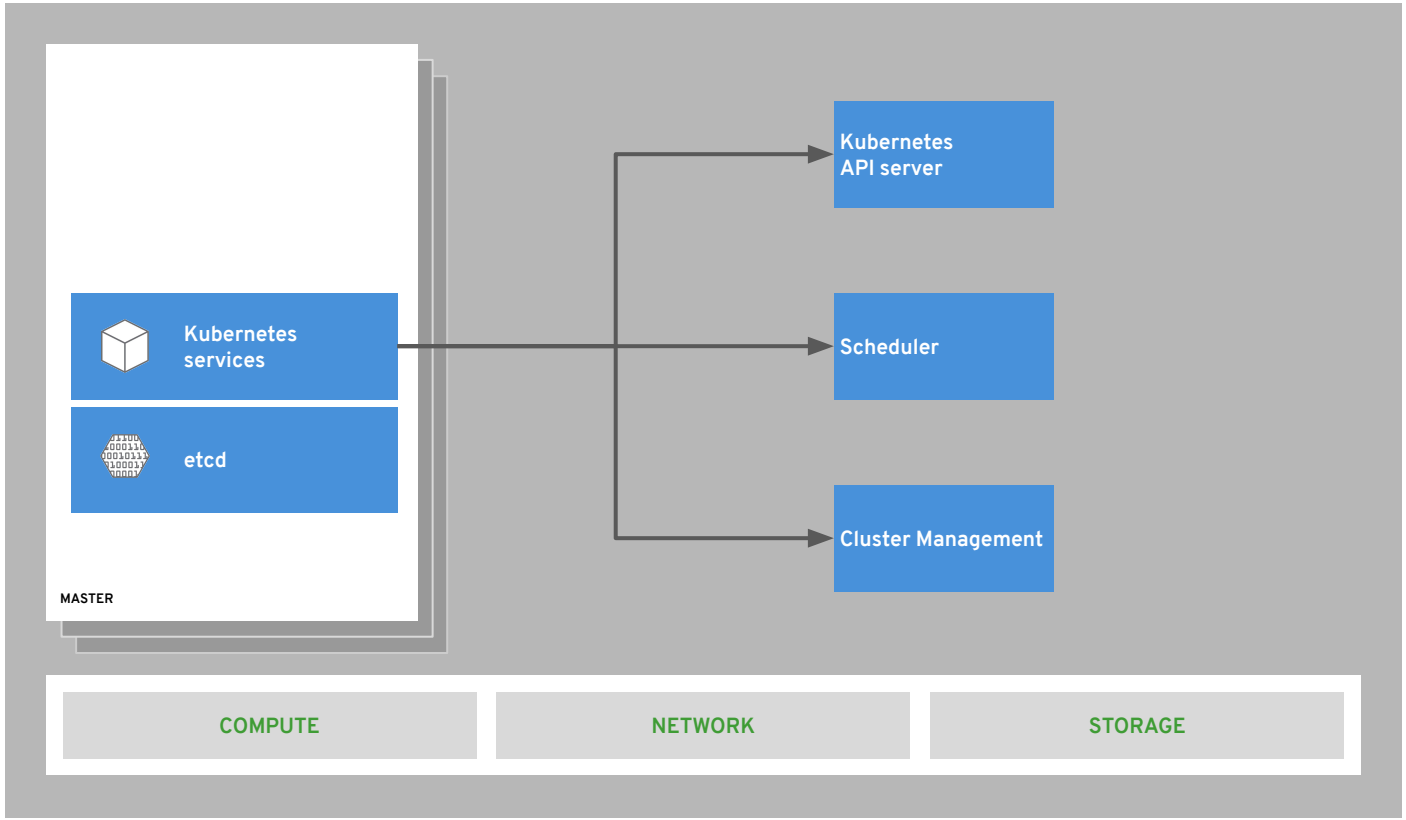
masters are the control plane

everything runs in pods

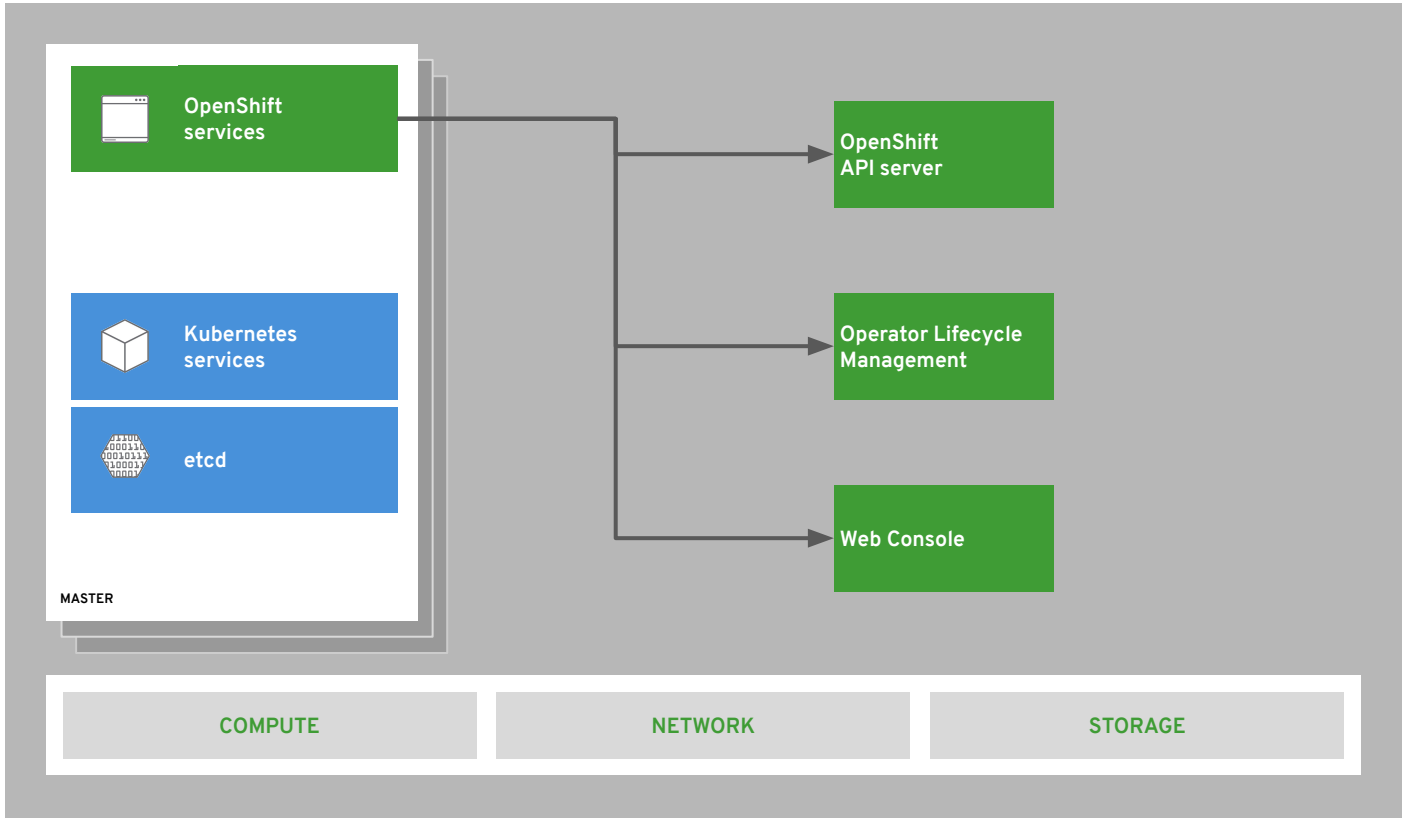


state of everything

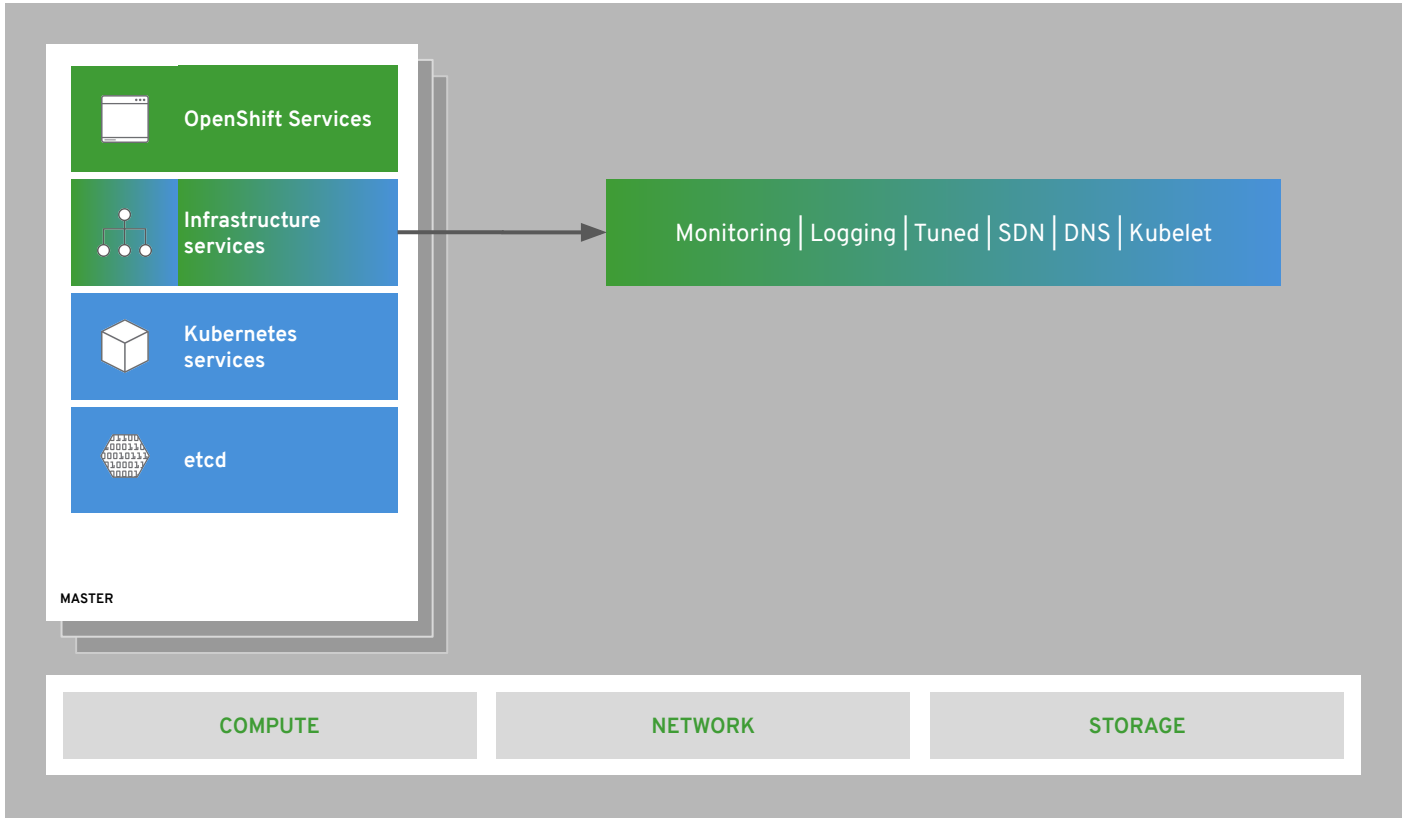
core kubernetes components



core OpenShift components



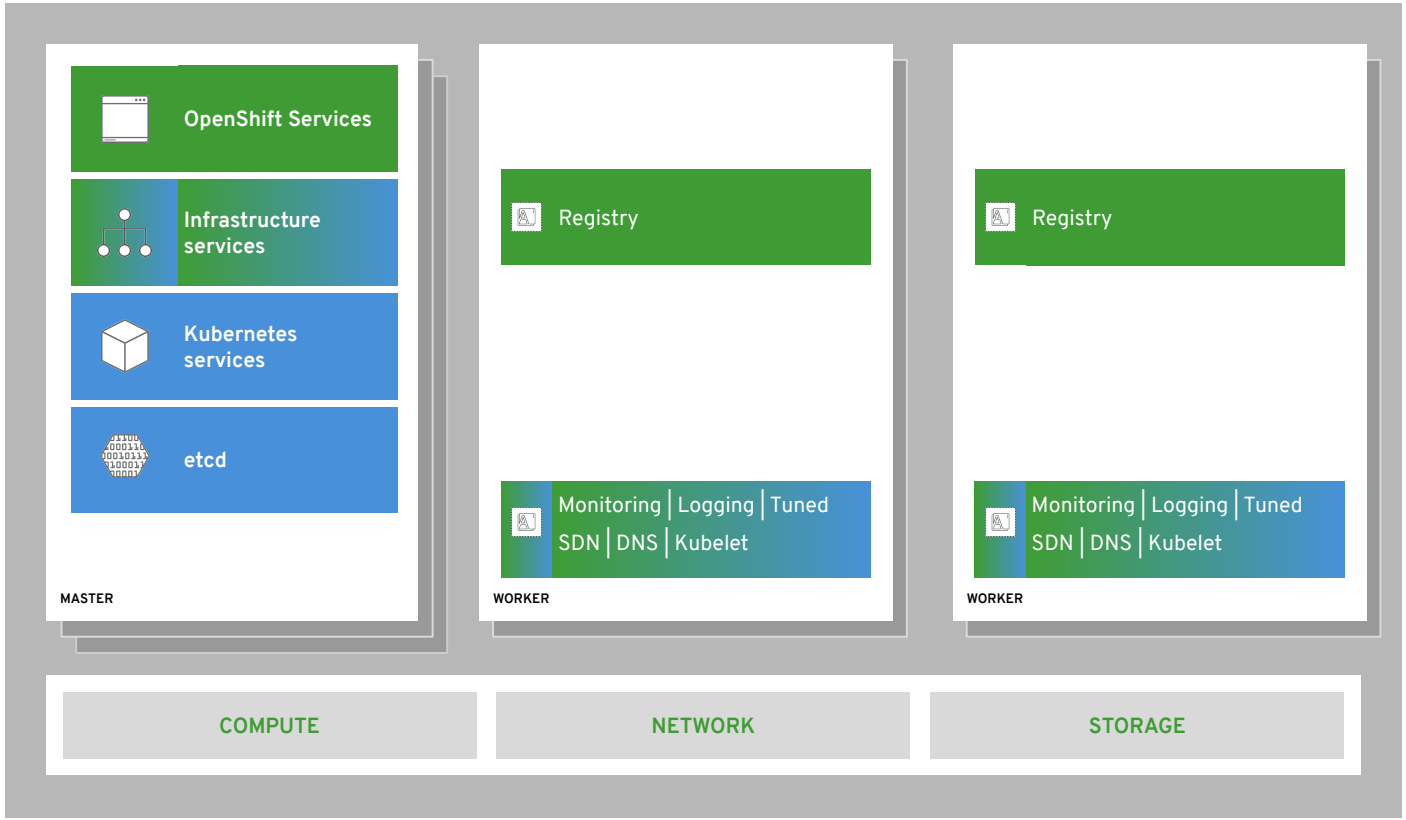
internal and support infrastructure services



Red Hat

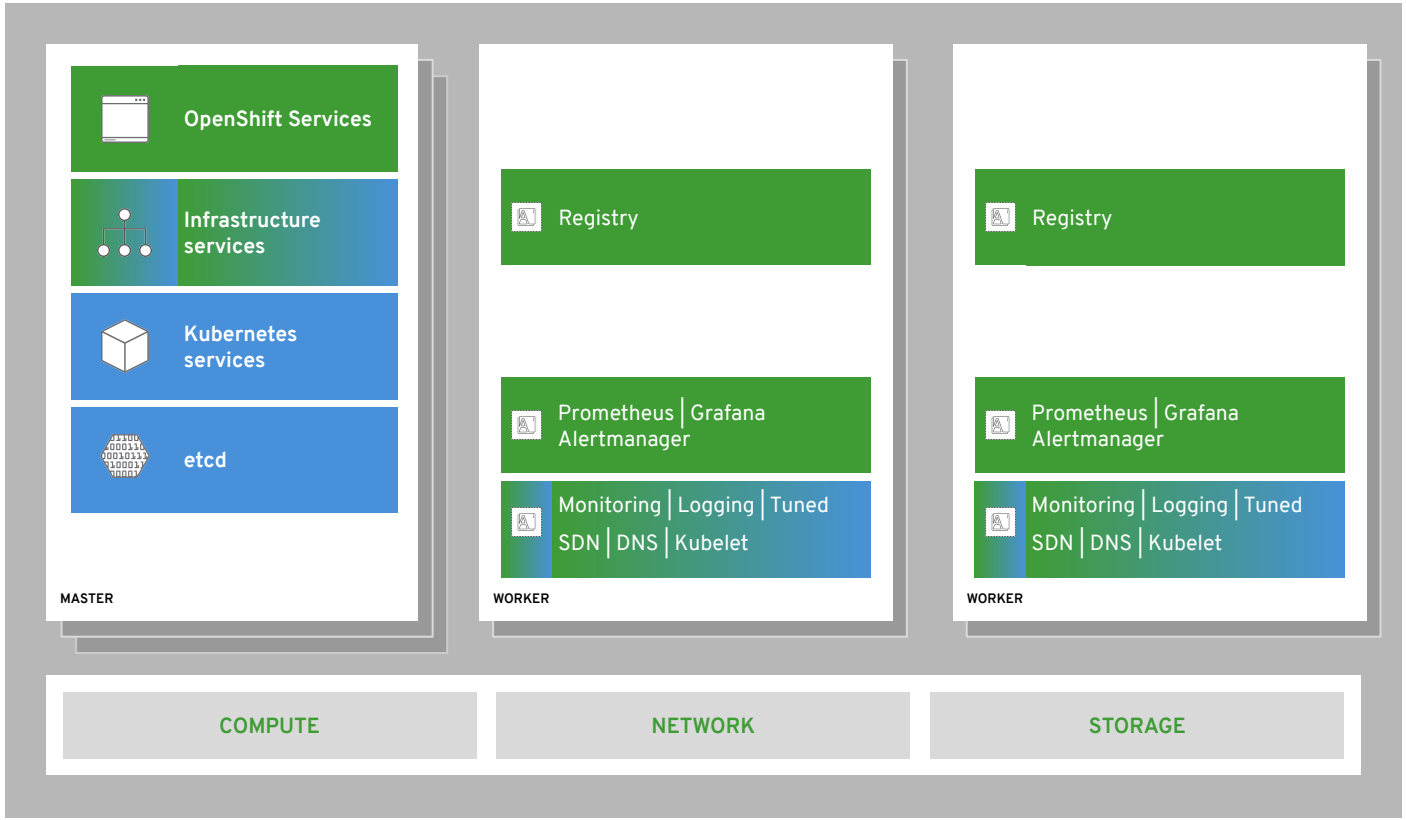
run on all hosts

integrated image registry



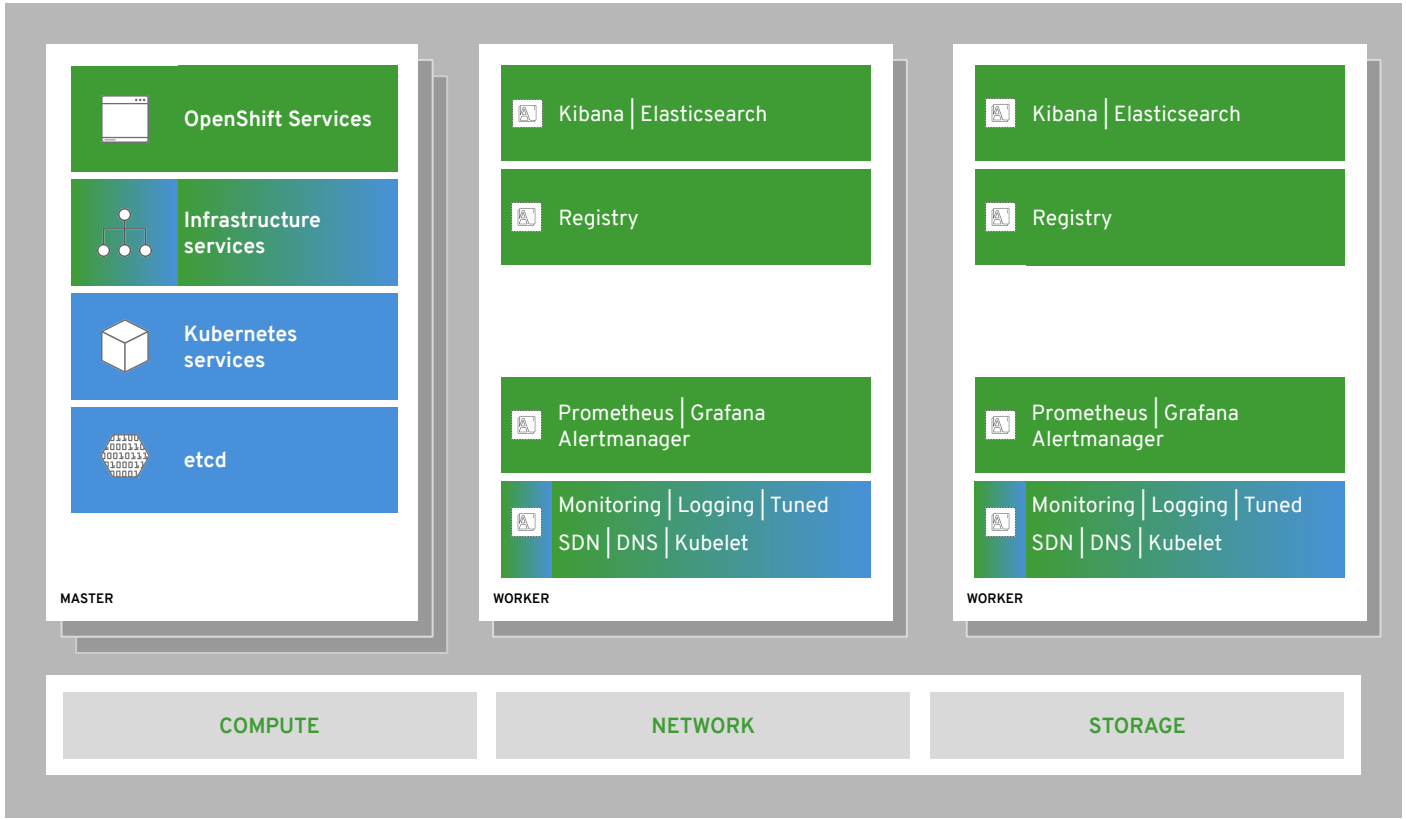
Red Hat

cluster monitoring

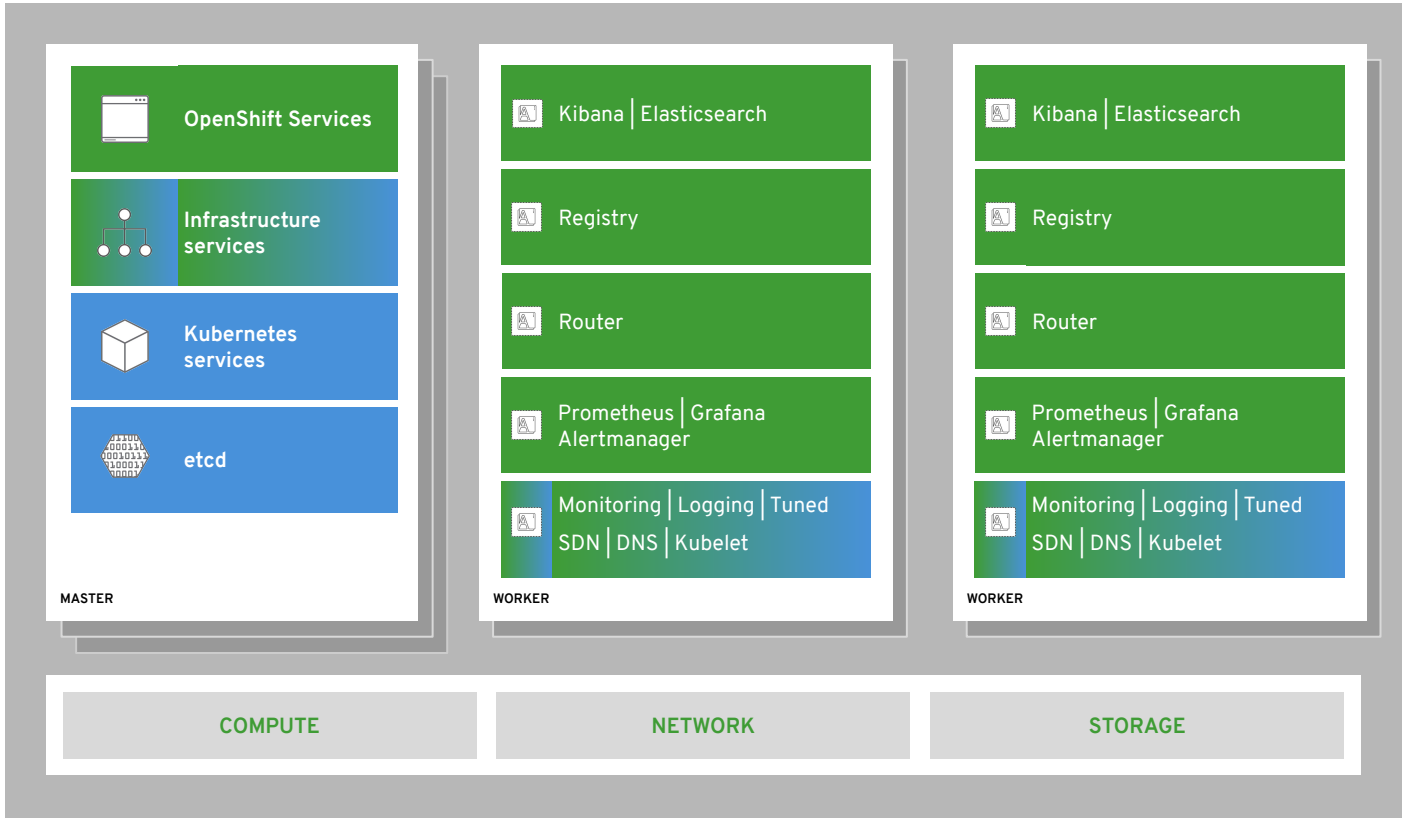


Red Hat

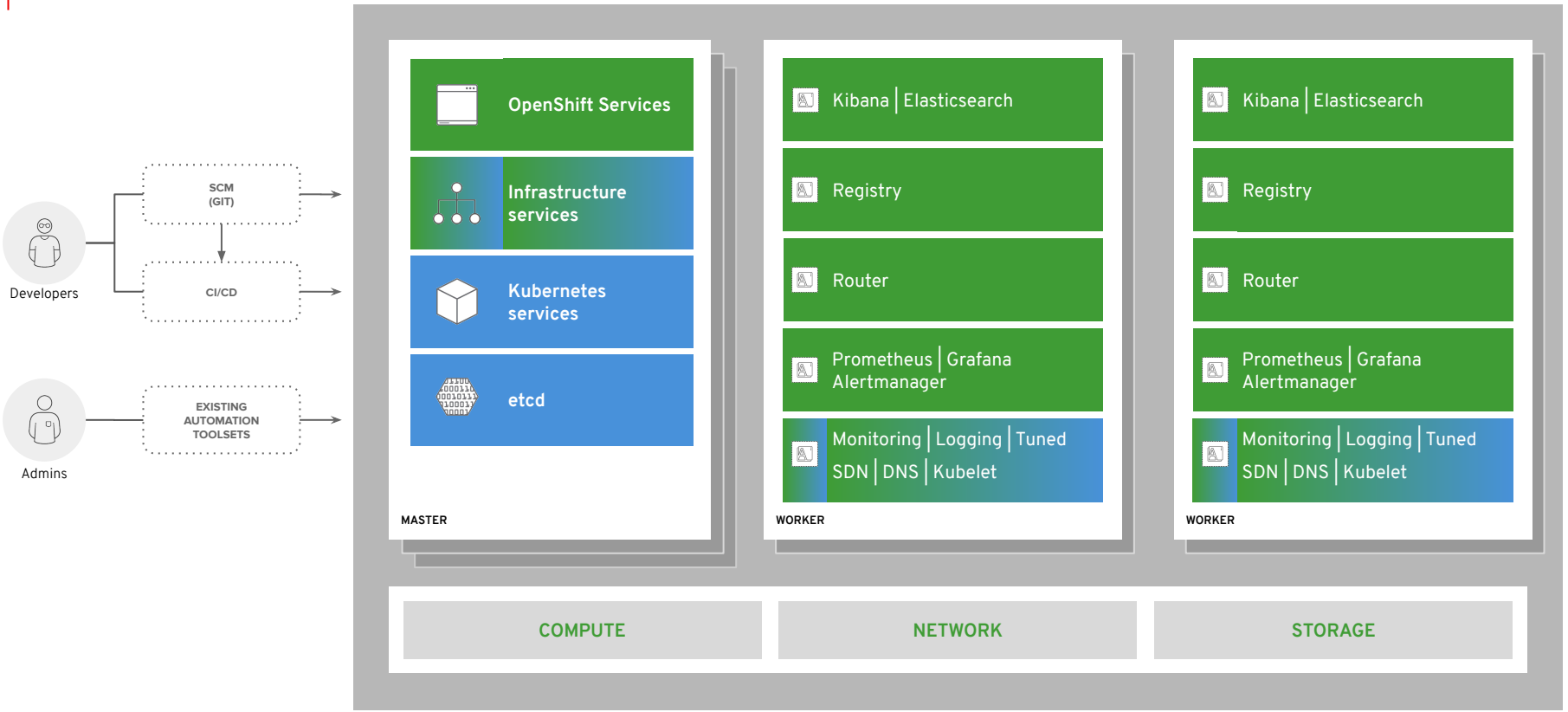
log aggregation



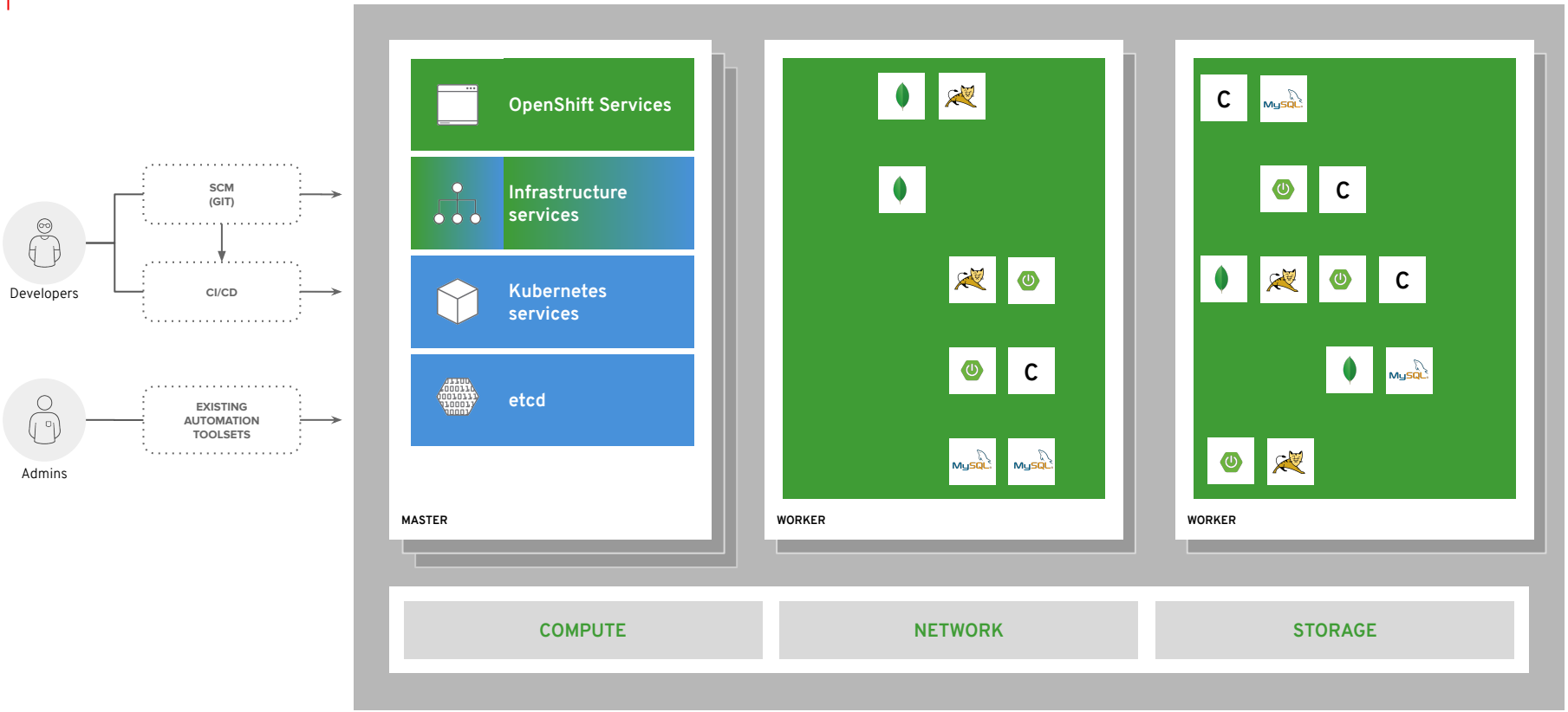
integrated routing



dev and ops via web, cli, API, and IDE

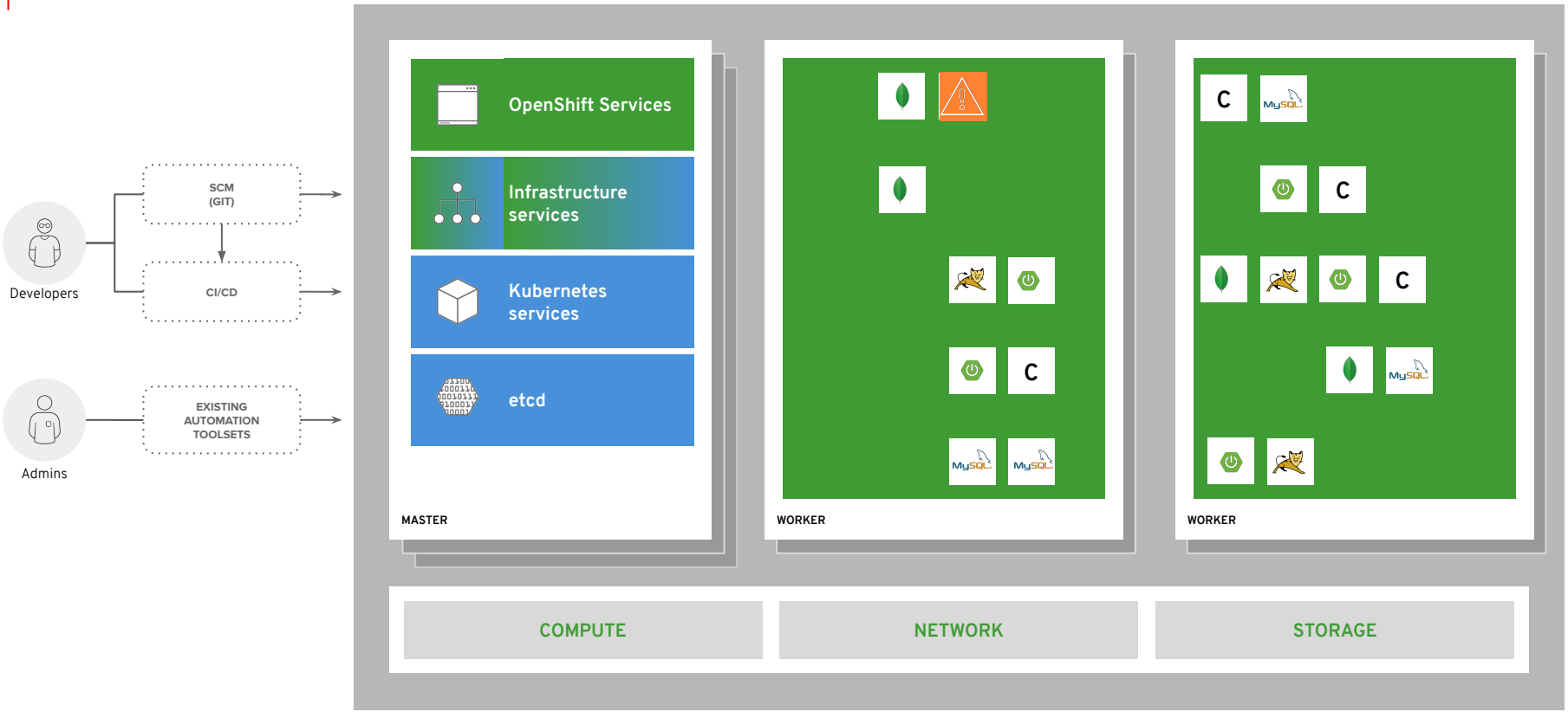


applications also run on workers



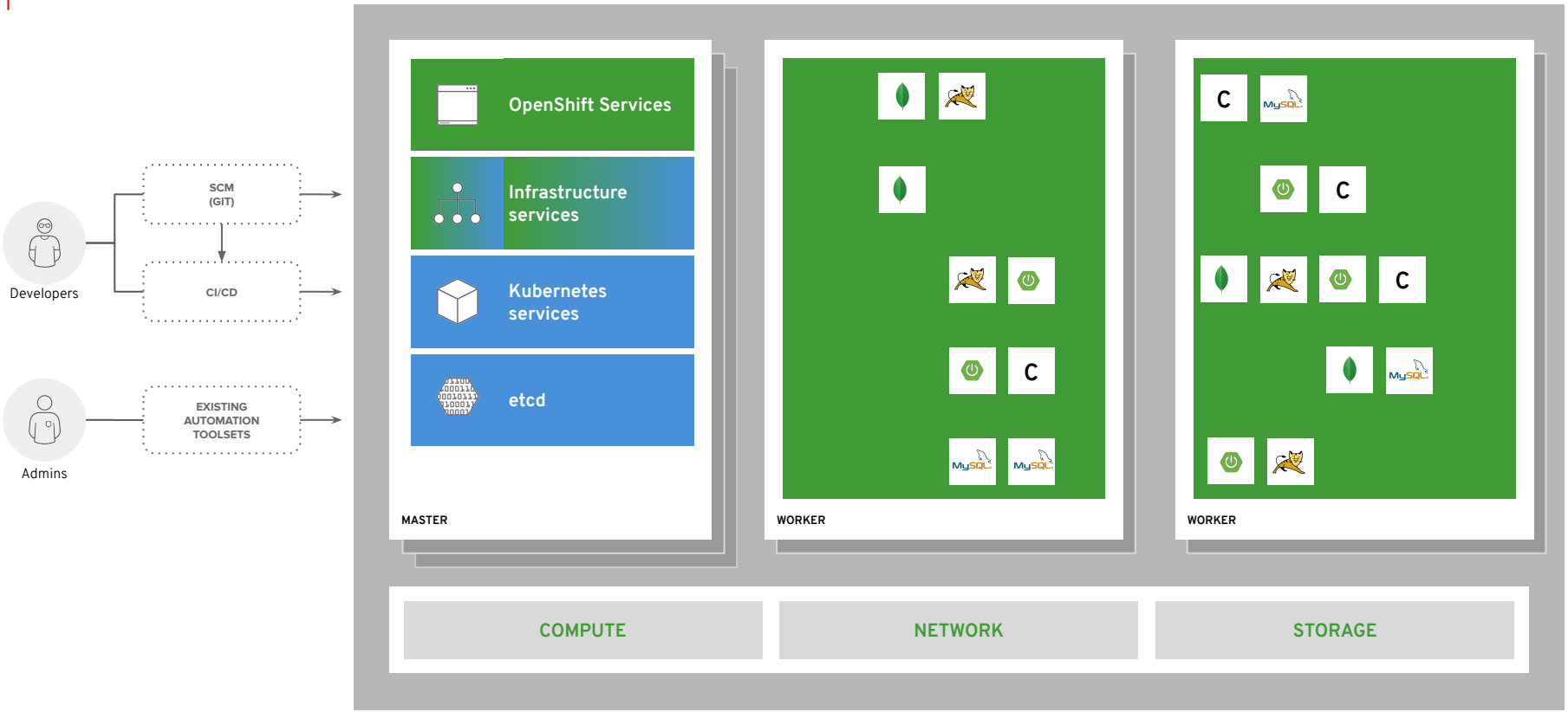
Red Hat

auto-healing failed pods



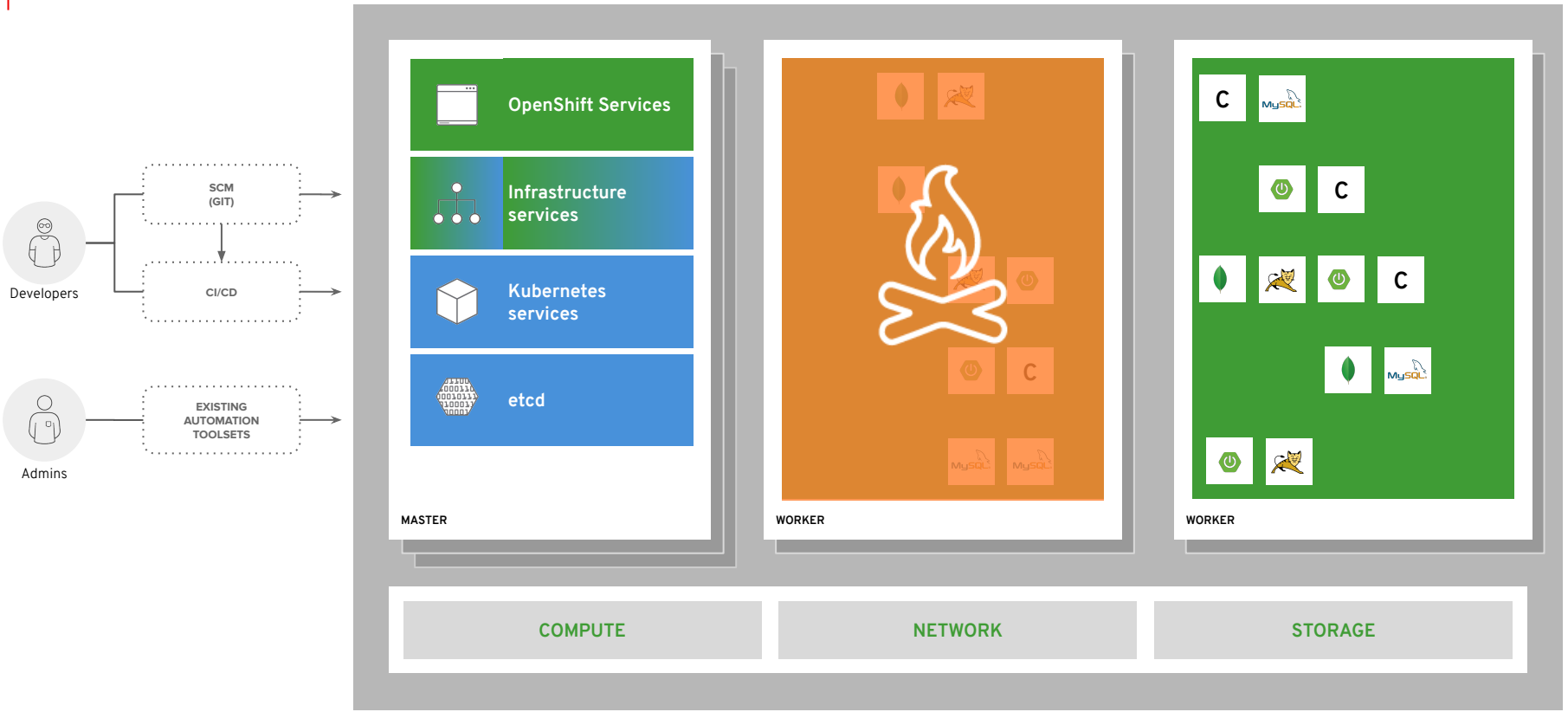
Red Hat

auto-healing failed pods



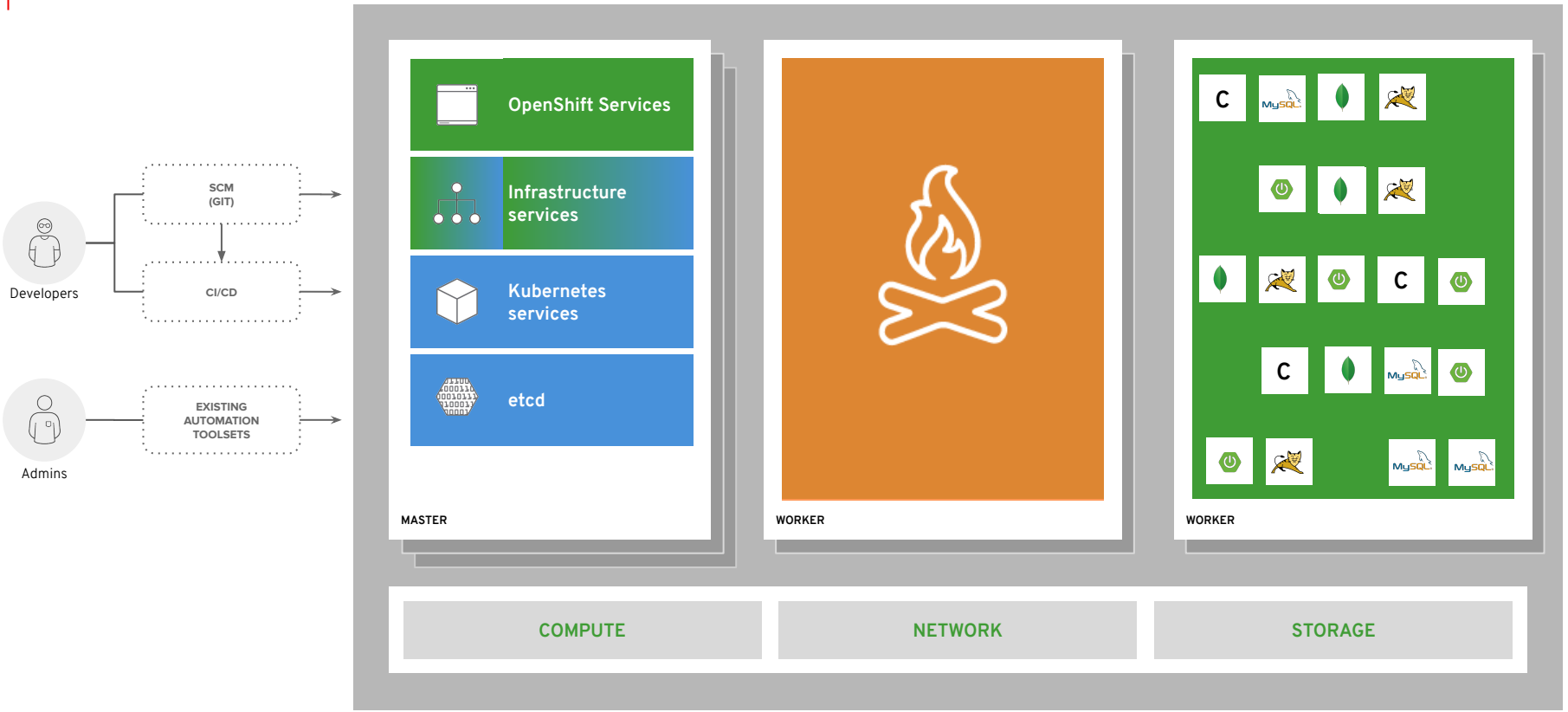
Red Hat

auto-healing for failed **nodes**

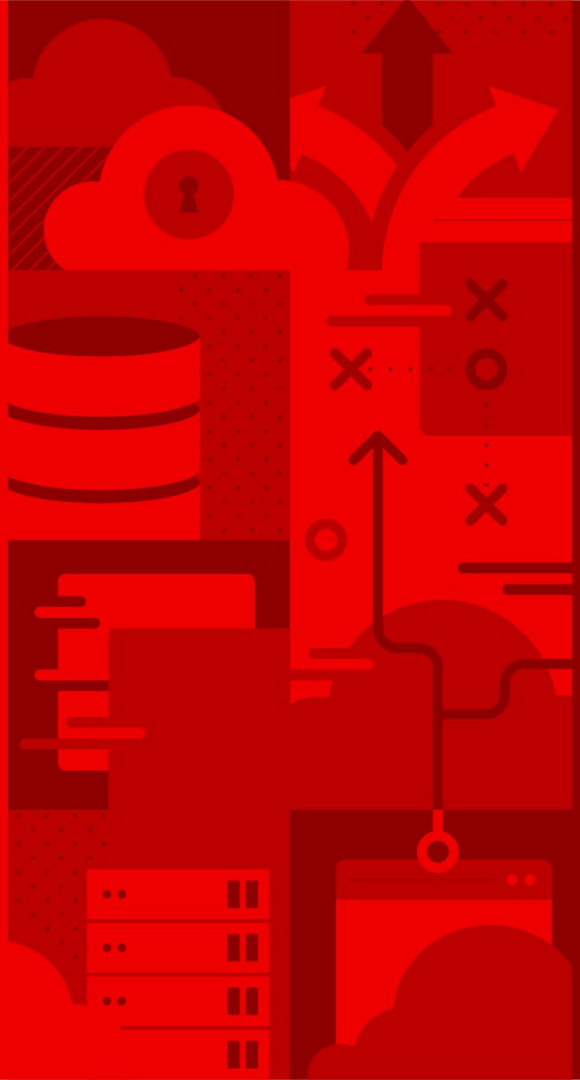


Red Hat

auto-healing for failed **nodes**



Red Hat



Developer Enablement with OpenShift

Red Hat

In order to be successful, developers have to manage...

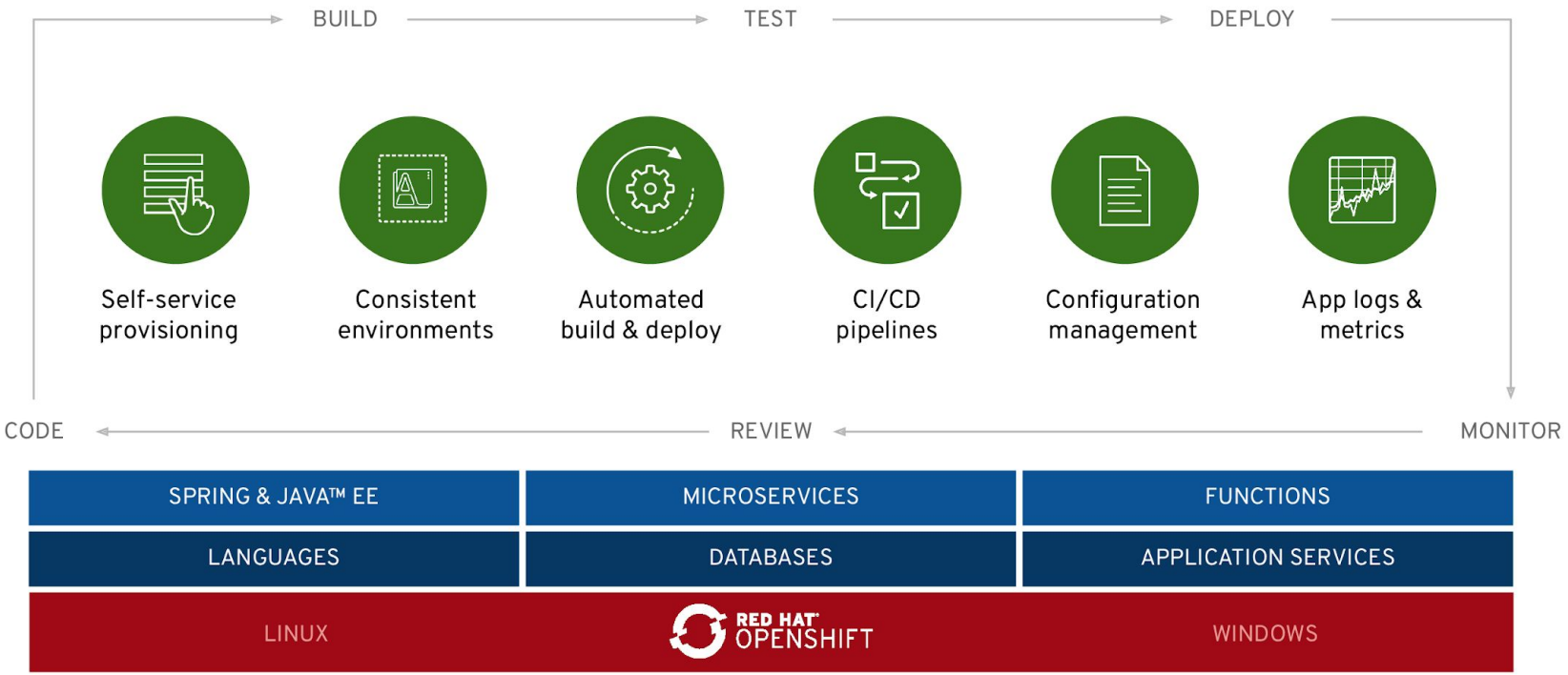
- Changing requirements
- Project source code
- Source code dependencies
- Application dependencies (applications written and maintained by other people)
- Developer tooling: language servers, debuggers, testing tools, security tools, etc...
- Build and packaging tools
- Resources of underlying architecture - “can I have a new VM please?” or “why is the mainframe sad today?”
- Differences in underlying architecture - “it works on my machine!”
- Release processes/windows

Enabling developers means removing the items on that list that are frustrating and that impede (and slow down) their ability to deliver software.

A lot of these issues are process or cultural issues, and they can't be solved by a particular tool, or even a set of tools.

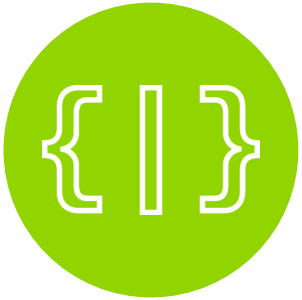
However, OpenShift and its surrounding ecosystem of tools can *help* with a lot of these pain points.

OpenShift enables developer productivity



Red Hat

Build Containers for Cloud-Native Architecture

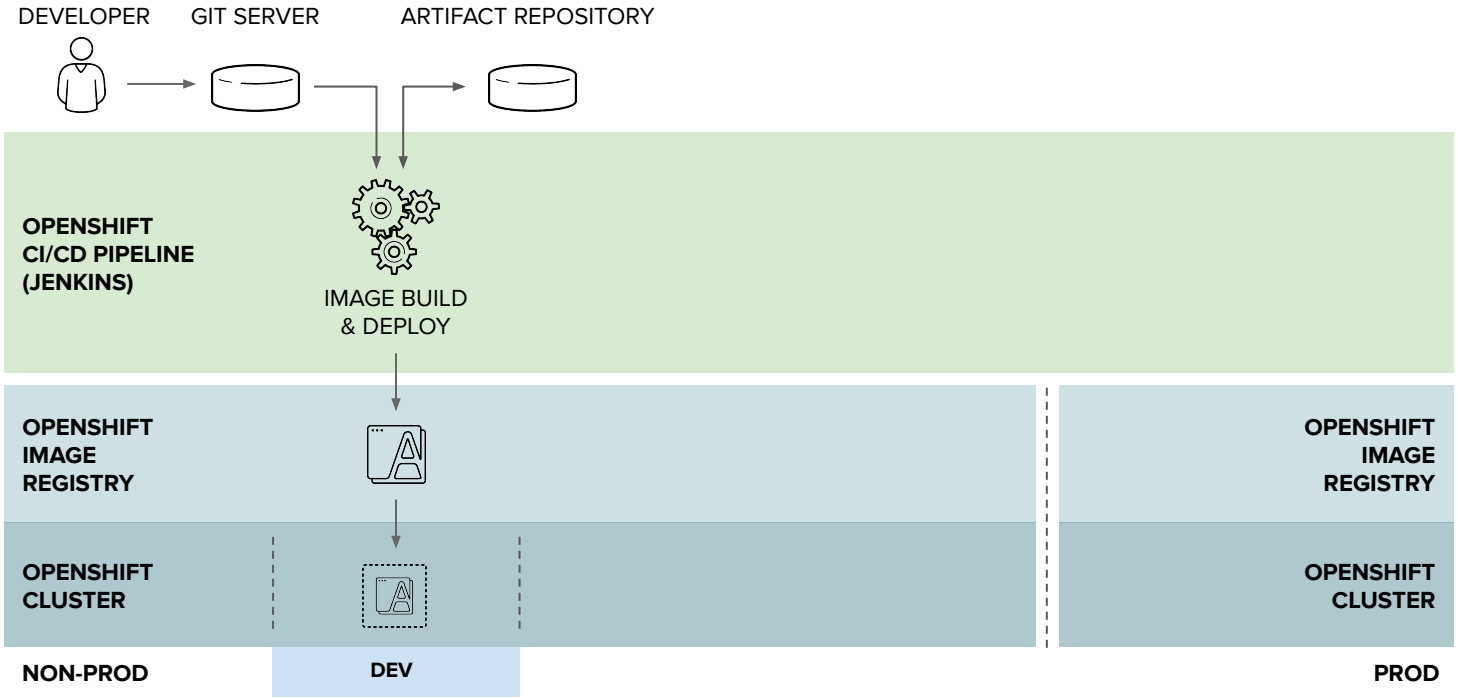


DEPLOY YOUR SOURCE CODE

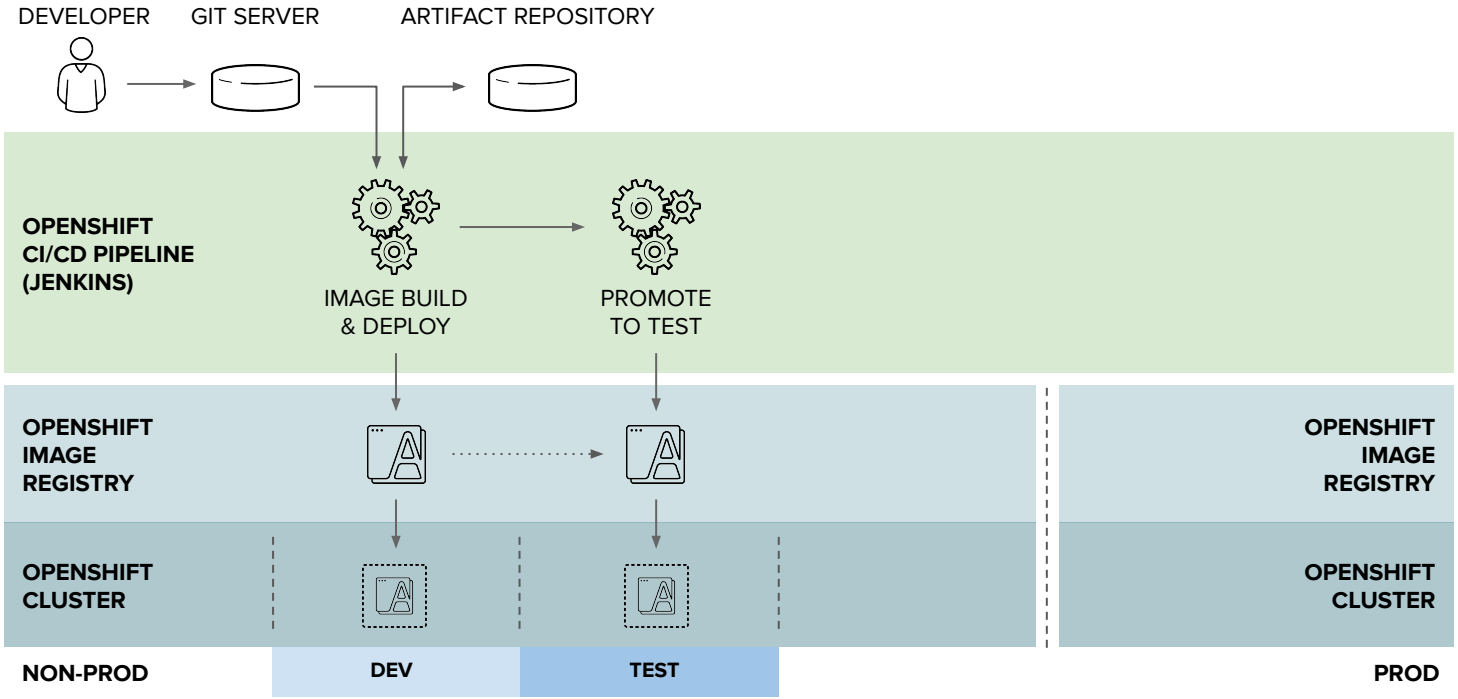
DEPLOY YOUR APP BINARY

DEPLOY YOUR CONTAINER IMAGE

CONTINUOUS DELIVERY PIPELINE

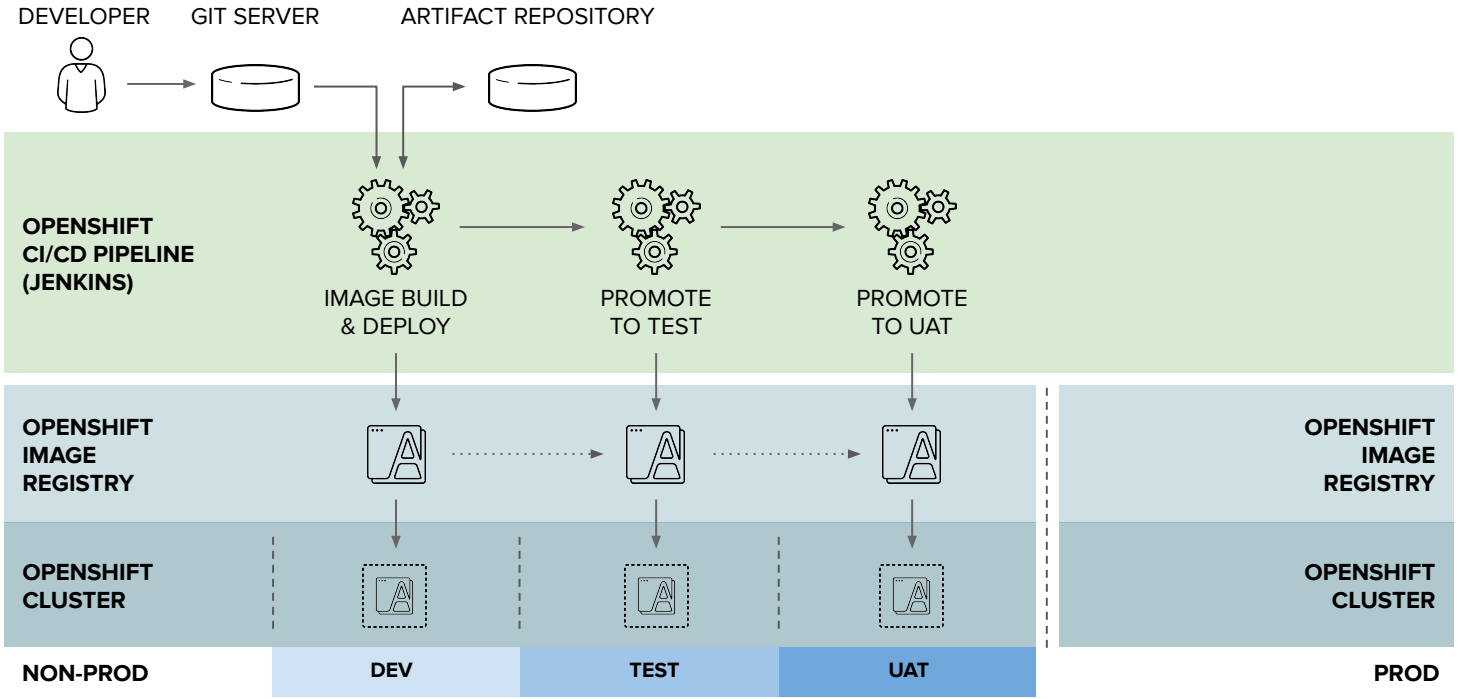


CONTINUOUS DELIVERY PIPELINE

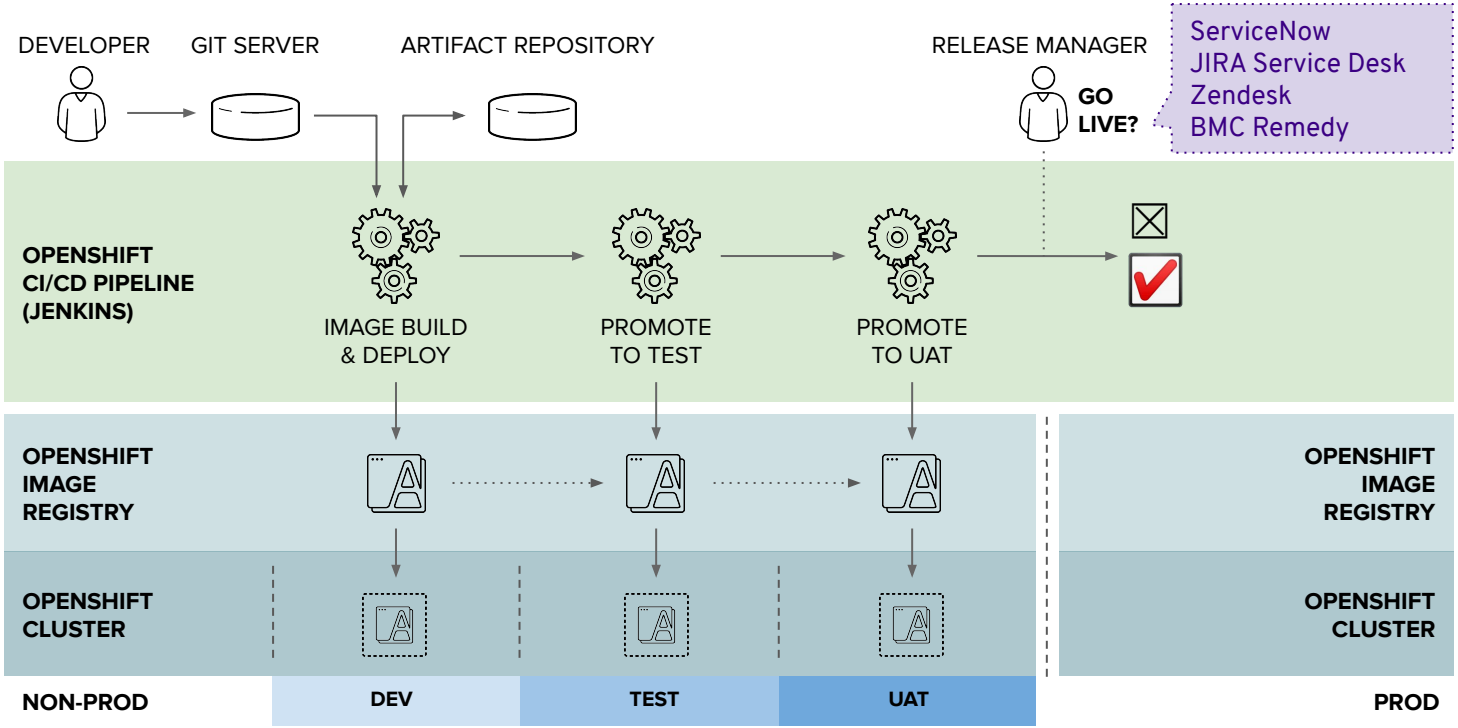


Red Hat

CONTINUOUS DELIVERY PIPELINE

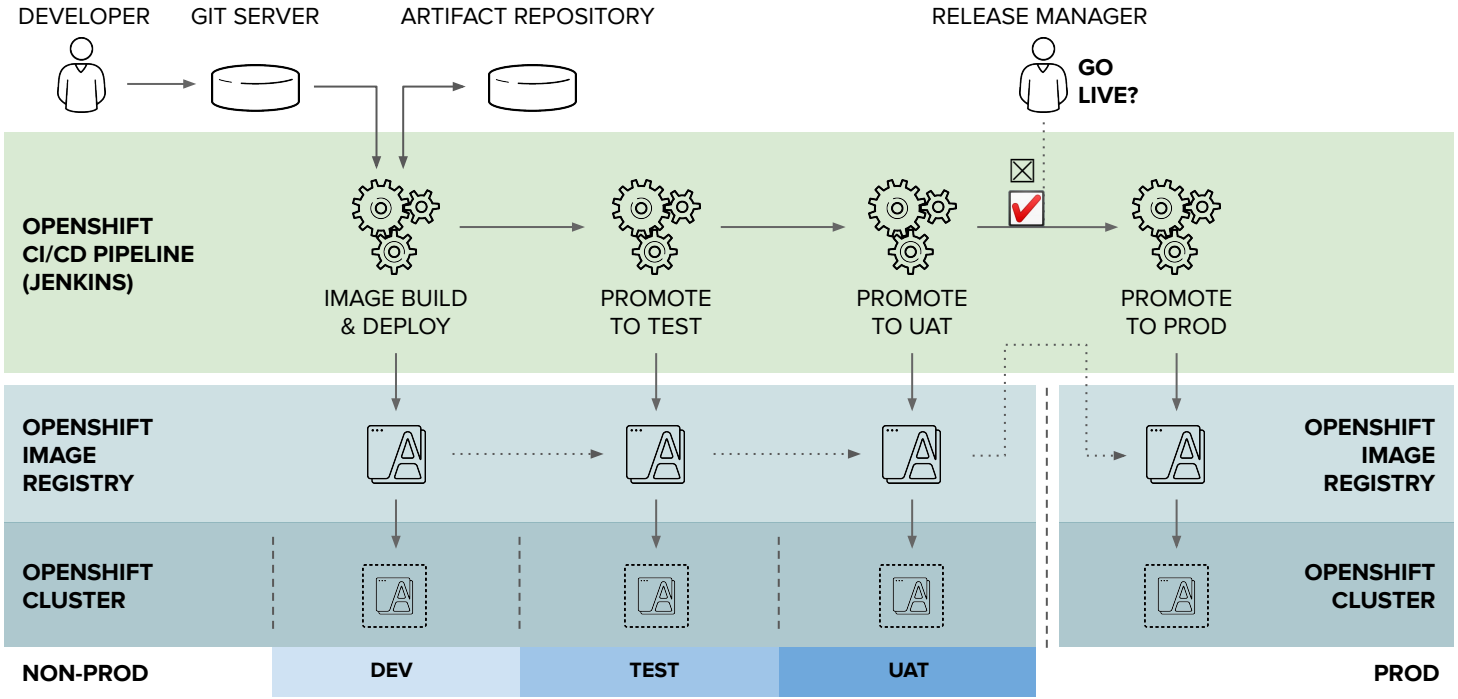


CONTINUOUS DELIVERY PIPELINE



Red Hat

CONTINUOUS DELIVERY PIPELINE

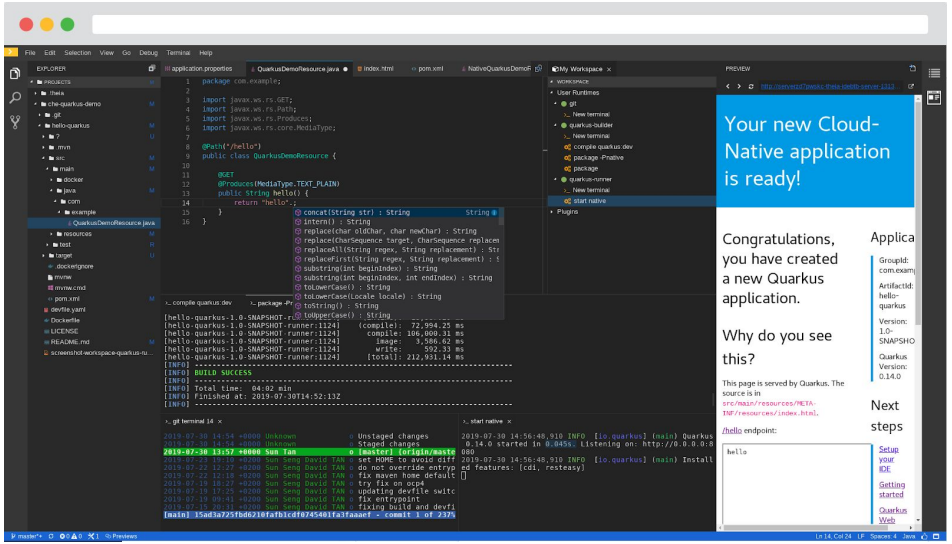


Red Hat

CodeReady Workspaces 2.0

Based on Eclipse Che 7

- **Kubernetes-based developer workspaces:** Fully containerized developer workspaces allowing to bring your K8S application runtime easily in your dev environment.
- **New Editor**: New default web-based editor provides a VSCode like experience in the browser.
- **Devfile:** Configure a devfile for your project and get reproducible and portable developer environments.
- **VSCode plug-ins compatibility**
- **Swappable Editor**
- **OpenShift VSCode Plug-in**
- **Easier to Monitor and Administrate:** Prometheus and Grafana dashboards.

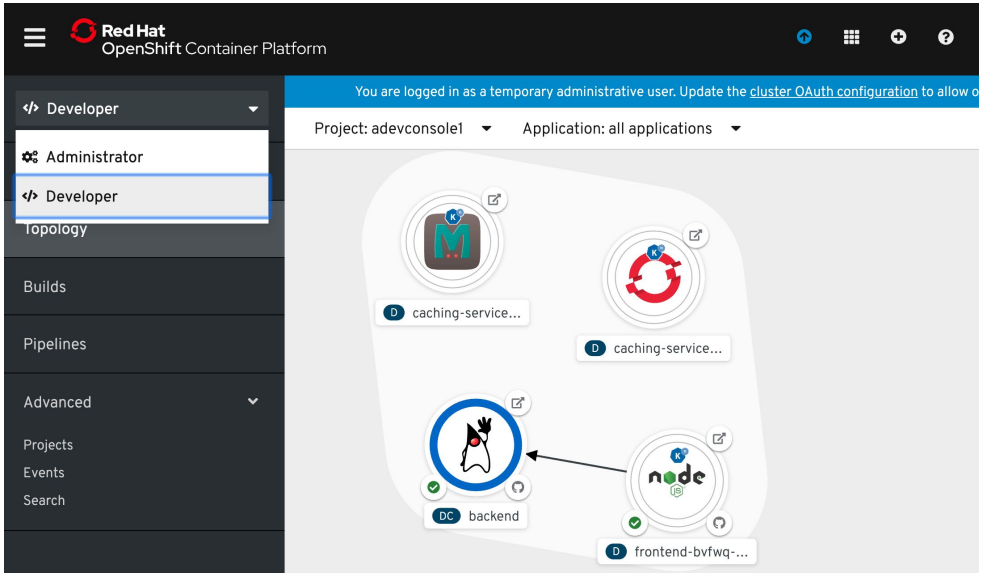


Red Hat

Application Visibility with the Web Console - Developer Perspective

An alternative perspective in the OpenShift UI that will sit beside the admin console and focus on developer use cases.

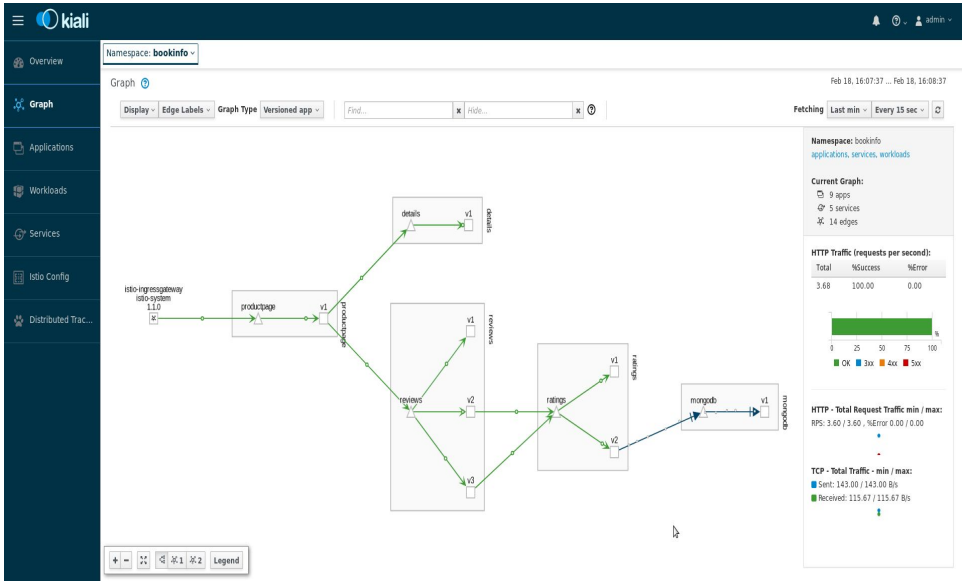
All OpenShift developer tool UIs will be surfaced here...though some (like CodeReady Workspaces) will be links out to unique UIs.



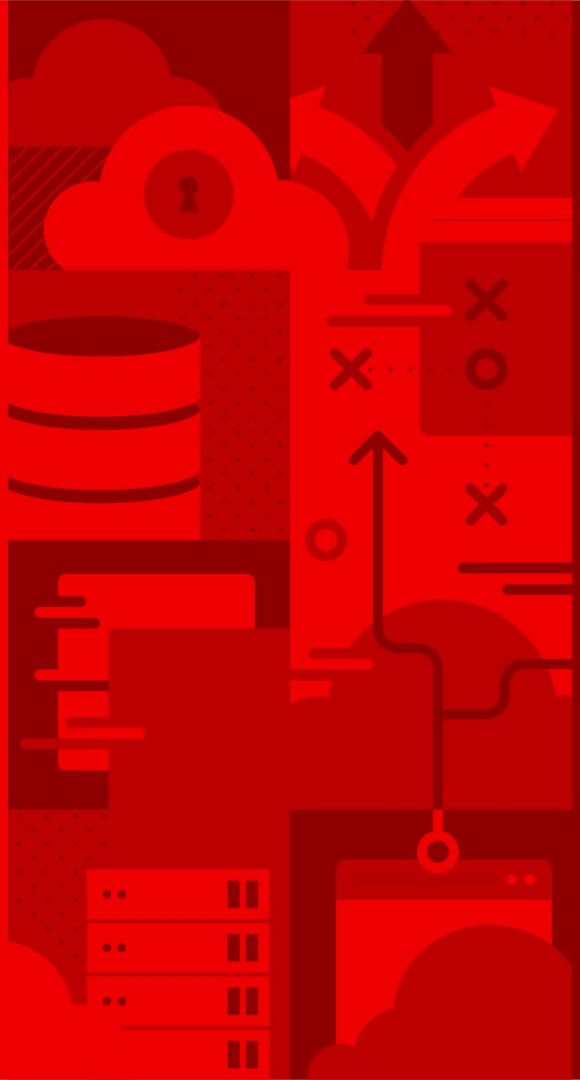
Red Hat Service Mesh

Key Features

- A dedicated network for service to service communications
- Observability and distributed tracing
- Policy-driven security
- Routing rules & chaos engineering
- Powerful visualization & monitoring
- Will be available via OperatorHub



Red Hat



Demo!

Red Hat