



ENTERPRISE KUBERNETES OPENSIFT


An introduction to the Container Platform





For NOVALUG, April 2021
By Peter Larsen


OPENSSHIFT CONTAINER PLATFORM

TECHNICAL OVERVIEW

 [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

Peter Larsen
Staff Domain Architect, NAPS

plarsen@redhat.com

peter@peterlarsen.org



Recapture - containers

- Linux Containers is a effective way to distribute and run software
- Containers run in isolation allowing for “conflicting” software to run on the same host
- To build containers, podman and buildah are used
- To run a container, “podman run” executes locally
- A container “is just” Linux processes
It runs as fast with access to the same hardware as normal processes
- Containers are embedded in Linux security: SELinux, CGroups and Namespaces

Great - so why a container platform?

- A single host has limits - memory, CPU, IO and more
- Full HA applications must not have single point of failures
- Managing individual containers is complex when you have thousands
- Software needs updating, failure, monitoring
- Access to features like Storage, network needs to be part of the solution

Solution: AUTOMATION

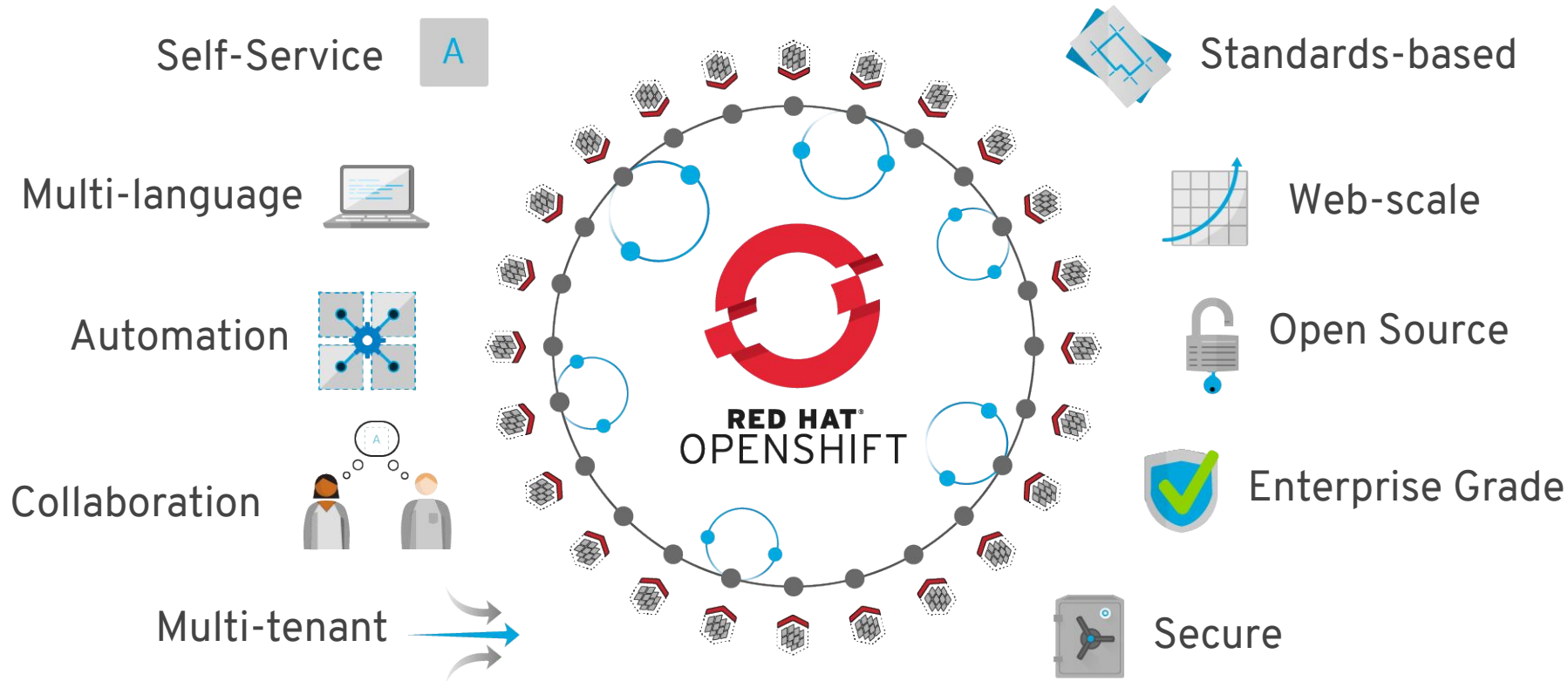
A container platform automates running containers - allows for much better integration into CI/CD pipelines

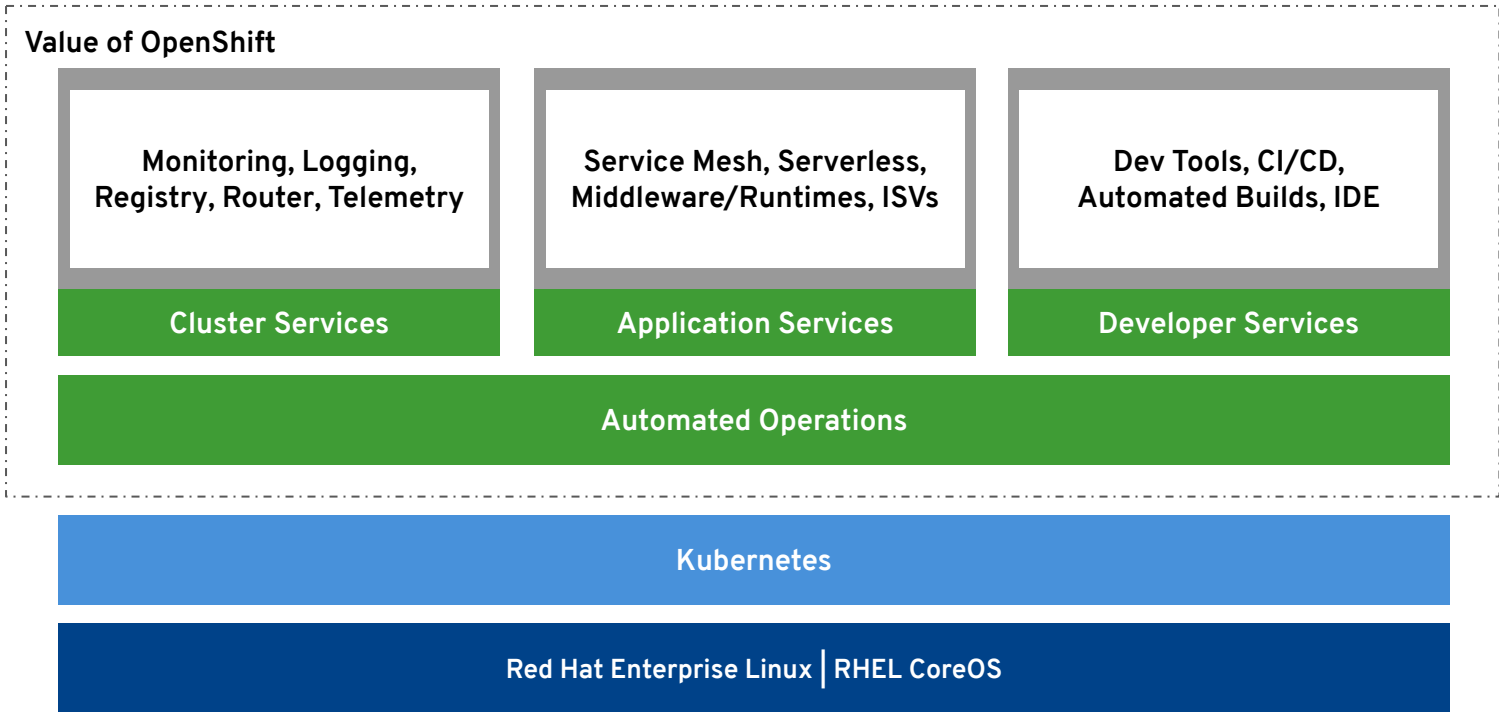
But this is enterprise - so it's not open source?

- FALSE!
- Upstream to Openshift is OKD: <https://www.okd.io/>
 - \$ openshift-install create cluster
- OpenShift / Kubernetes does require a lot of resources
 - But limited development single-node installations are available
 - We'll be focused on full installations in this talk
- The enterprise OpenShift is free for everyone to try for 60 days:
 - <https://cloud.redhat.com>



Functional overview

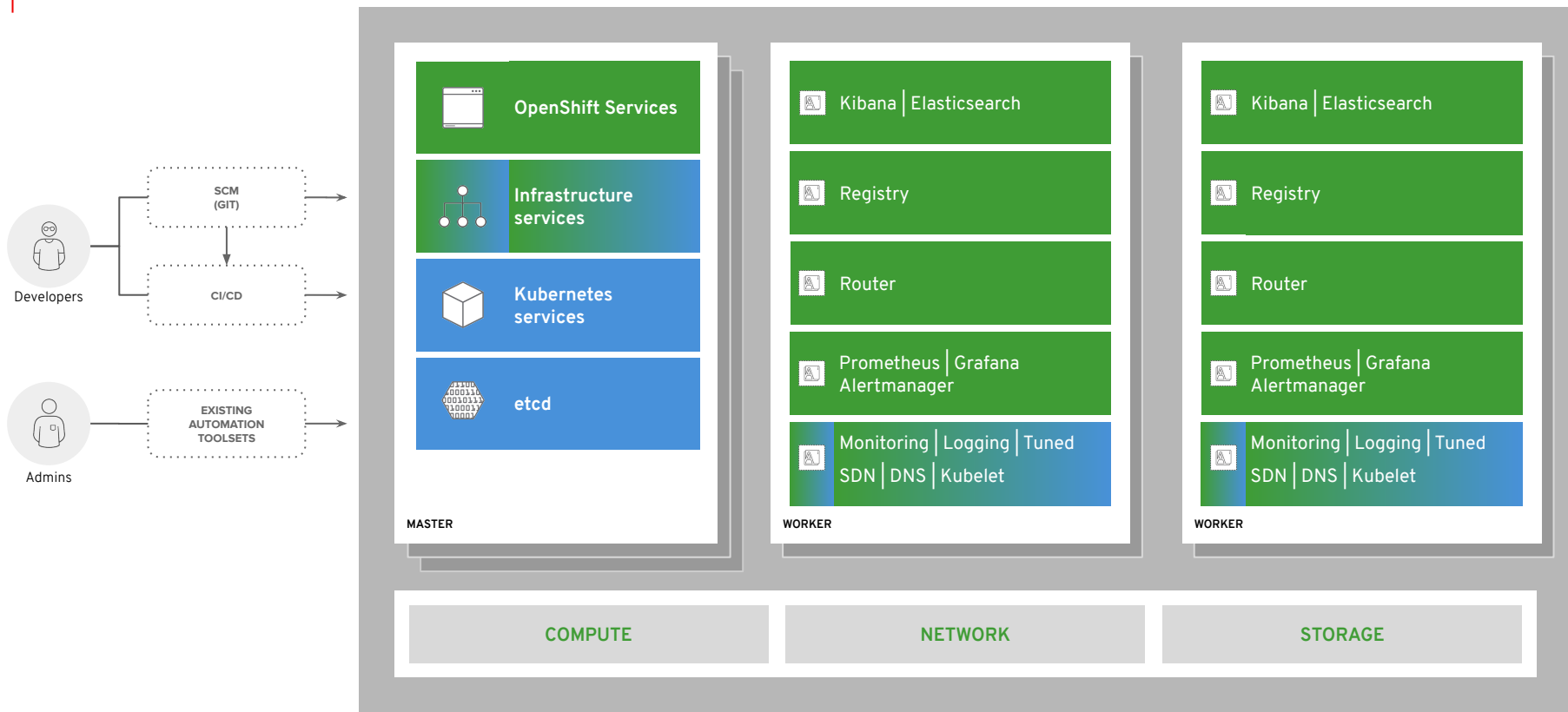


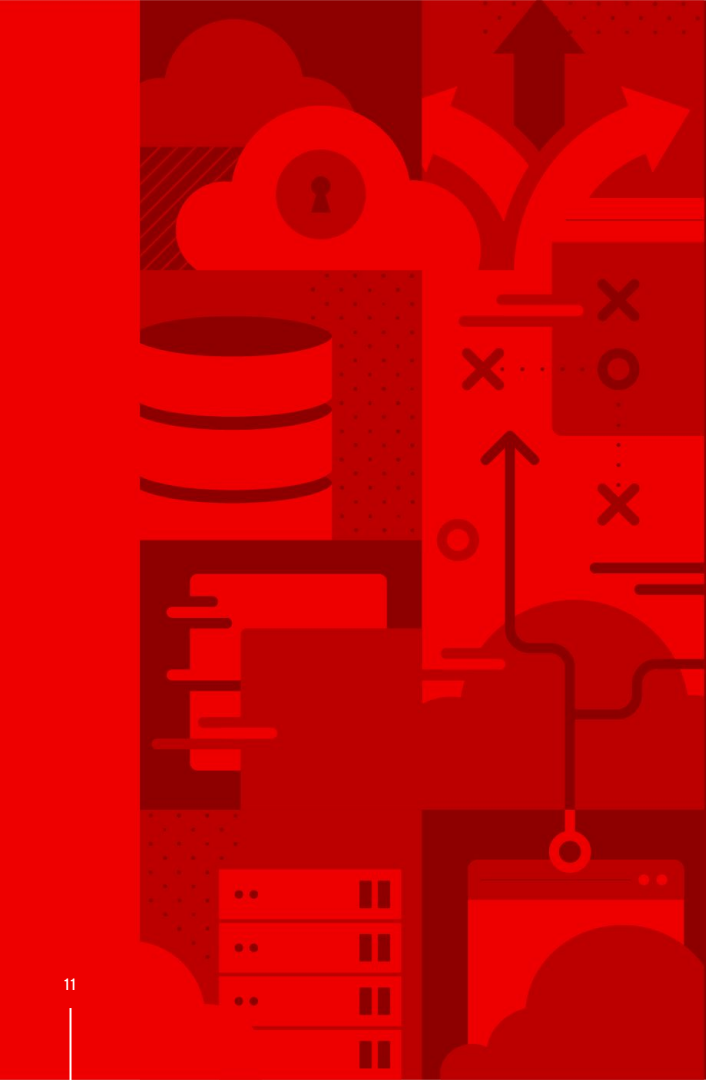


Best IT Ops Experience

CaaS ↔ PaaS ↔ FaaS

Best Developer Experience





OpenShift and Kubernetes core concepts

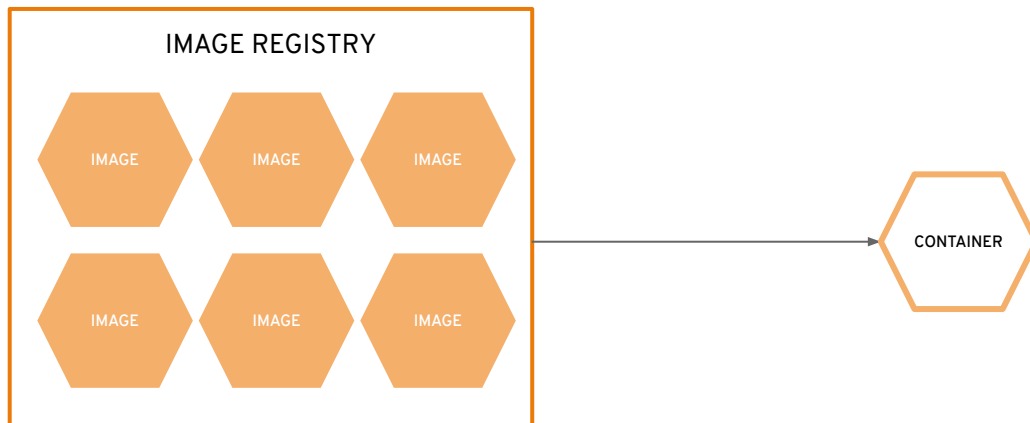
a container is the smallest compute unit



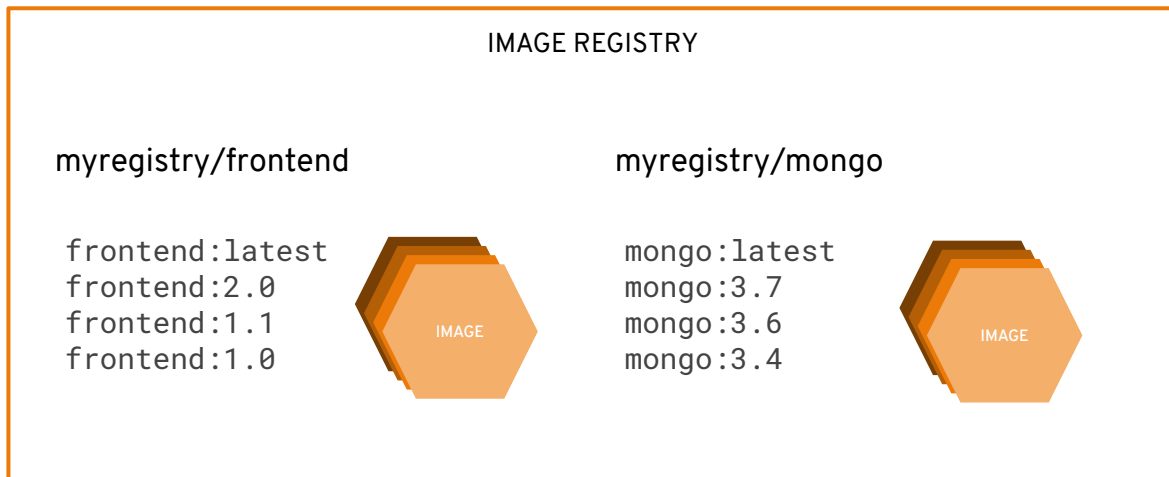
containers are created from container images



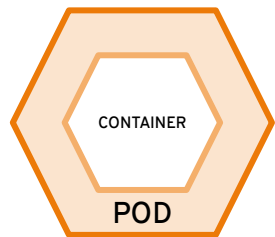
container images are stored in an image registry



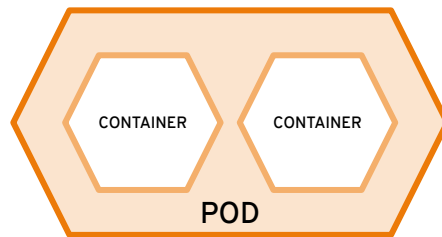
an image repository contains all versions of an image in the image registry



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

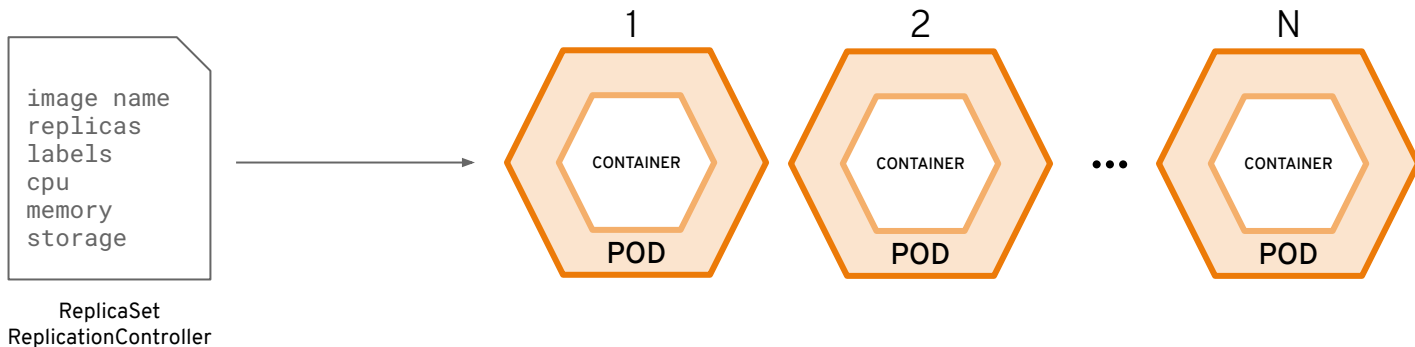


10.140.4.44

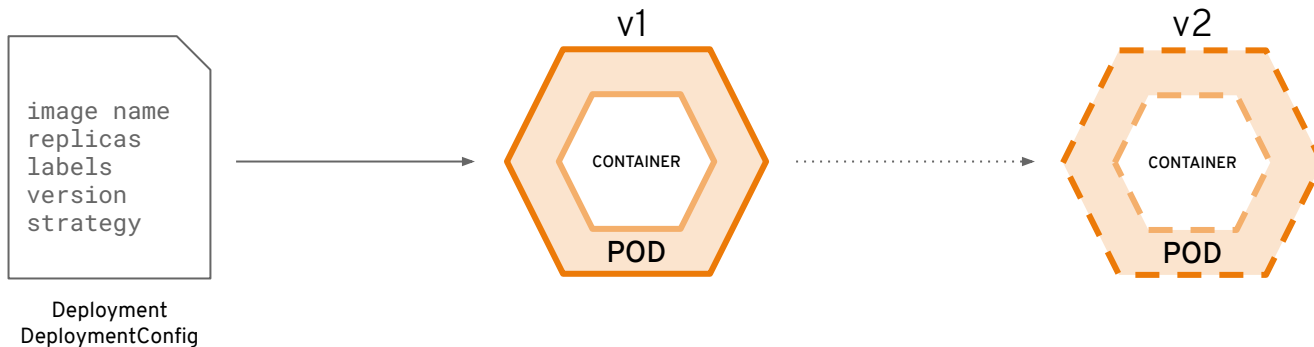


10.15.6.55

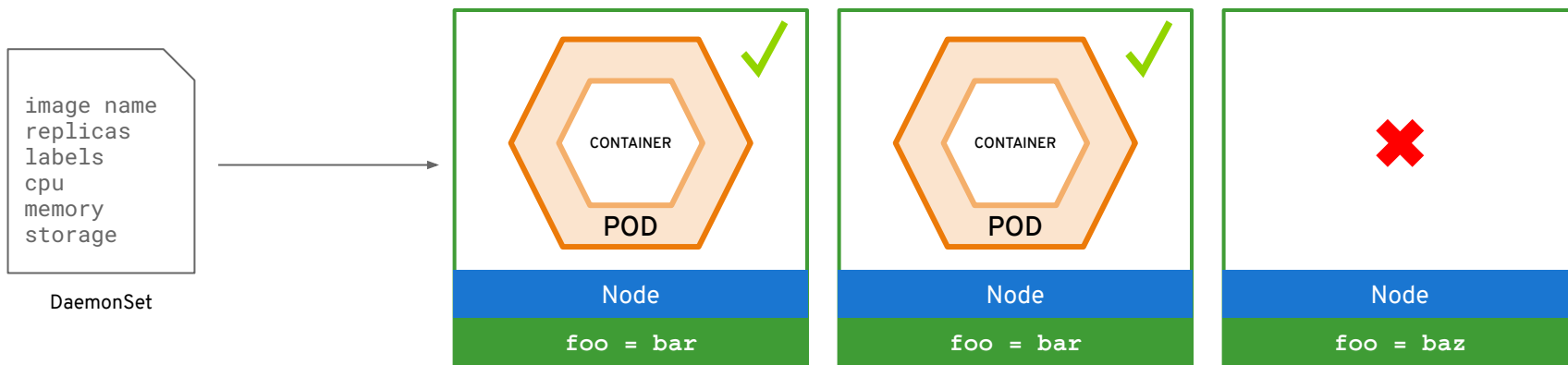
ReplicationControllers & ReplicaSets ensure a specified number of pods are running at any given time



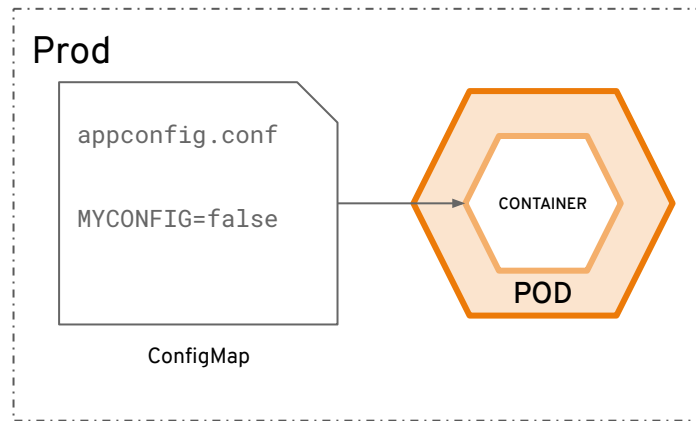
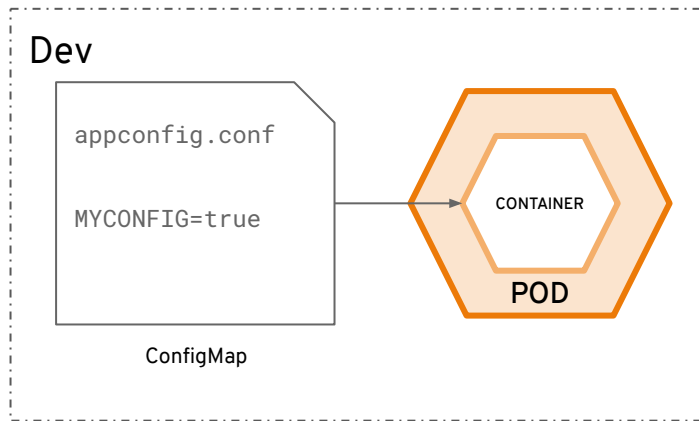
Deployments and DeploymentConfigurations define how to roll out new versions of Pods



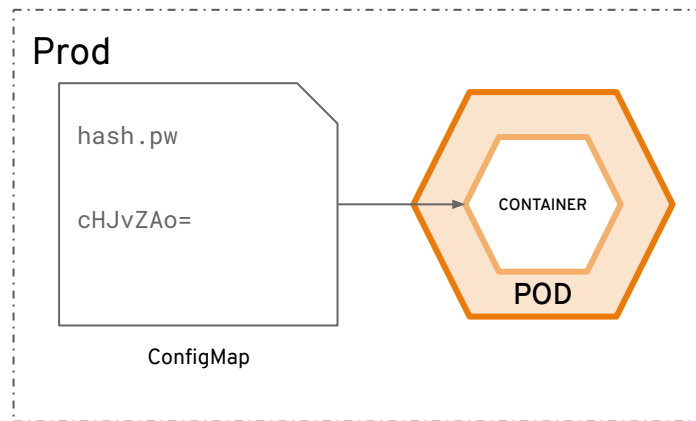
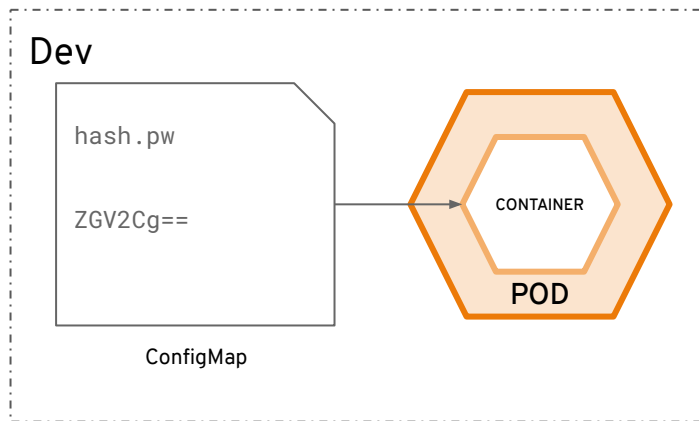
a daemonset ensures that all
(or some) nodes run a copy of a
pod



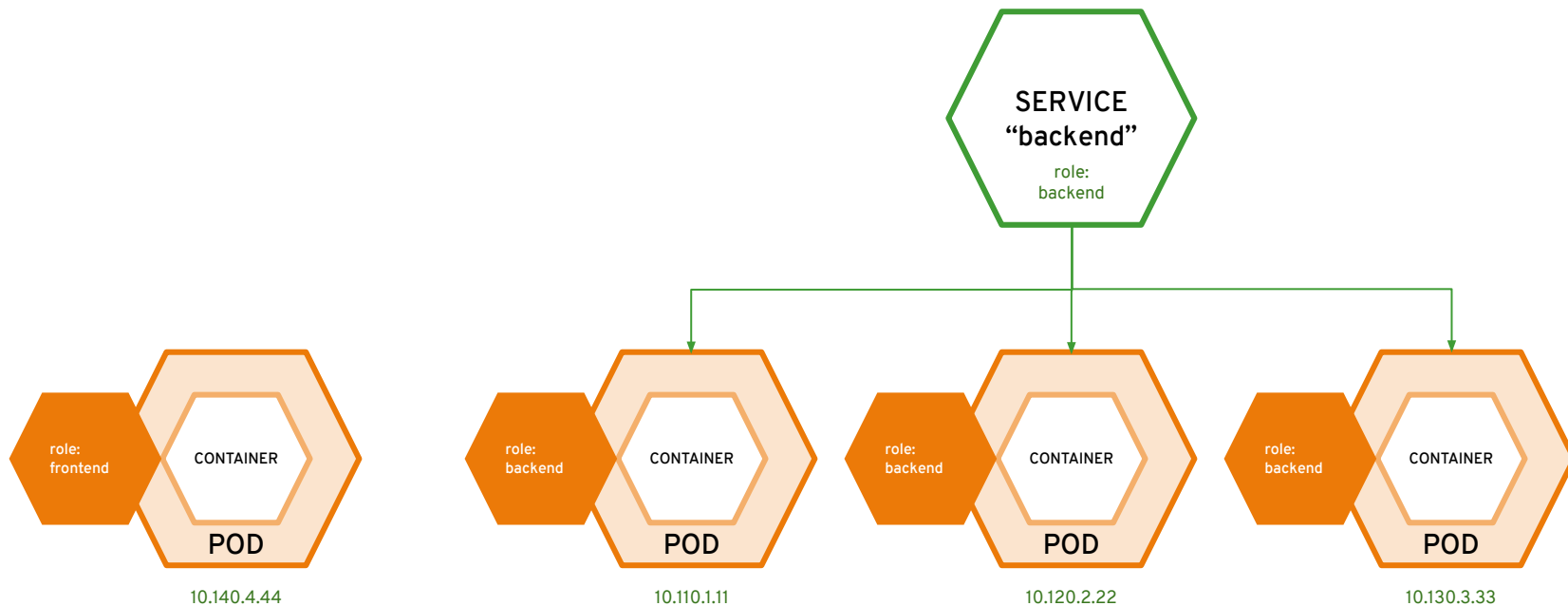
configmaps allow you to decouple configuration artifacts from image content



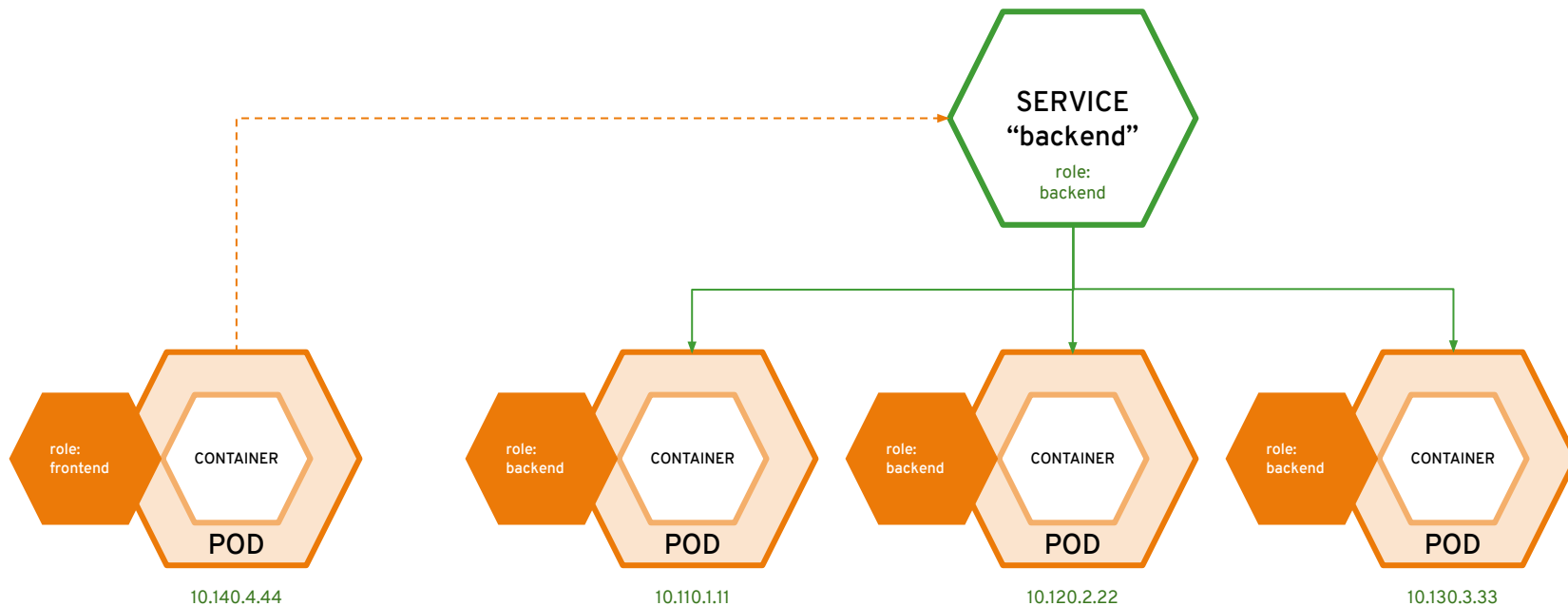
secrets provide a mechanism to hold sensitive information such as passwords



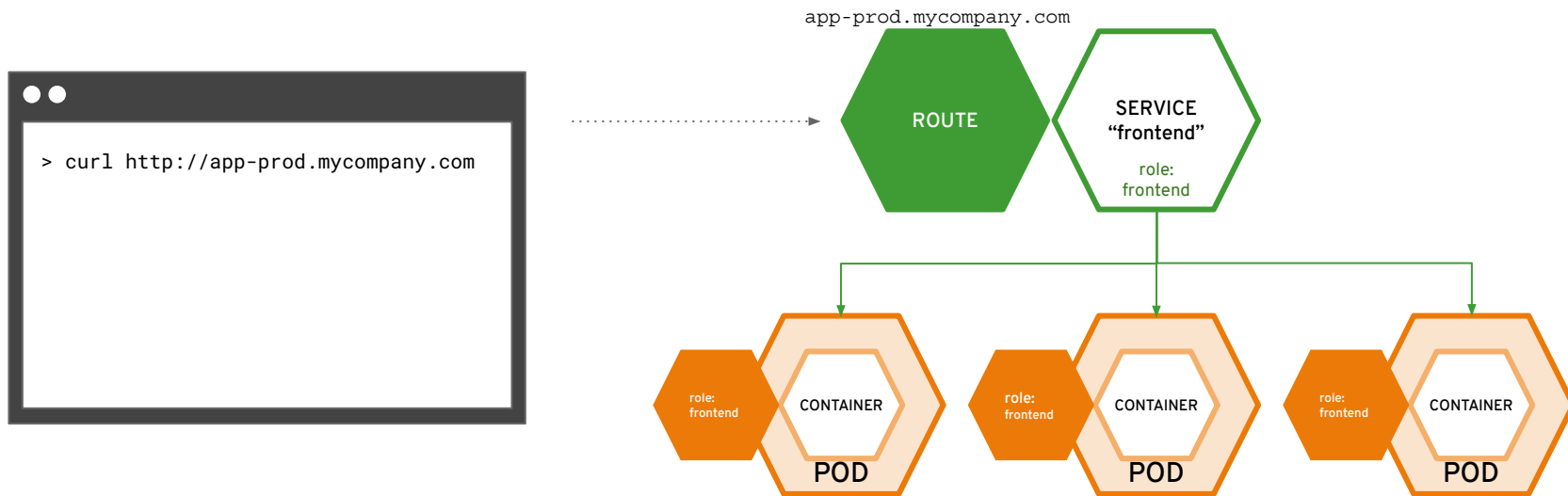
services provide internal load-balancing and service discovery across pods



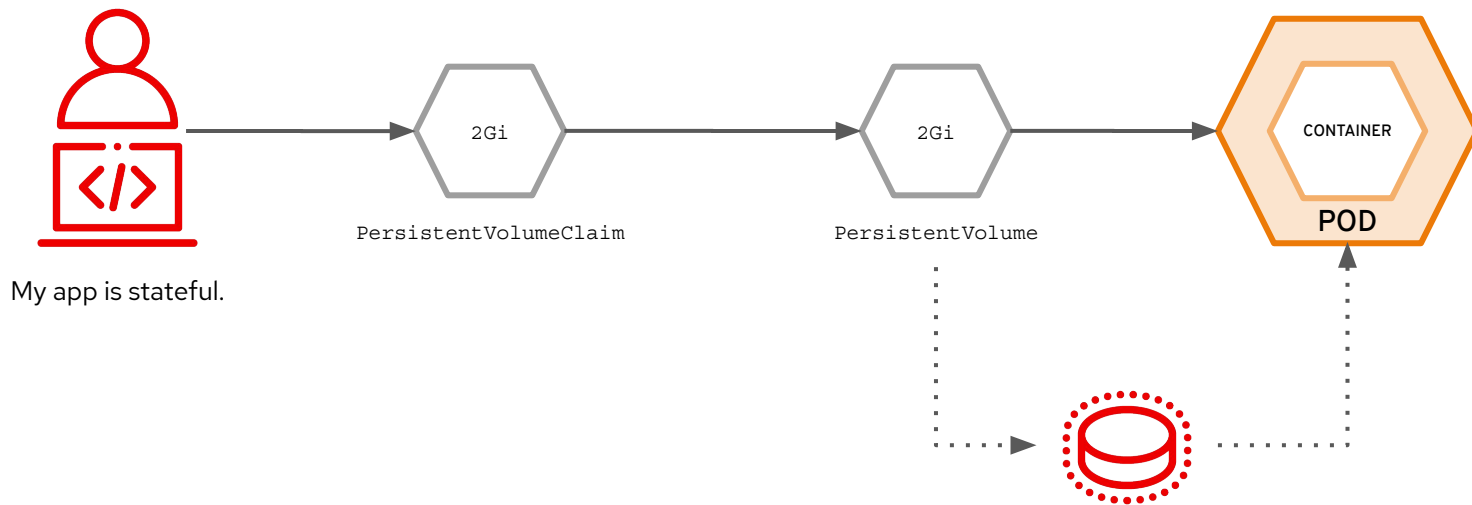
apps can talk to each other via services



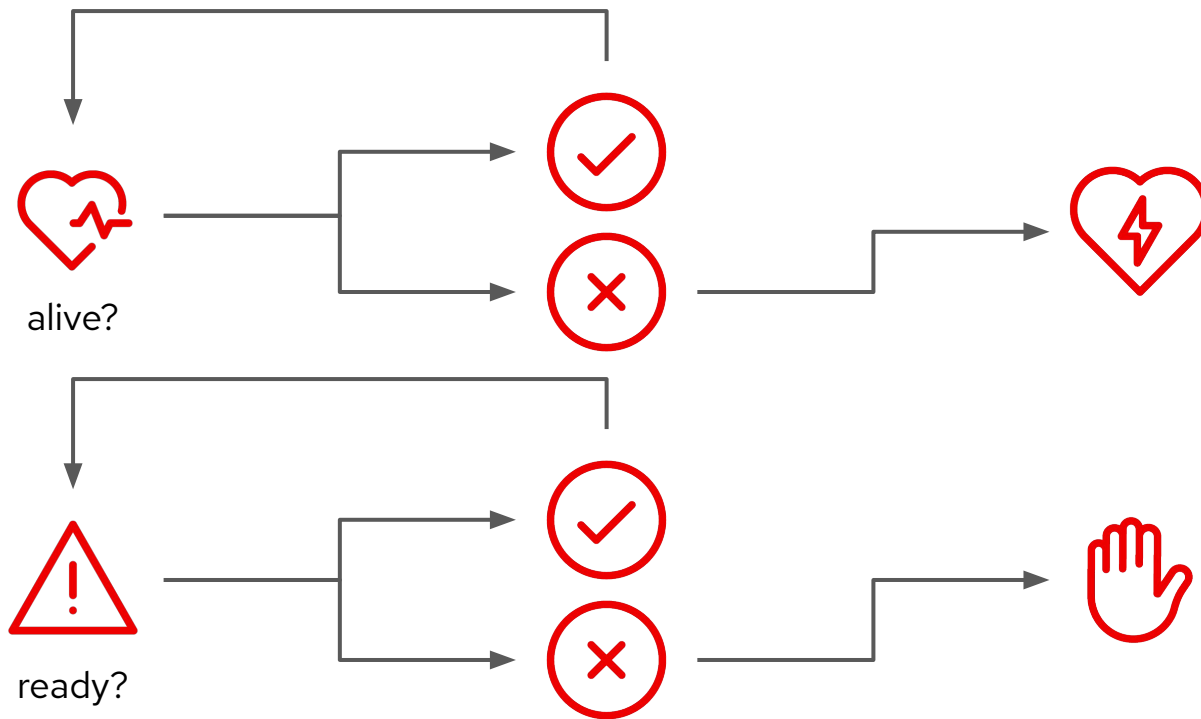
routes make services accessible to clients outside the environment via real-world urls



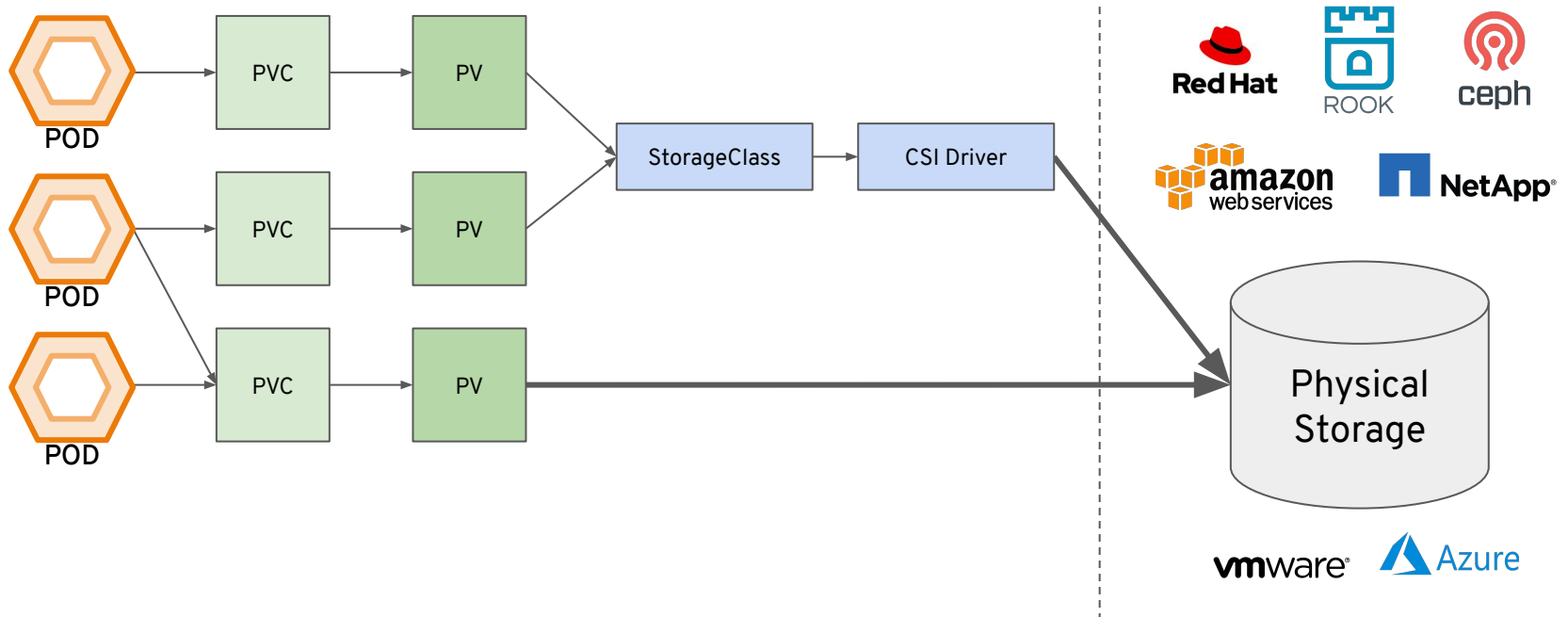
Persistent Volume and Claims



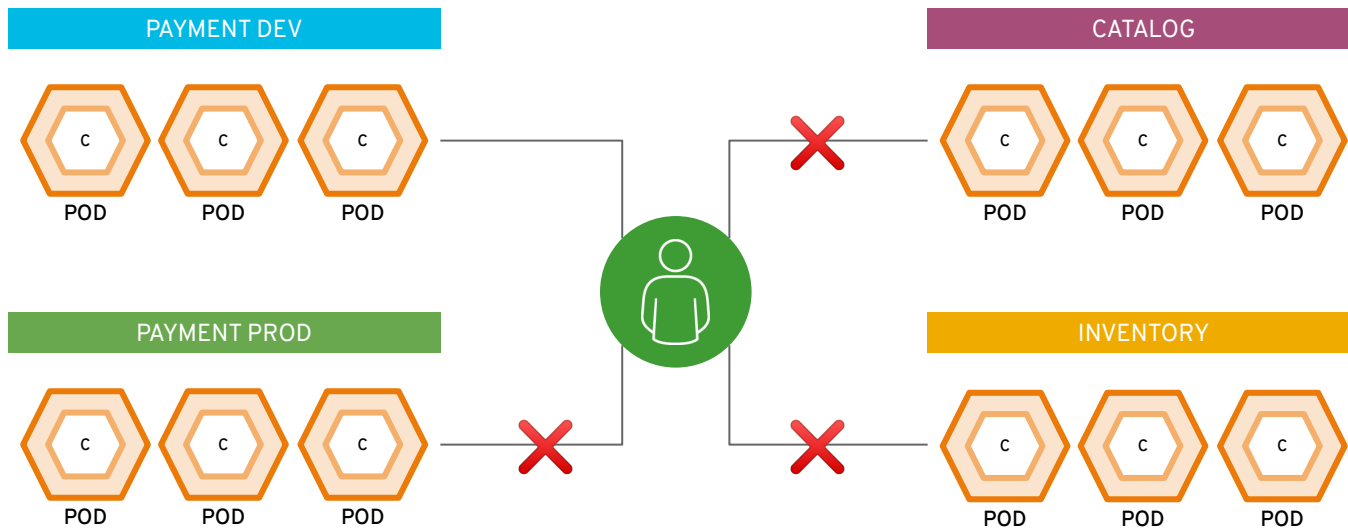
Liveness and Readiness



Persistent Volume and Claims



projects isolate apps across environments, teams, groups and departments





OpenShift 4 Architecture

your choice of infrastructure

COMPUTE

NETWORK

STORAGE

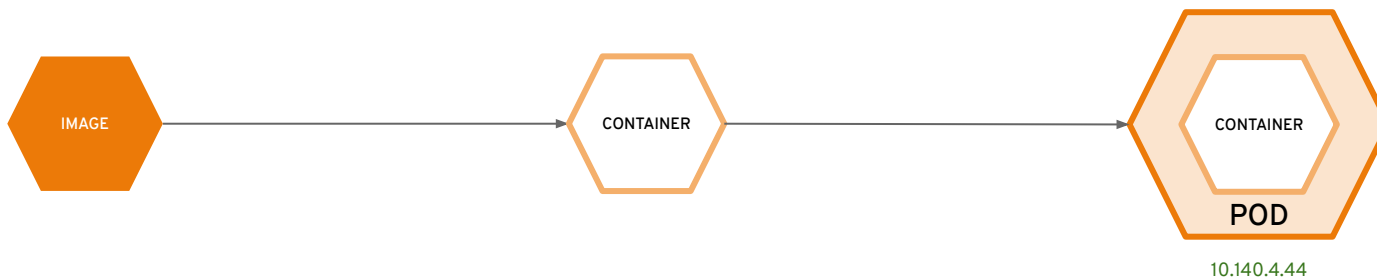
workers run workloads



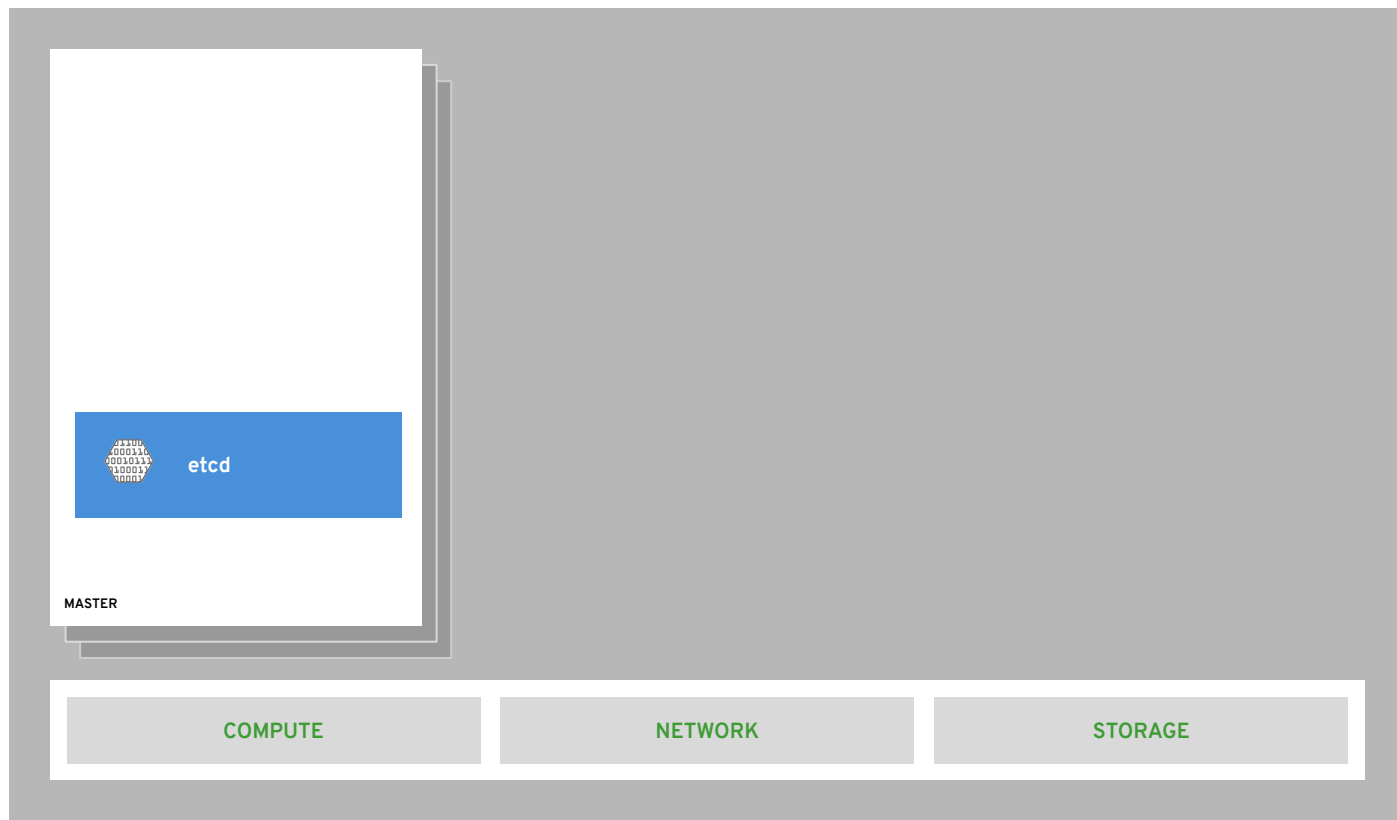
masters are the control plane



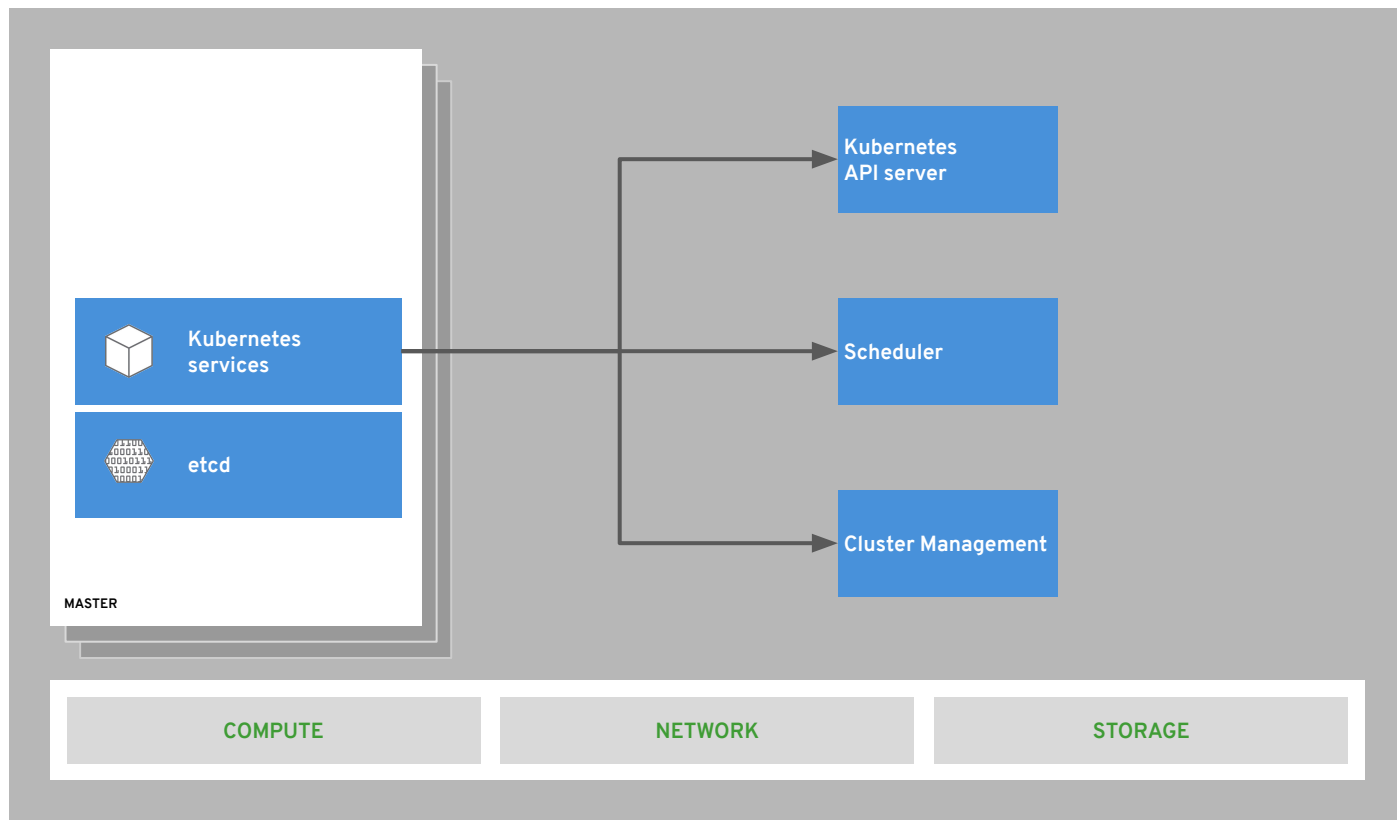
everything runs in pods



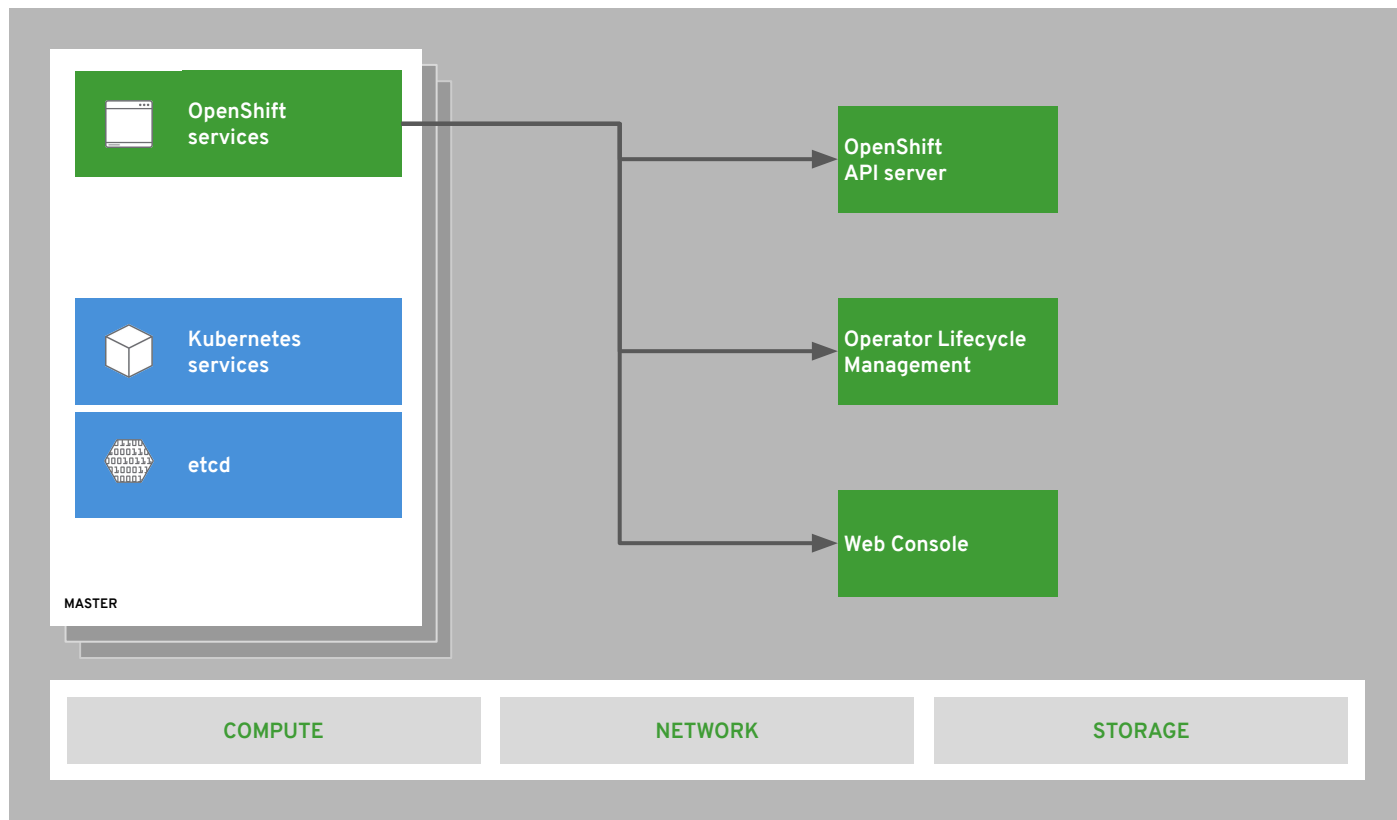
state of everything



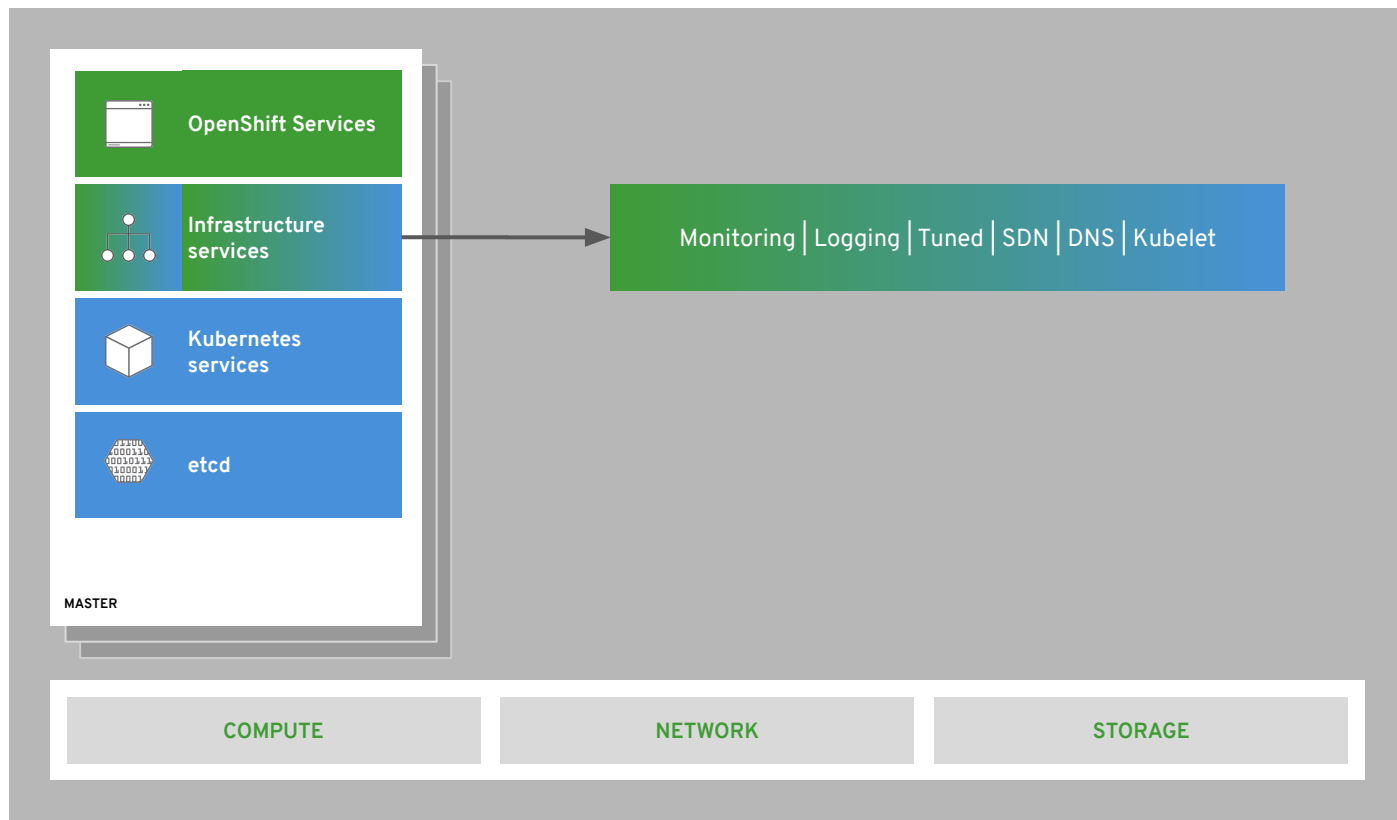
core kubernetes components



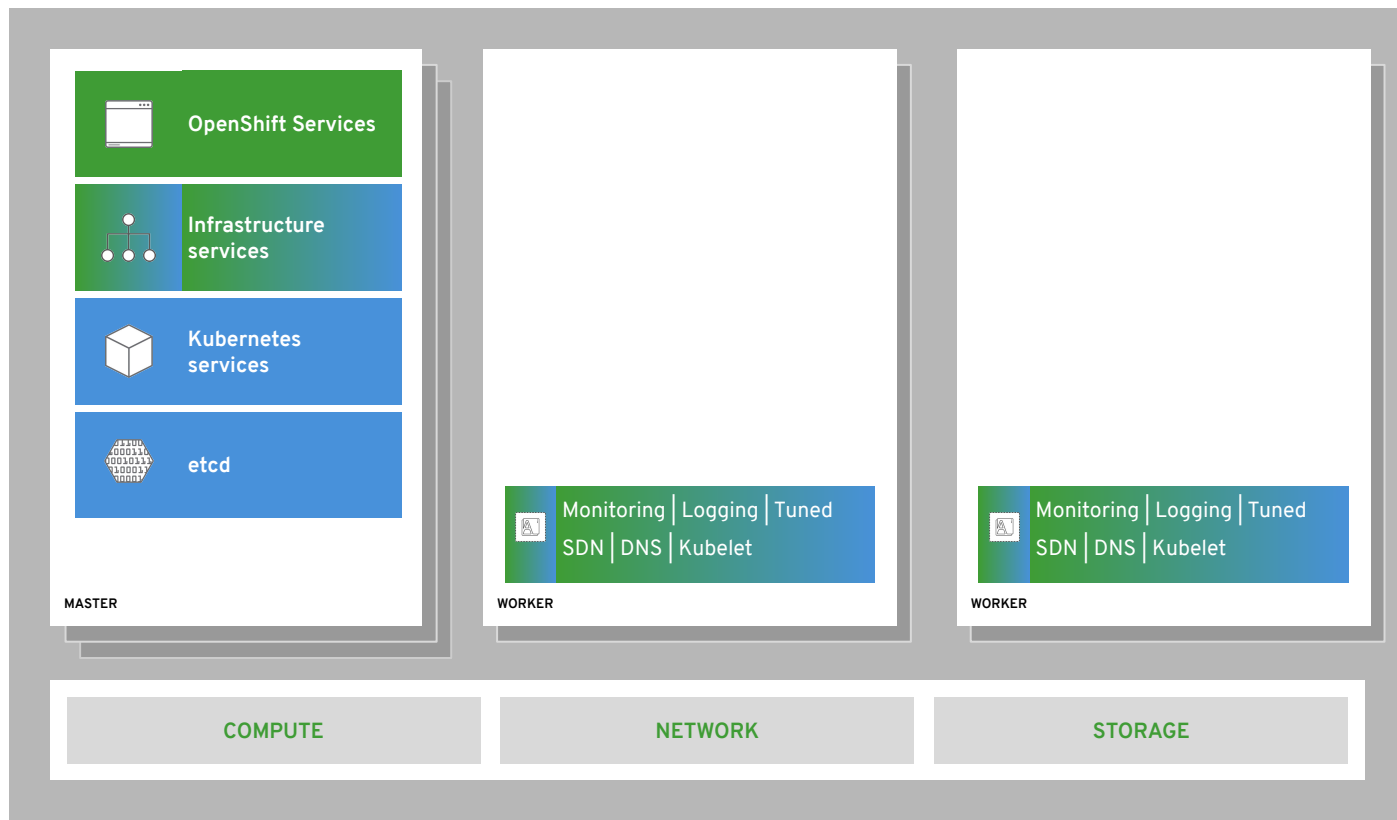
core OpenShift components



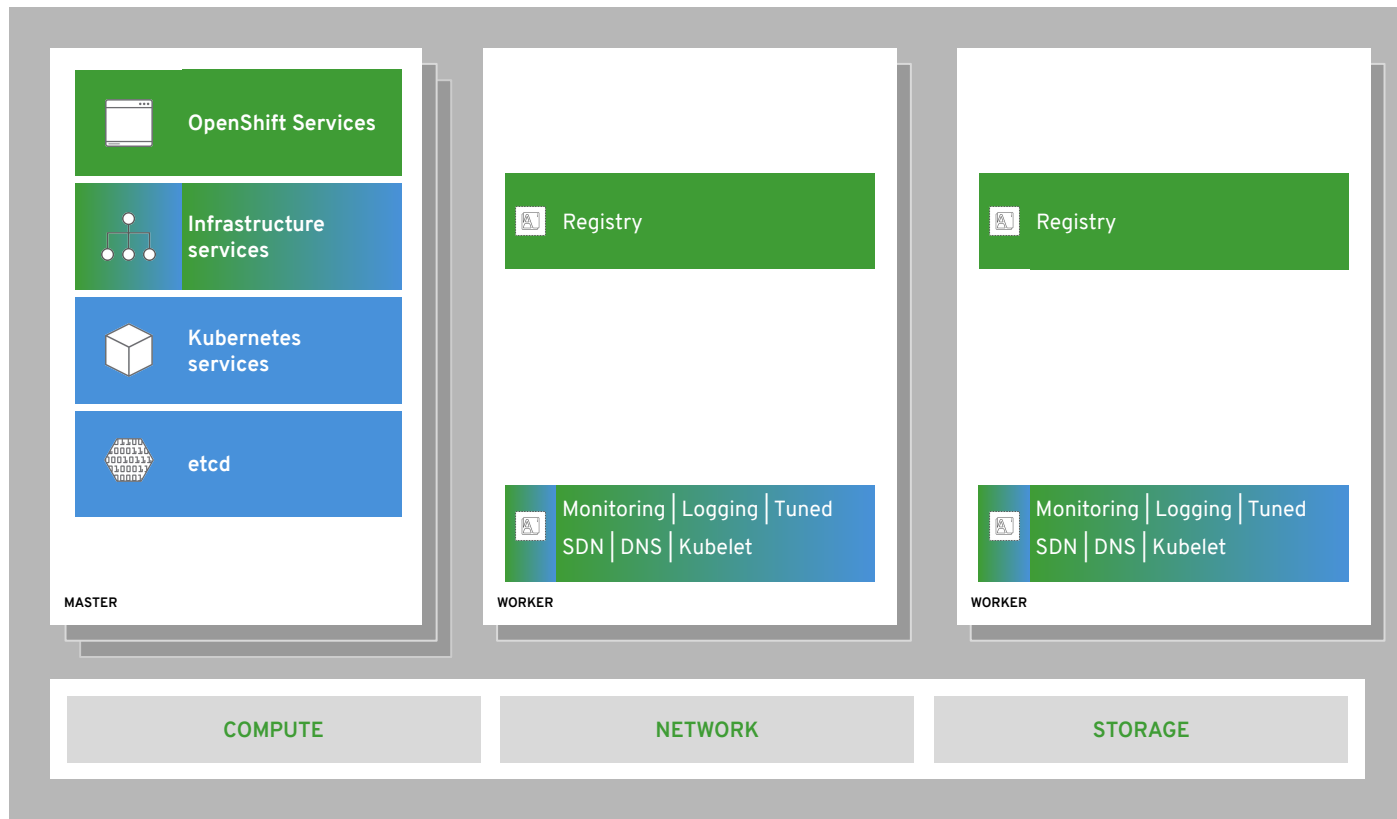
internal and support infrastructure services



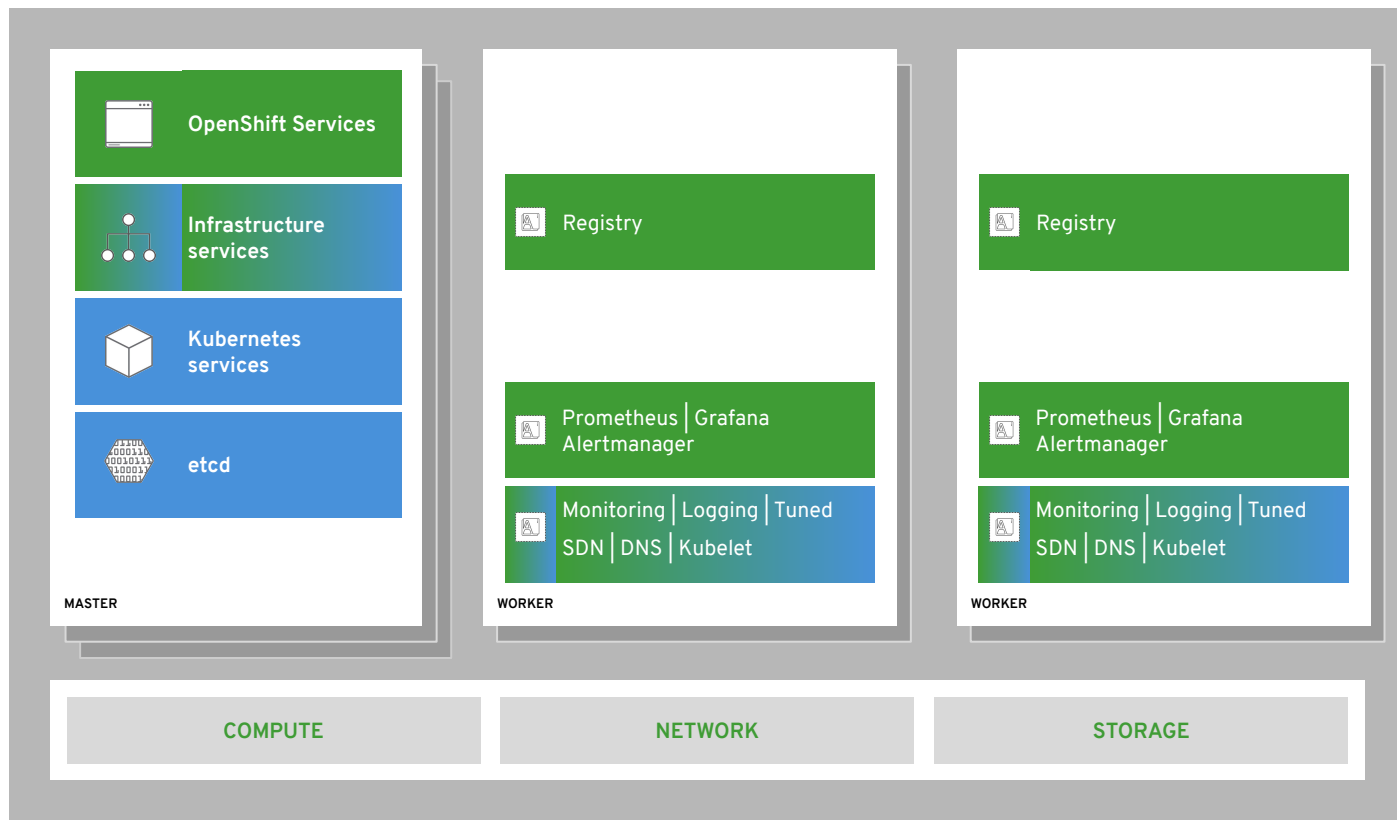
run on all hosts



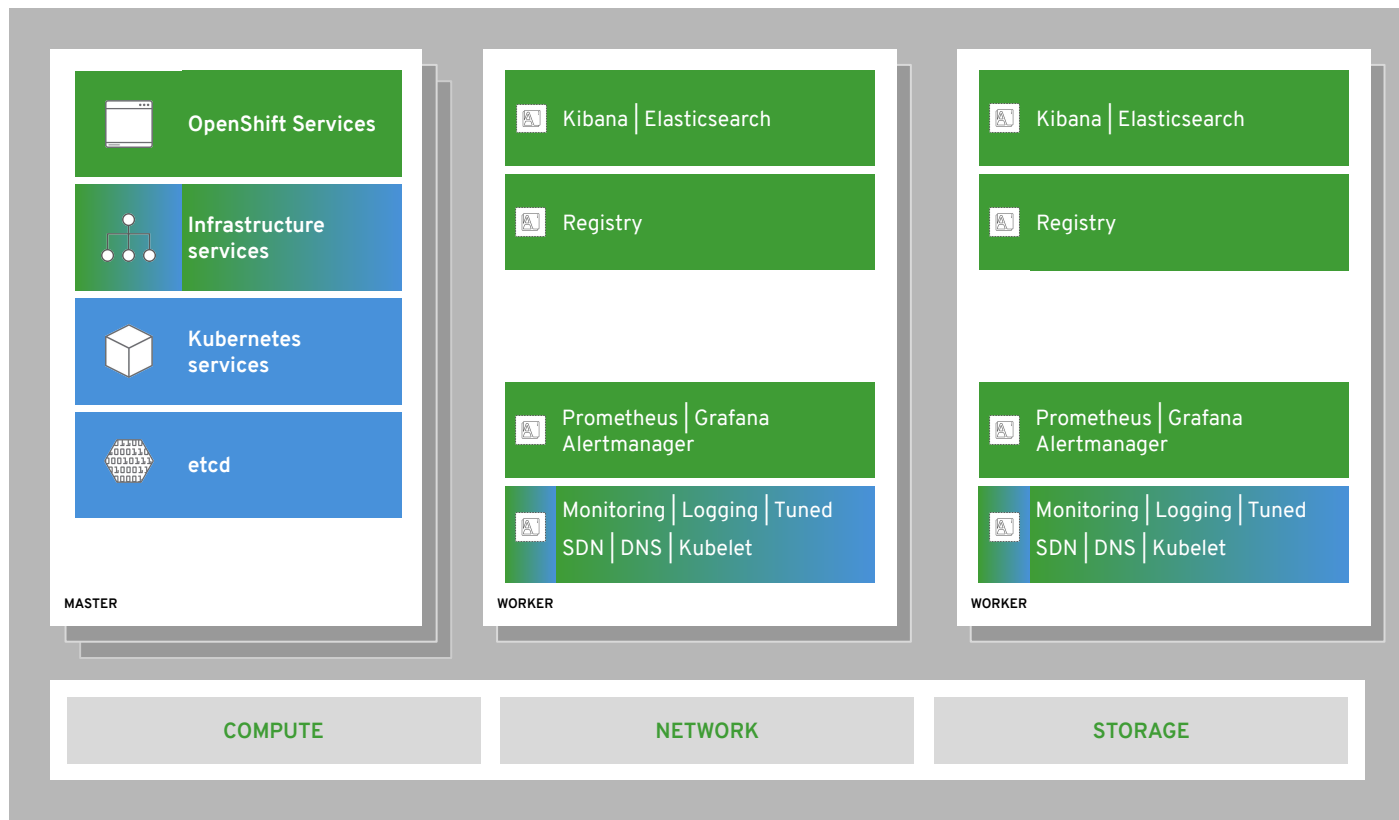
integrated image registry



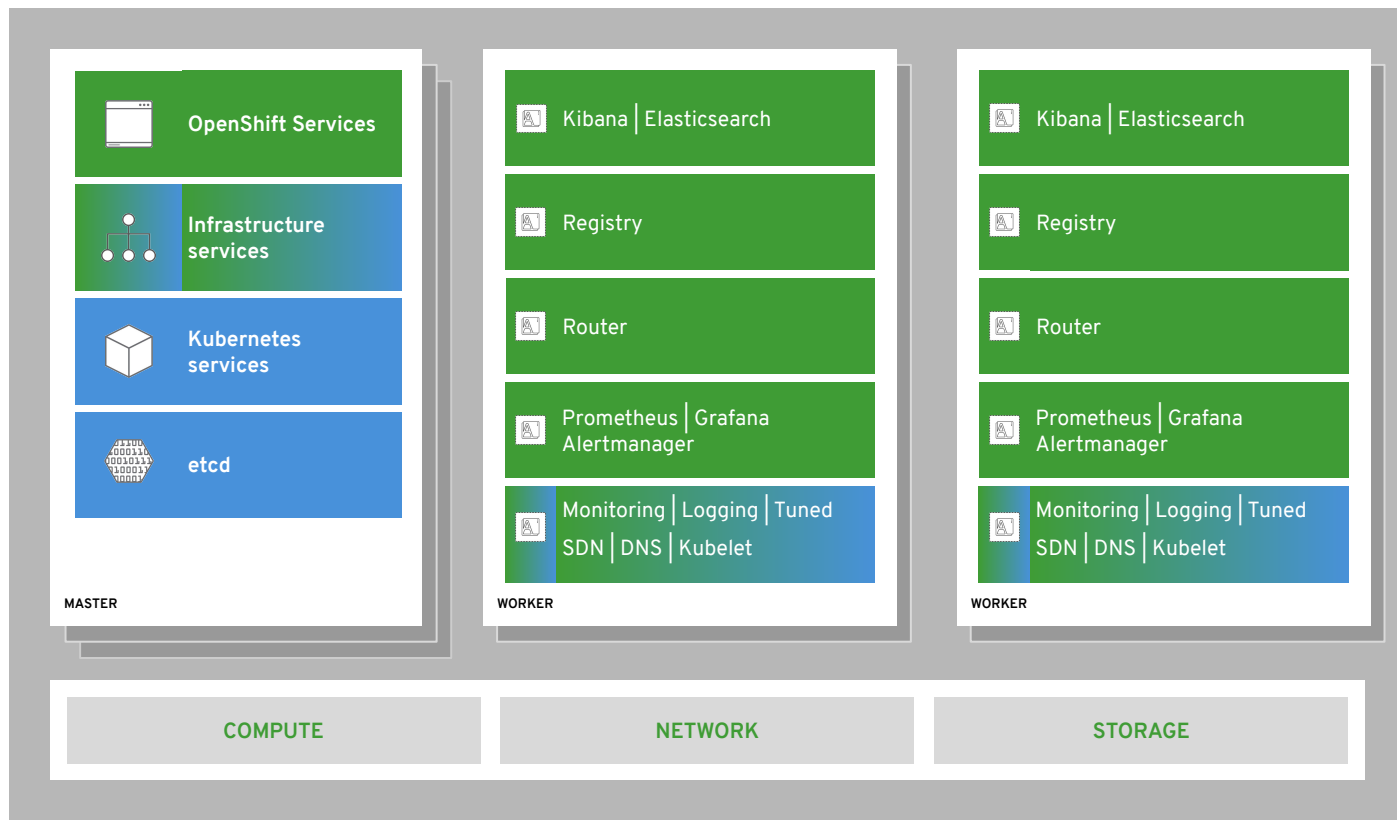
cluster monitoring



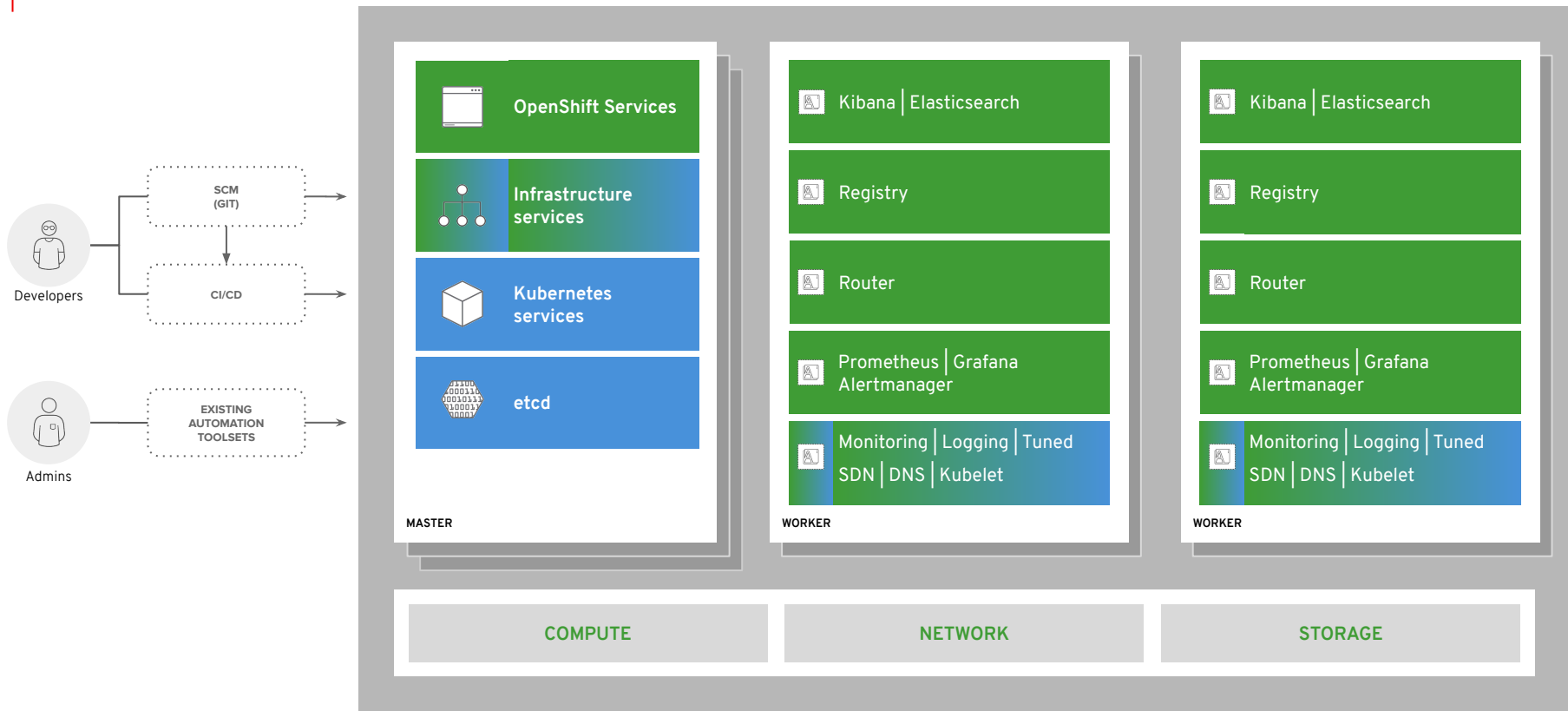
log aggregation



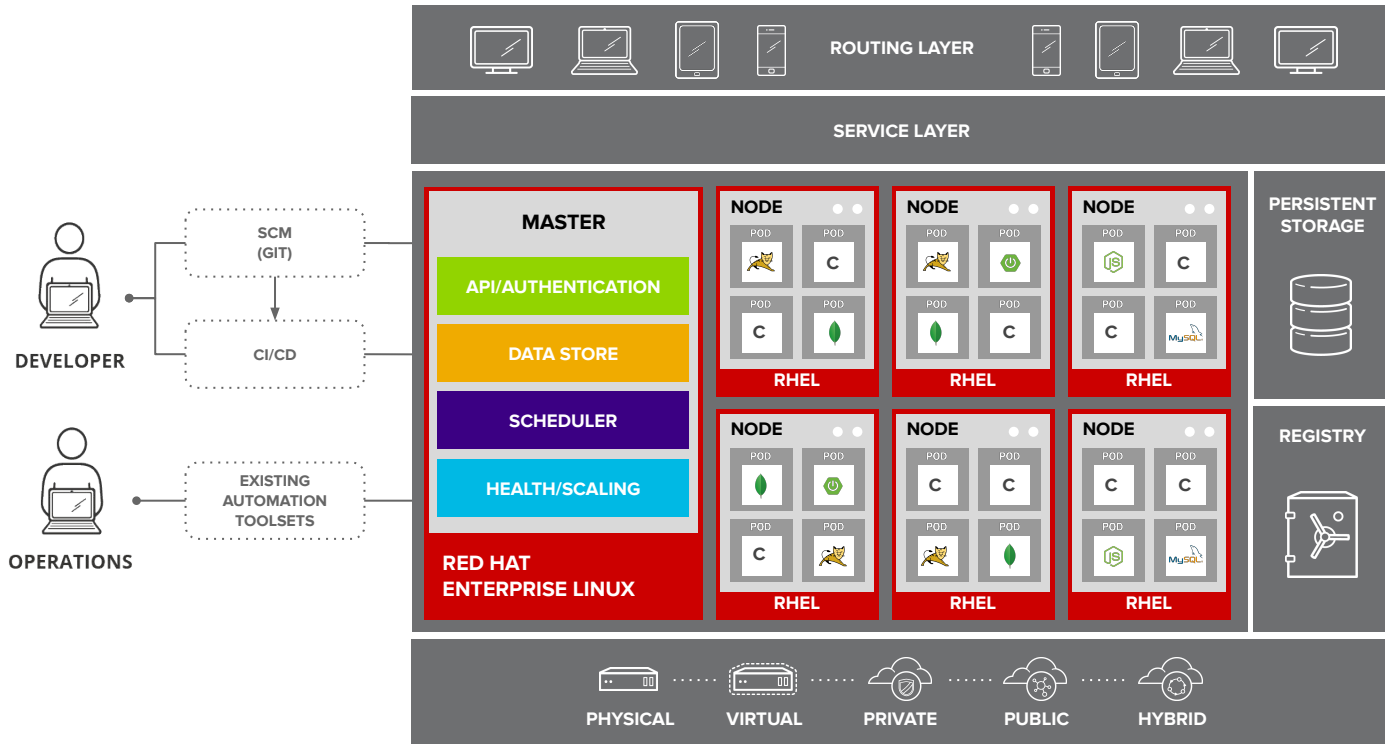
integrated routing

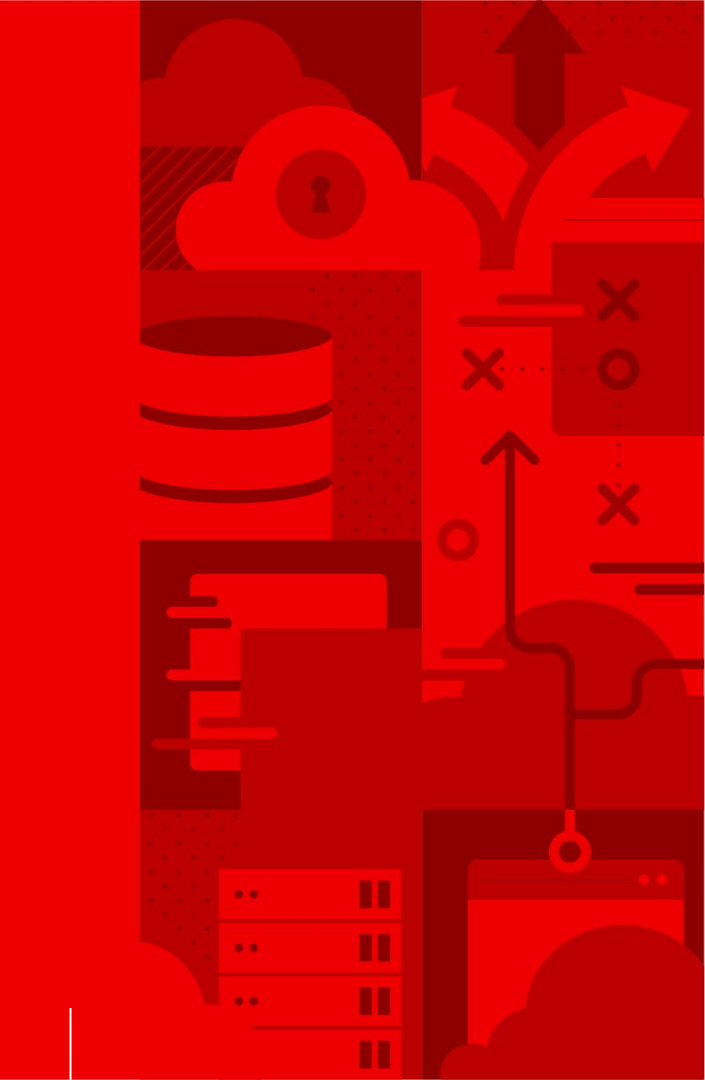


dev and ops via web, cli, API, and IDE




Another Architectural visualization







Monitoring Application Health

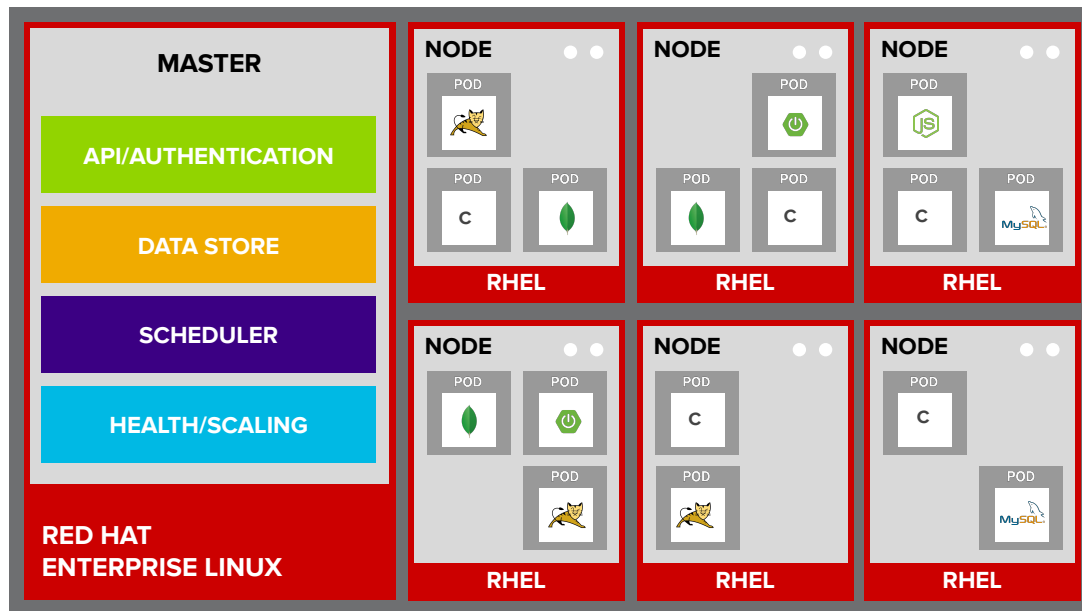
 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

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

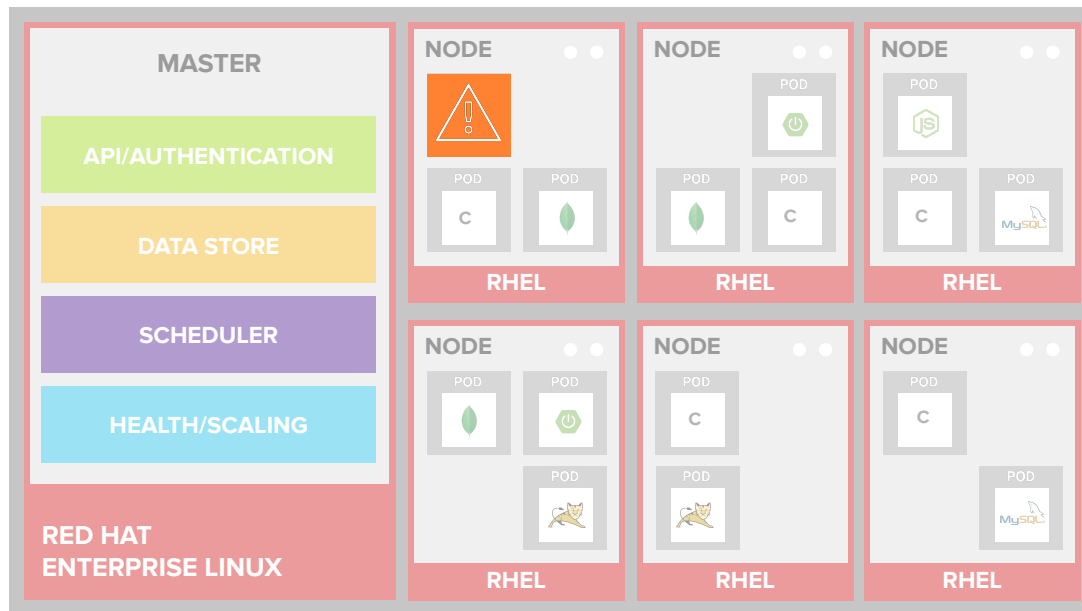
 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 twitter.com/RedHat

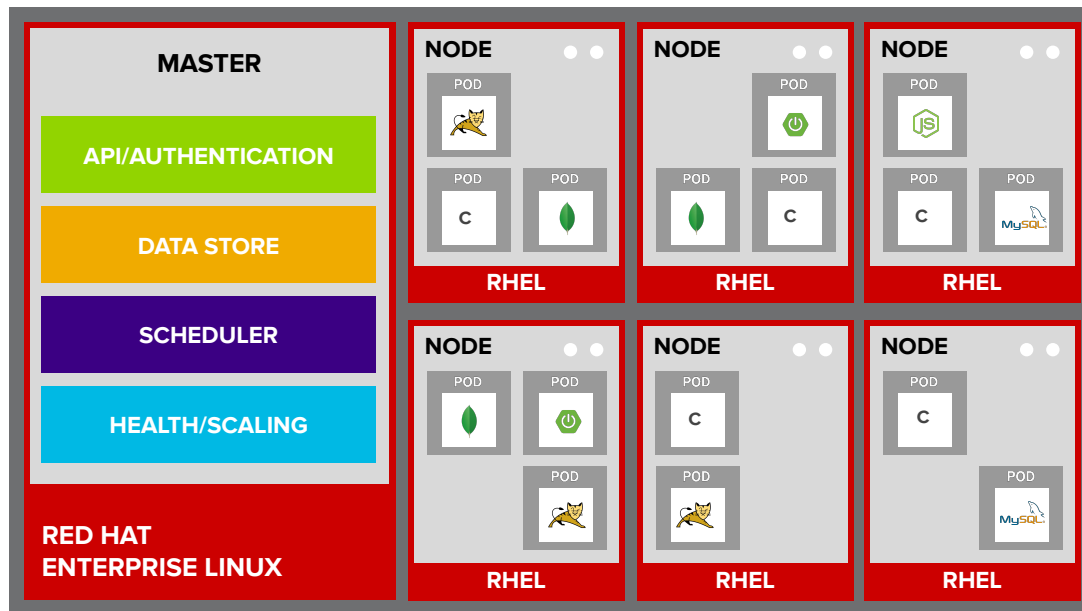
AUTO-HEALING FAILED PODS



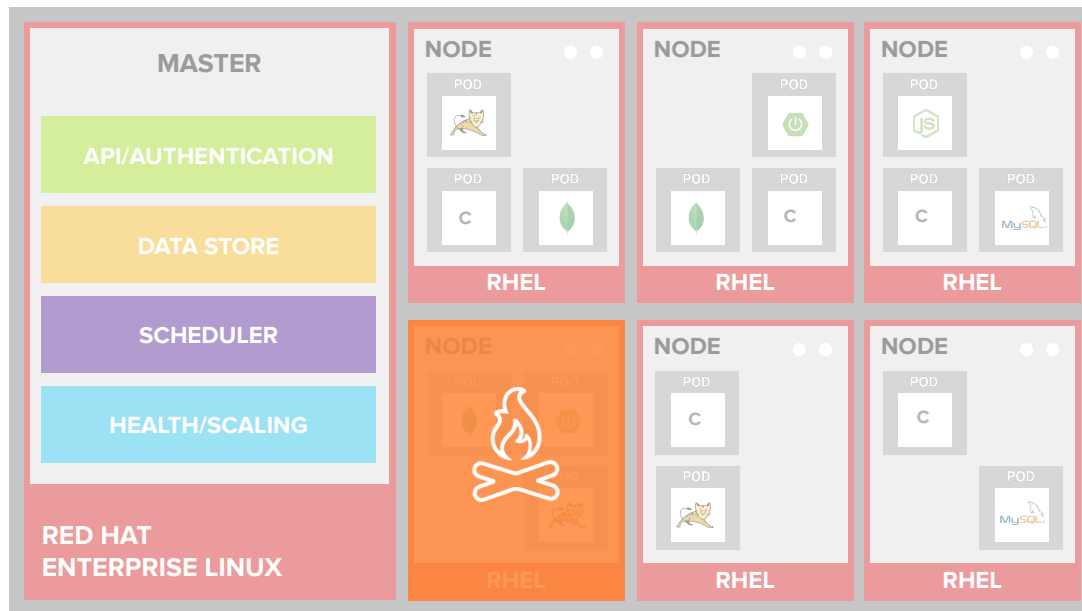
AUTO-HEALING FAILED CONTAINERS



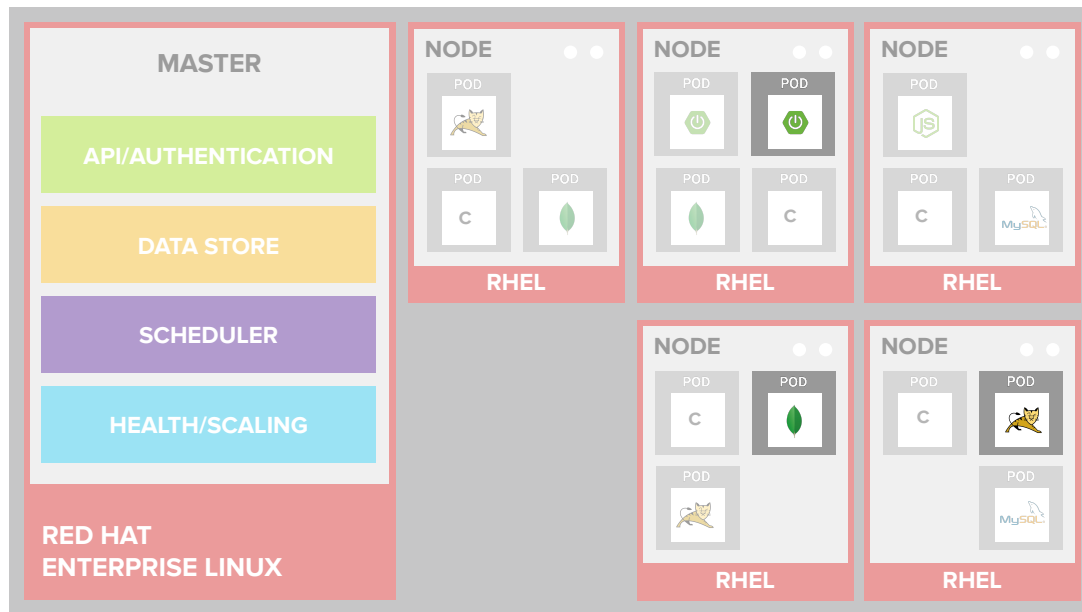
AUTO-HEALING FAILED CONTAINERS



AUTO-HEALING FAILED CONTAINERS



AUTO-HEALING FAILED CONTAINERS





OpenShift lifecycle, installation & upgrades

OpenShift 4 Installation

Two new paradigms
for deploying clusters

Installation Paradigms

OPENSIFT CONTAINER PLATFORM

Full Stack Automated

Simplified opinionated “Best Practices” for cluster provisioning

Fully automated installation and updates including host container OS.



Pre-existing Infrastructure

Customer managed resources & infrastructure provisioning

Plug into existing DNS and security boundaries



HOSTED OPENSIFT

Azure Red Hat OpenShift

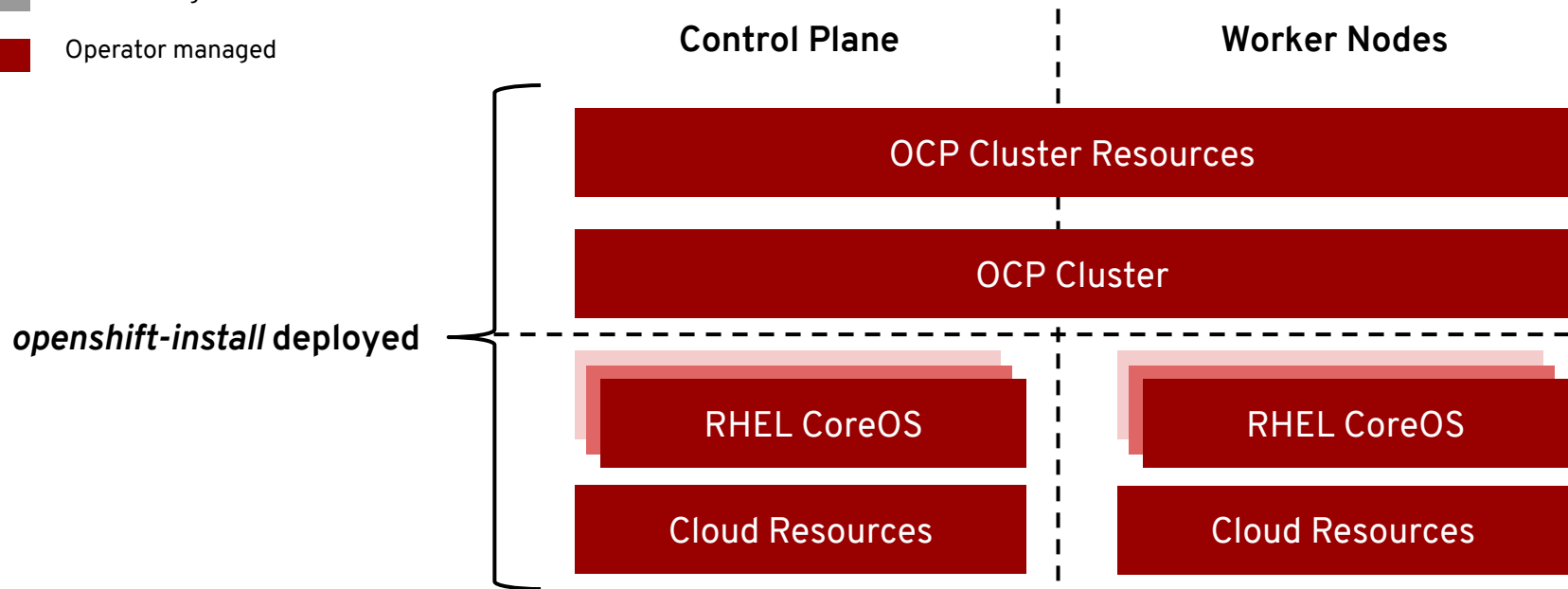
Deploy directly from the Azure console. Jointly managed by Red Hat and Microsoft Azure engineers.

OpenShift Dedicated

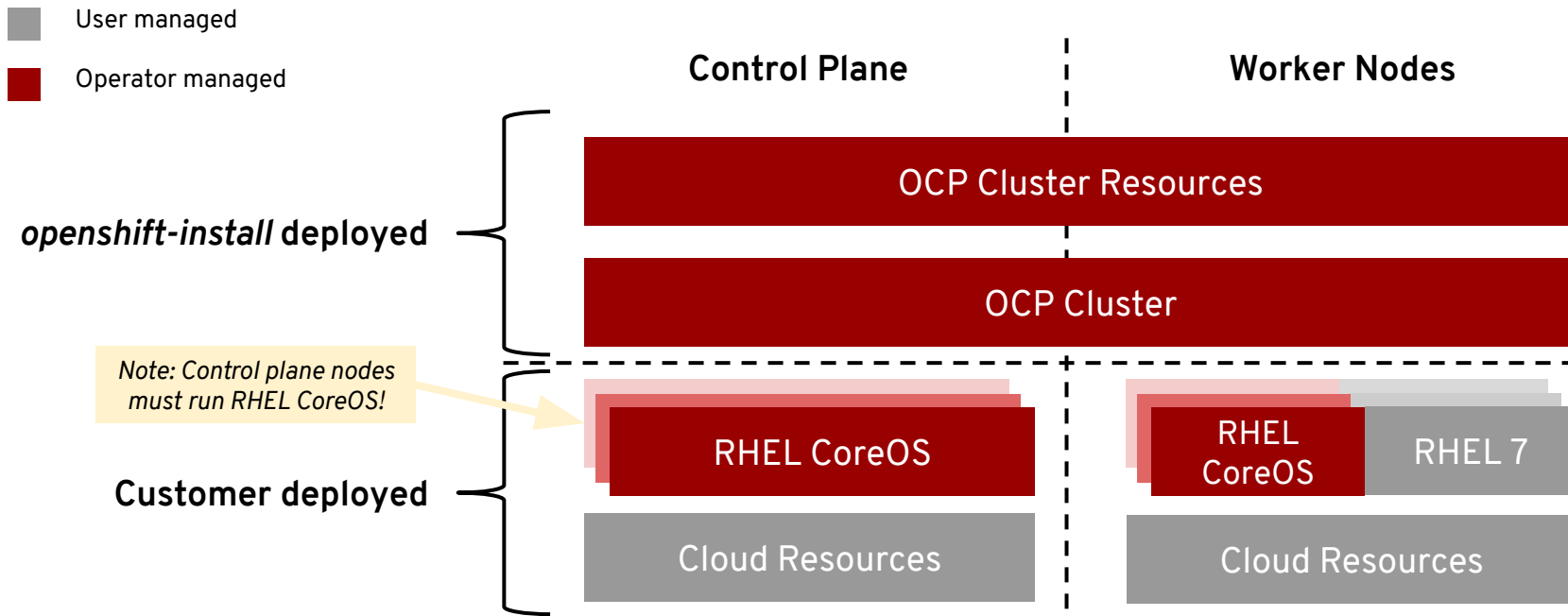
Get a powerful cluster, fully Managed by Red Hat engineers and support.

Full-stack Automated Installation

- User managed
- Operator managed



Pre-existing Infrastructure Installation



Comparison of Paradigms

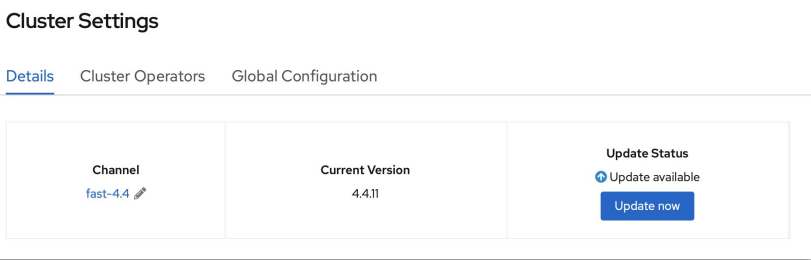
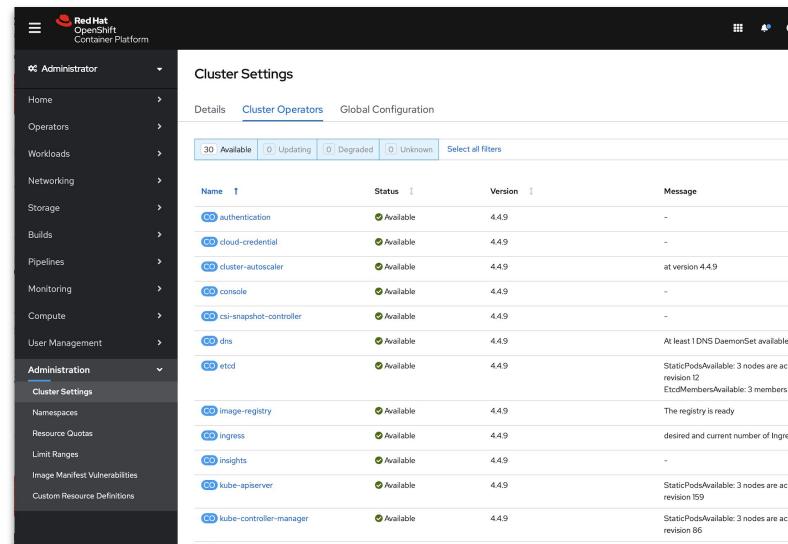
| | Full Stack Automation | Pre-existing Infrastructure |
|---------------------------------|------------------------|---|
| Build Network | Installer | User |
| Setup Load Balancers | Installer | User |
| Configure DNS | Installer | User |
| Hardware/VM Provisioning | Installer | User |
| OS Installation | Installer | User |
| Generate Ignition Configs | Installer | Installer |
| OS Support | Installer: RHEL CoreOS | User: RHEL CoreOS + RHEL 7 |
| Node Provisioning / Autoscaling | Yes | Only for providers with OpenShift Machine API support |

OpenShift 4 Lifecycle

Supported paths for
upgrades and
migrations

Each OpenShift release is a collection of Operators

- 100% automated, in-place upgrade process
- 30 Operators run every major part of the platform:
 - Console, Monitoring, Authentication, Machine management, Kubernetes Control Plane, etcd, DNS, and more.
- Operators constantly strive to meet the desired state, merging admin config and Red Hat recommendations
- CI testing is constantly running install, upgrade and stress tests against groups of Operators



OpenShift Upgrades and Migrations

Happy path = upgrade through each version

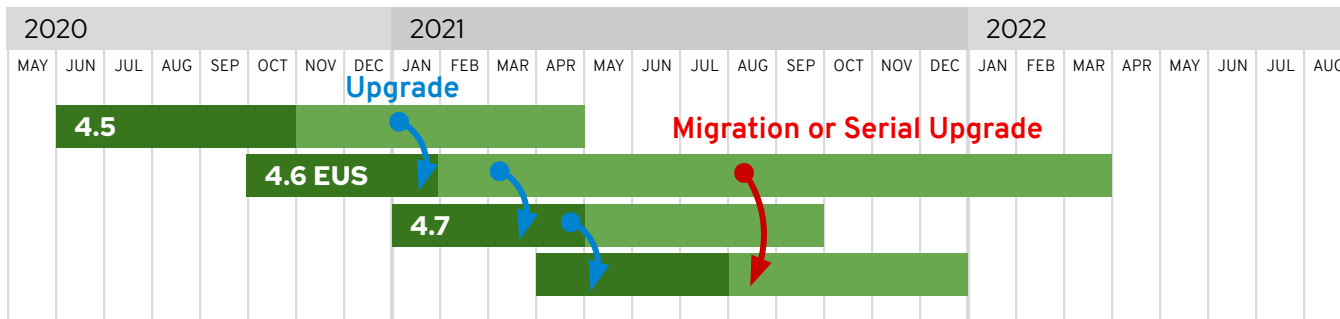
- On a regular cadence, upgrade to the next supported version.

Optional path = migration tooling

- To skip versions or catch up, use the application migration tooling to move to a new cluster.

What is Extended Update Support (EUS) ?

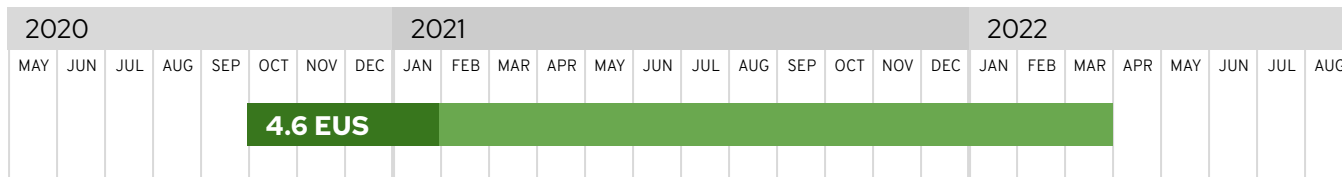
- Extended timeframe for critical security and bug fixes
- Work within a customer’s release management philosophies
- Goal to provide a serial pathway to update from EUS to EUS
 - Augmented by Migration Tool and/or Advanced Cluster Management (ACM) based on use-case



■ N release
Full support, RFEs, bugfixes, security

■ N-2 release
OTA pathway to N release, critical bugs and security

4.6 EUS for Layered Products/Add-ons



Complete "hands off" EUS

Remain on single supported version for the entire EUS period

- OpenShift Logging
- OpenShift Container Storage
- Advanced Cluster Manager
- Process Automation
- OpenShift CNF
- Jaeger

Mid-cycle refresh during EUS

The EUS cycles for these products refresh during the OpenShift EUS

- Cluster Migration Tool
- Red Hat SSO
- JBoss EAP
- Quarkus
- Thorntail
- Spring Boot
- Vert.x
- JWS (Tomcat)
- DataGrid

LAYERED PRODUCT UPGRADE

Normal updates during EUS

Follows the normal support window for the add-on, shorter than EUS

- OpenShift Virtualization
- OpenShift Serverless
- OpenShift Pipelines
- OpenShift Service Mesh
- CodeReady Containers
- Red Hat Quay / CSO

LAYERED UPGRADE

LAYERED UPGRADE

LAYERED UPGRADE


LAYERED UPGRADE

LAYERED UPGRADE





DEMO TIME

Let's play with OpenShift

 [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

No local installation needed

- <https://learn.openshift.com>
 - Free courses and demo systems to use
- Everyone is encouraged to follow along on their own systems
- Use “cloud.redhat.com” if you want to install on your own
- <https://docs.openshift.com>
 - All documentation to OKD and Enterprise OpenShift, Managed OpenShift etc.

Getting Started with OpenShift

- Navigate to [Learn.openshift.com](https://learn.openshift.com)
- If you do not have an account, create one - login
- Click “Getting Started with ...”
- We’ll learn how to:
 - Login
 - Grant access
 - Deploy existing images
 - Scale images
 - Building from source

Red Hat OpenShift

PRODUCTS ▾ LEARN ▾ COMMUNITY ▾ SUPPORT ▾ FREE TRIAL REPORT AN ISSUE

Interactive Learning Portal

Our Interactive Learning Scenarios provide you with a pre-configured OpenShift® instance, accessible from your browser without any downloads or configuration. Use it to experiment, learn OpenShift and see how we can help solve real-world problems.

- Getting Started with OpenShift for Developers
- Logging in to an OpenShift Cluster
- Developing with `odo`
- Deploying Applications From Images
- Deploying Applications From Source
- Using the CLI to Manage Resource Objects

WOAH - a lot of stuff happened!

- Yes - container platforms are complex with a lot of moving parts
- We're focusing on "getting started" here - a lot of things happened we didn't talk about or focus on.
- BUT
 - The Developer experience was simple and straightforward
 - Fast - from code to running in less than 30 seconds
 - Easy to scale - easy to recover from issues
 - Lots of languages and platforms (anything that can run on Linux)
- Advanced deployments are more complex
- If time permits, we can look a bit behind the scenes of what is actually going on.

You can run any code - learn.openshift.com

- Experiment - you don't have to just run the code mentioned in the guide
 - Try other containers from other container registries
 - Try your own code
- S2I is a open source project to automate builds of software
 - <https://github.com/openshift/okd> covers how to use it with a wide set of languages
- More developer focused demos
 - <https://developers.redhat.com/courses/openshift>
 - <https://developers.redhat.com/developer-sandbox>
 - Goes a lot deeper with advanced deployments, dependencies etc.

Important concepts

- **Projects/Namespaces**

Placeholder for objects - used for security, collaboration etc

- **API**

Server entry point - all interaction goes through the API

- Command line tool: **oc** and **kubectl**

Note you can use kubectl from other builds

“oc” has all OpenShift extensions built into it

- **Deployment**

An object defining a pod (container) runtime

- **Route or Ingress**

Entry point from the outside to runtime

Concepts - continued

- **Service**
Internal network exposure of a pod - think DNS and firewall in one
- **Build**
Source-to-Image taking a git-repository and turning it into a binary container
- **Secret**
Stored encrypted and protected data used for credentials and similar
- **ConfigMap**
Clear text configuration data that can be exposed to a pod using vary ways



THANK YOU