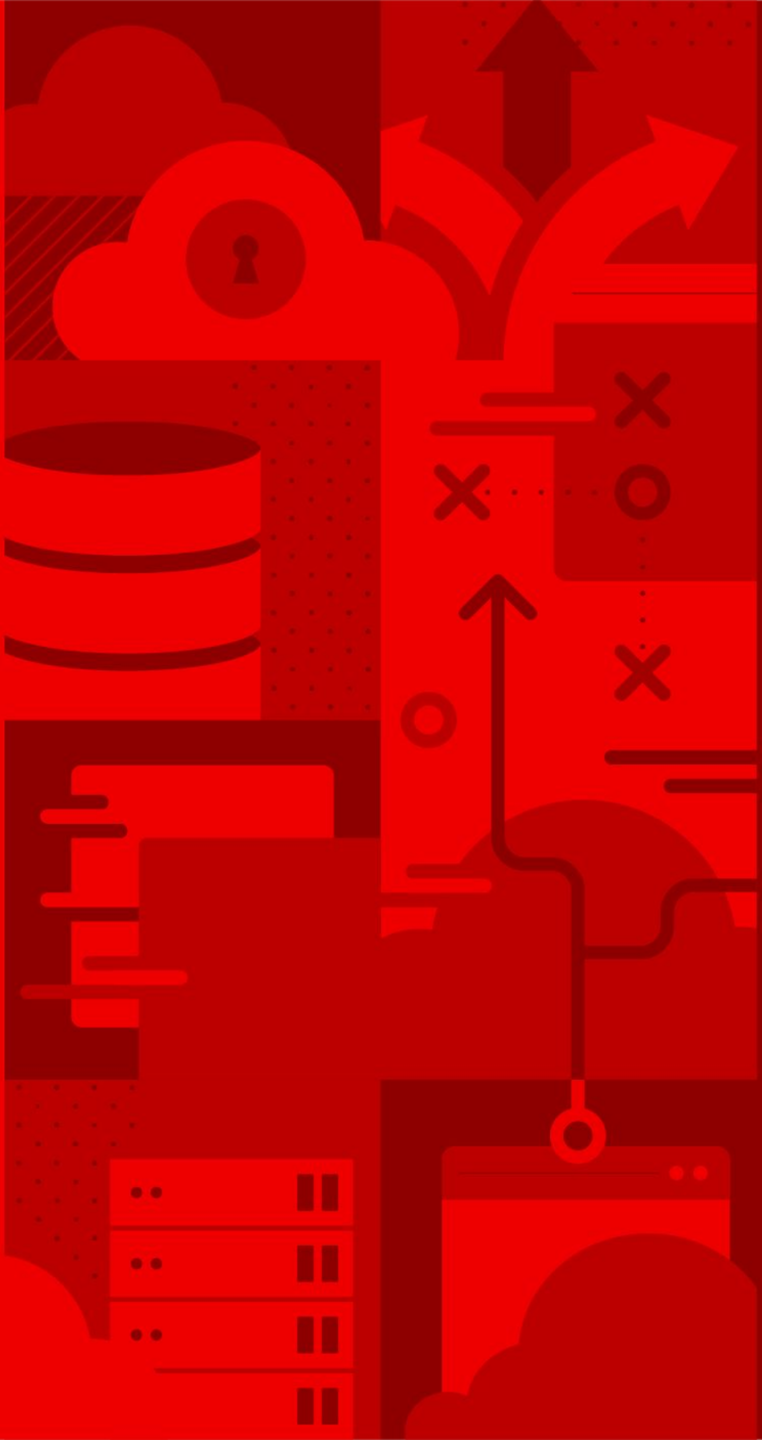
Red Hat
OpenShift
Container Platform

OpenShift Architecture

As Part of OpenShift Architecture Workshop



Functional overview

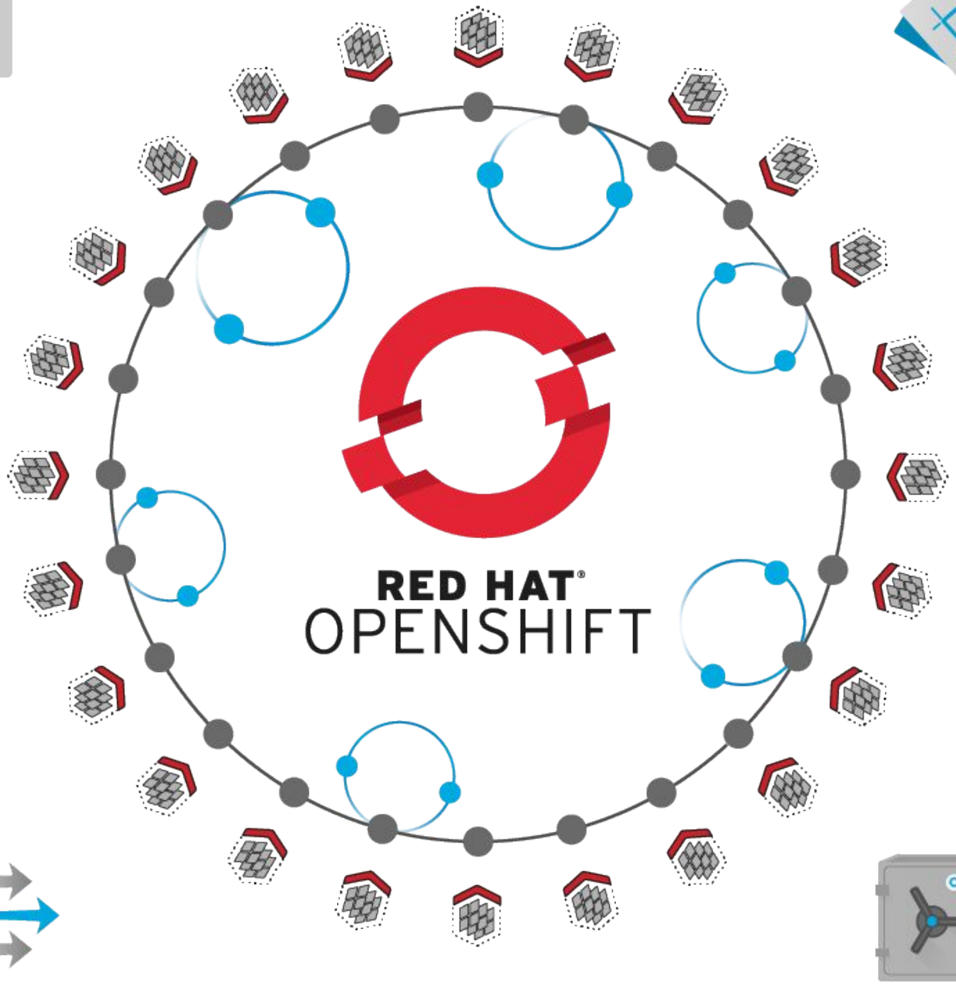
Self-Service

Multi-language

Automation

Collaboration

Multi-tenant



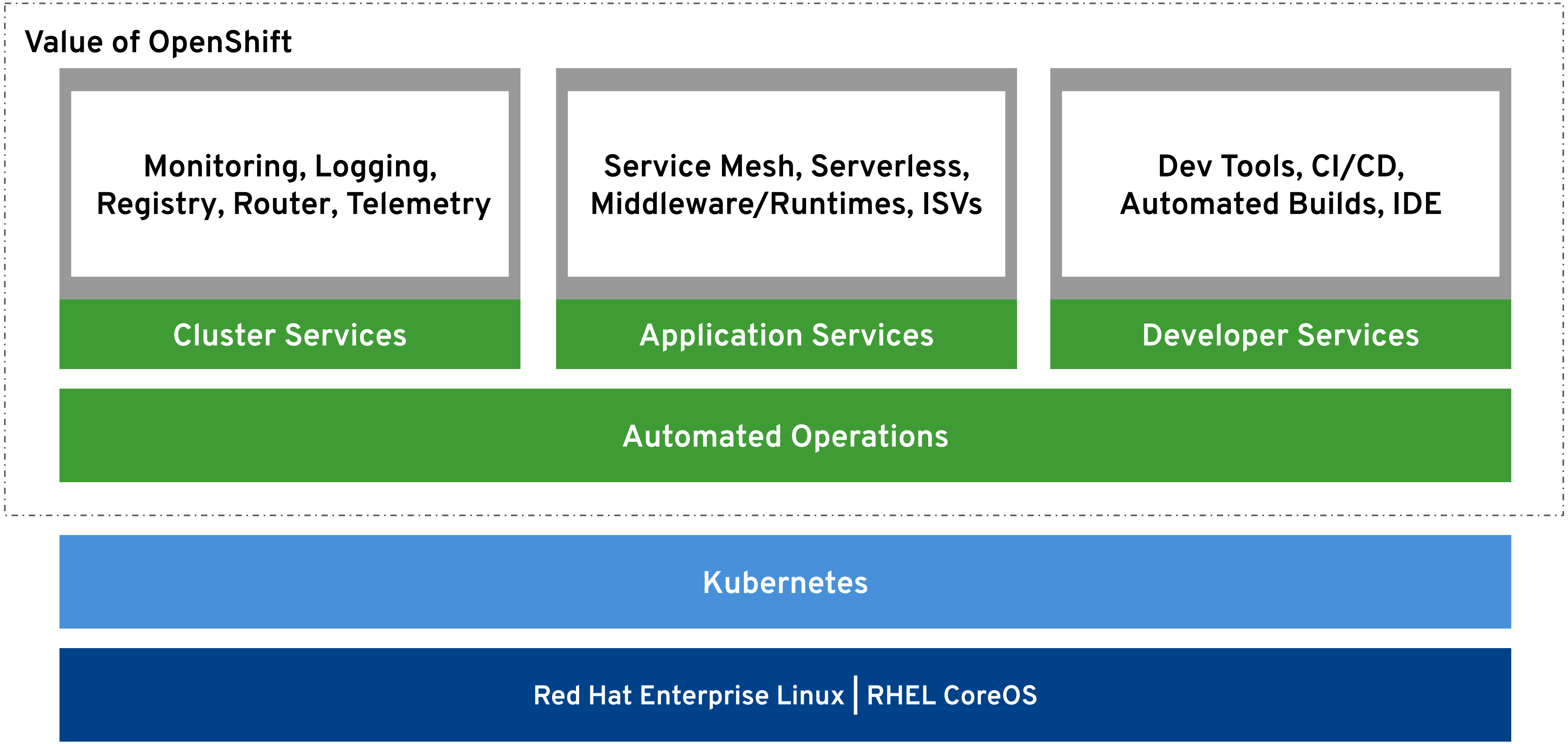
Standards-based

Web-scale

Open Source

Enterprise Grade

Secure



Value of OpenShift

Monitoring, Logging, Registry, Router, Telemetry

Service Mesh, Serverless, Middleware/Runtimes, ISVs

Dev Tools, CI/CD, Automated Builds, IDE

Cluster Services

Application Services

Developer Services

Automated Operations

Kubernetes

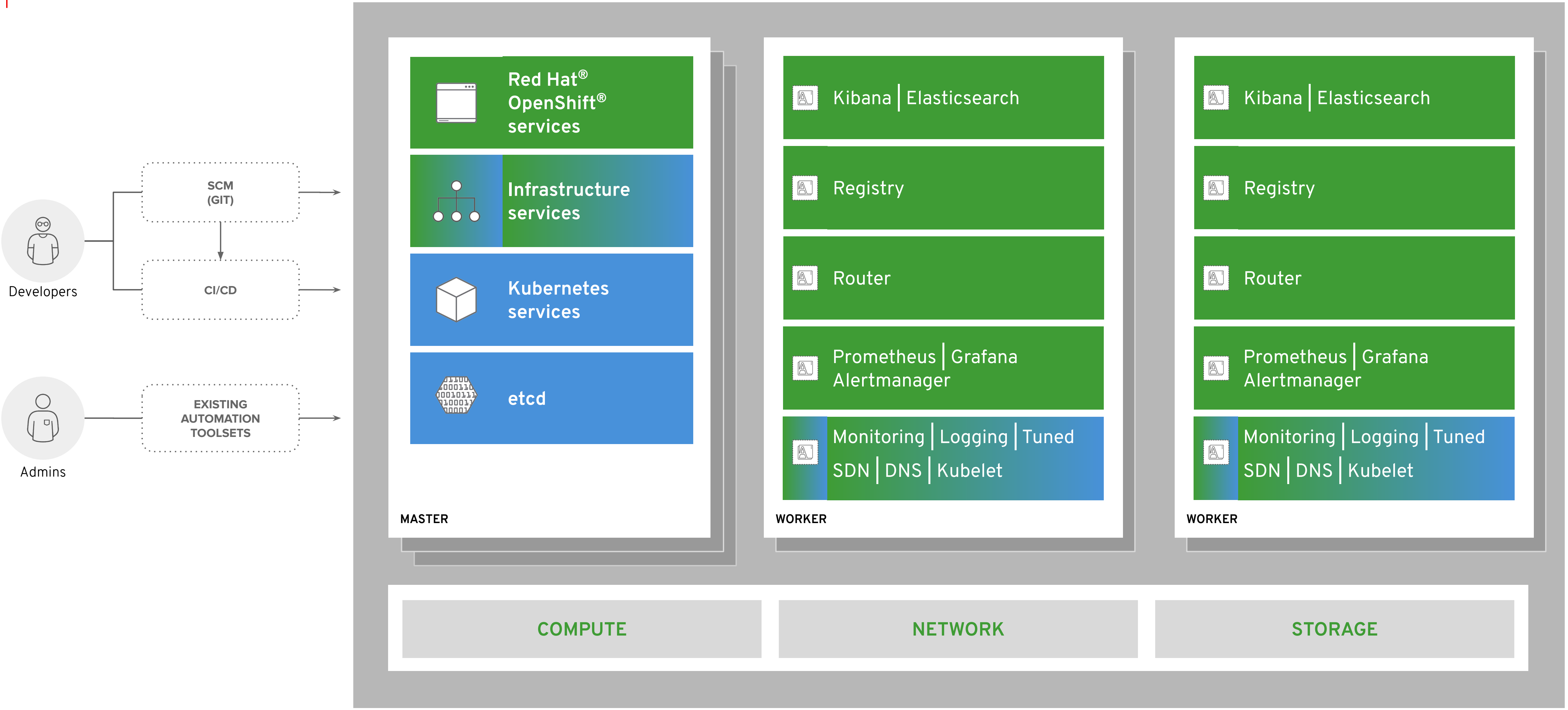
Red Hat Enterprise Linux | RHEL CoreOS

Best IT Ops Experience

CaaS ⟷ PaaS ⟷ FaaS

Best Developer Experience

OPENSHIFT CONTAINER PLATFORM | Architectural Overview



Developers

SCM (GIT)

CI/CD

Admins

EXISTING AUTOMATION TOOLSETS

MASTER

Red Hat® OpenShift® services

Infrastructure services

Kubernetes services

etcd

WORKER

Kibana | Elasticsearch

Registry

Router

Prometheus | Grafana Alertmanager

Monitoring | Logging | Tuned

SDN | DNS | Kubelet

WORKER

Kibana | Elasticsearch

Registry

Router

Prometheus | Grafana Alertmanager

Monitoring | Logging | Tuned

SDN | DNS | Kubelet

COMPUTE

NETWORK

STORAGE

CNCF Cloud Native Landscape

2019-07-12T03:51:02Z 024e93b

Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io

Greyed logos are not open source

App Definition and Development

Database

KV CNCF Incubating

Vitess CNCF Incubating

Streaming & Messaging

NATS CNCF Incubating

Application Definition & Image Build

HELM CNCF Incubating

Continuous Integration & Delivery

Orchestration & Management

Scheduling & Orchestration

kubernetes CNCF Graduated

Coordination & Service Discovery

CoreDNS CNCF Graduated

etcd CNCF Incubating

Remote Procedure Call

gRPC CNCF Incubating

Service Proxy

envoy CNCF Graduated

API Gateway

Service Mesh

LINKERD CNCF Incubating

Runtime

Cloud-Native Storage

ROOK CNCF Incubating

Container Runtime

containerd CNCF Graduated

cri-o CNCF Incubating

rkt CNCF Incubating

Cloud-Native Network

CNI CNCF Incubating

Provisioning

Automation & Configuration

Container Registry

HARBOR CNCF Incubating

Security & Compliance

Notary CNCF Incubating

Open Policy Agent CNCF Incubating

TUF CNCF Incubating

Key Management

Platform

Certified Kubernetes - Distribution

Certified Kubernetes - Hosted

Certified Kubernetes - Installer

PaaS/Container Service

Observability and Analysis

Monitoring

Prometheus CNCF Graduated

Logging

fluentd CNCF Graduated

Tracing

JAEGER CNCF Incubating

OPENTRACING CNCF Incubating

Chaos Engineering

Serverless

Cloud

Public

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path

l.cncf.io

CLOUD NATIVE COMPUTING FOUNDATION

CLOUD NATIVE Landscape

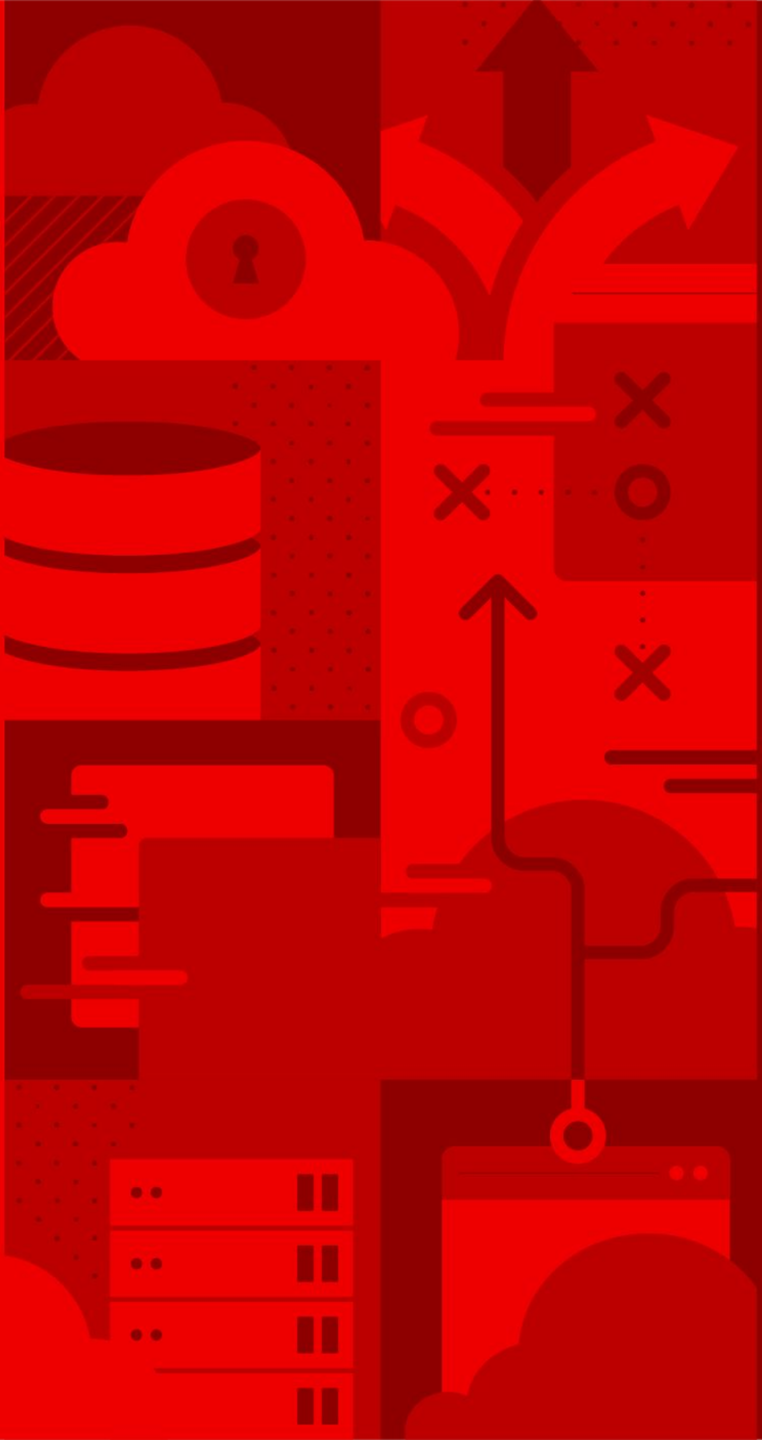
Redpoint

Amplify PARTNERS

Special

Kubernetes Certified Service Provider

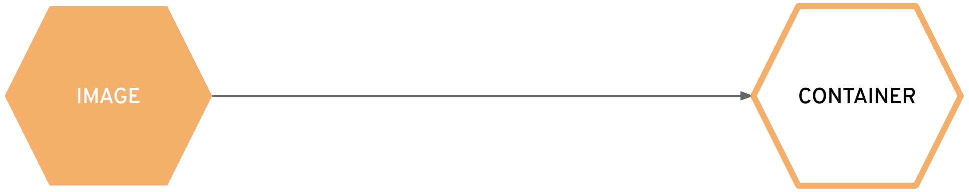
Kubernetes Training Partner



OpenShift and Kubernetes core concepts

a container is the smallest compute unit

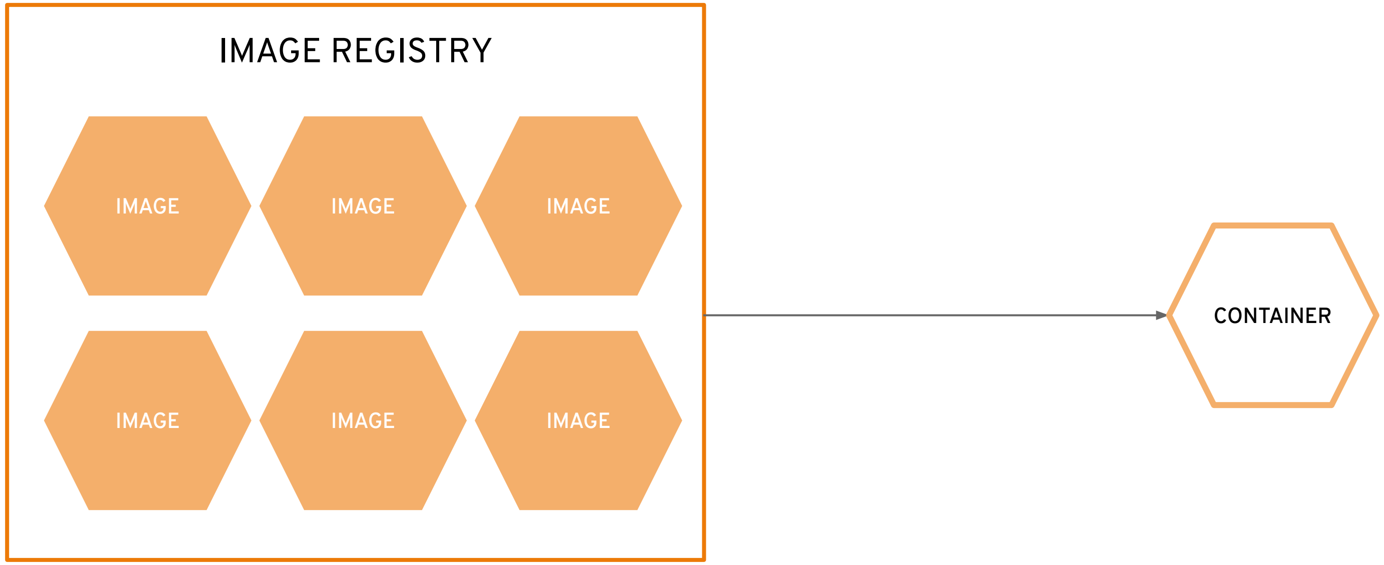
containers are created from container images



BINARY

RUNTIME

container images are stored in an image registry



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

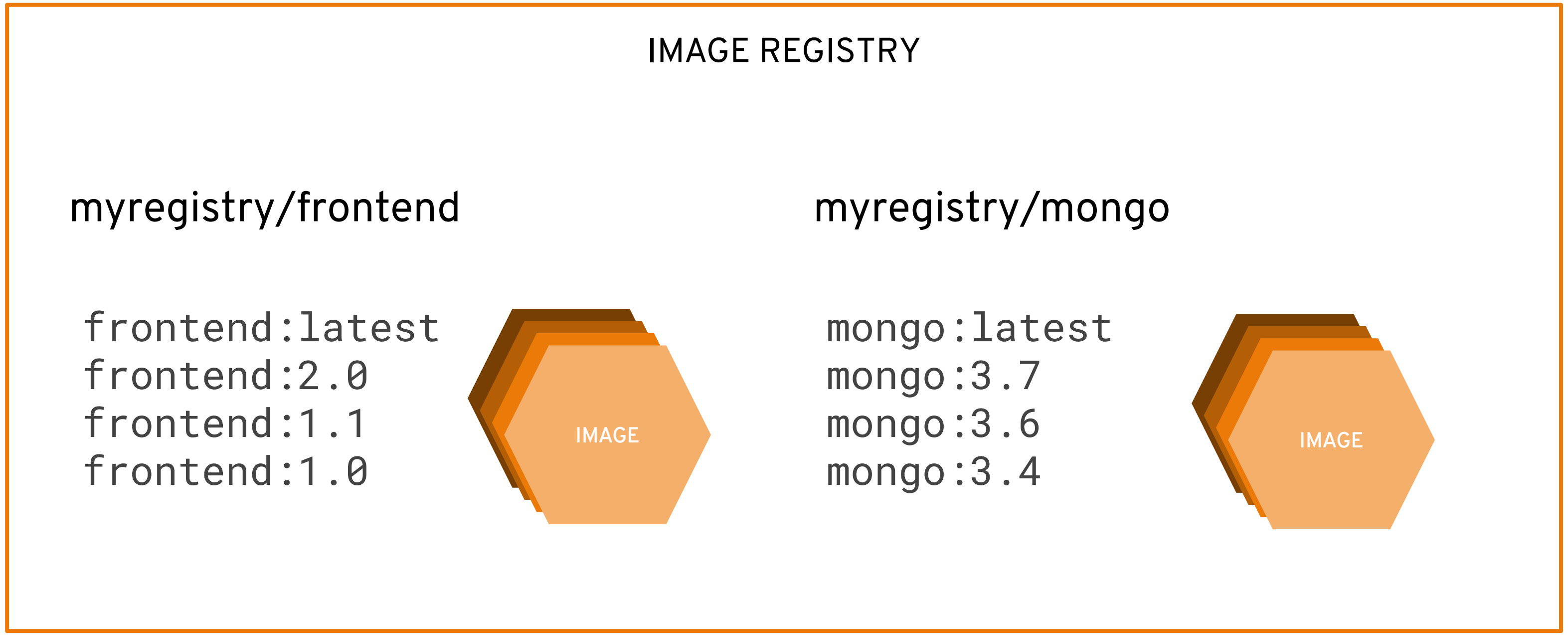


IMAGE REGISTRY

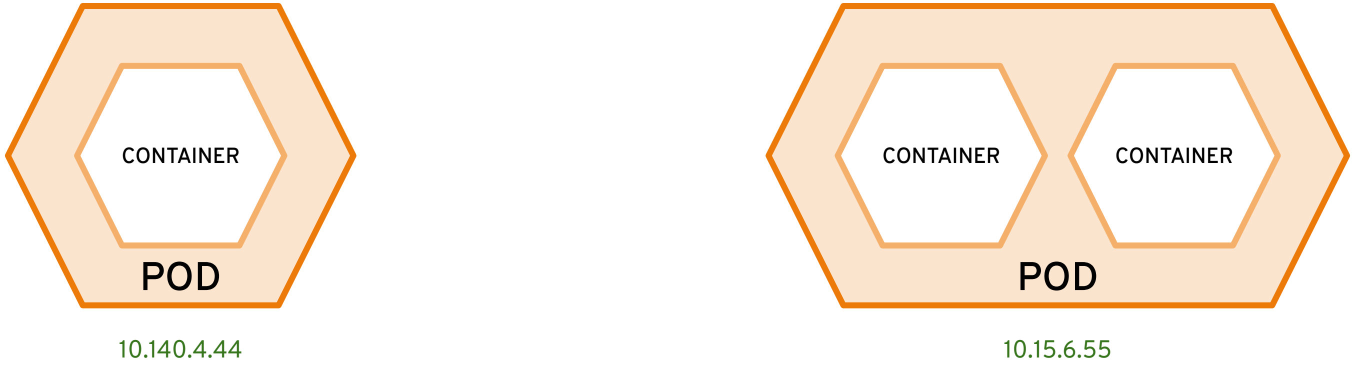
myregistry/frontend

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

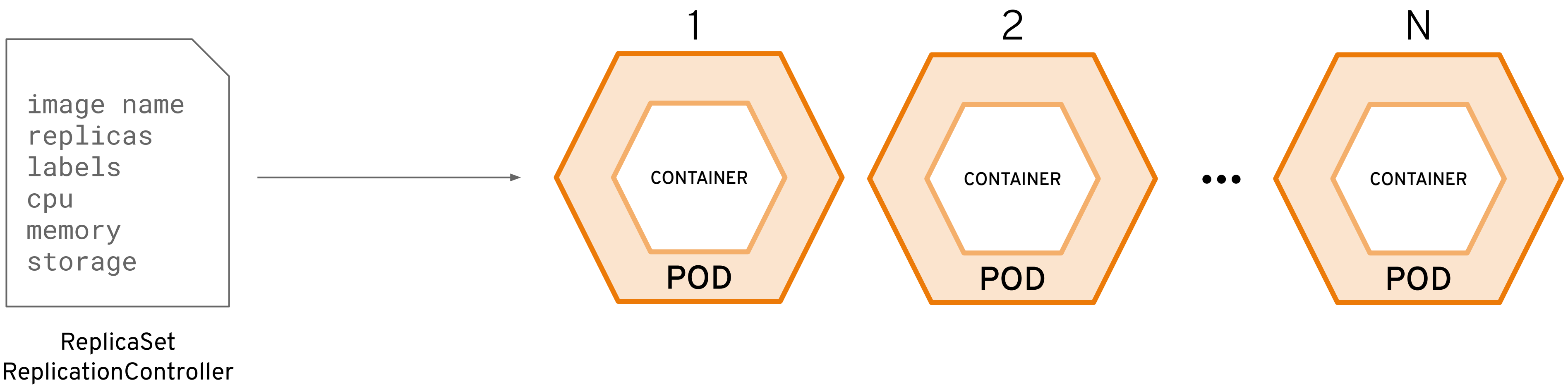
myregistry/mongo

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

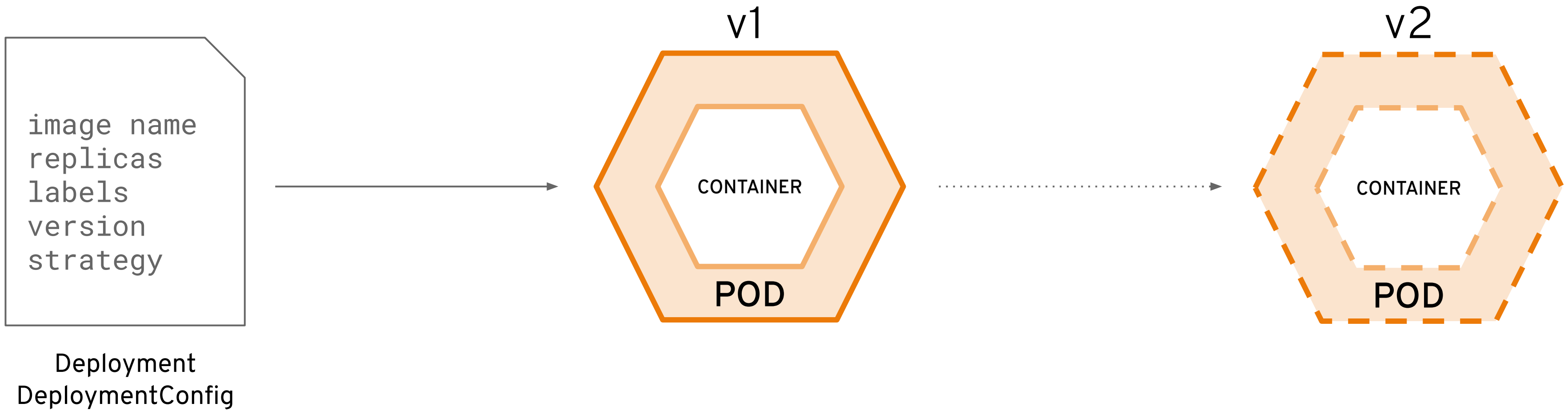
containers are wrapped in pods which are units of deployment and management



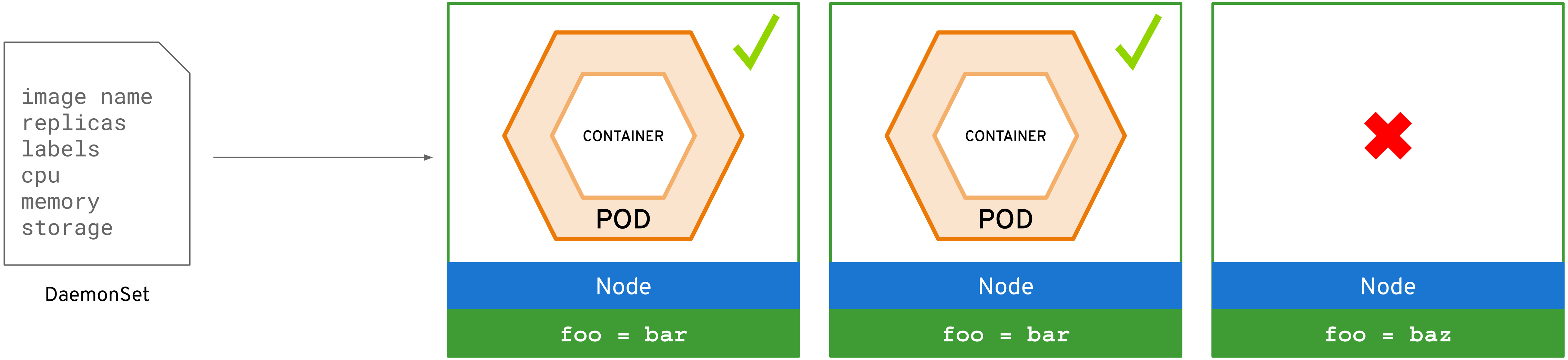
`ReplicationControllers` & `ReplicaSets` ensure a specified number of pods are running at any given time



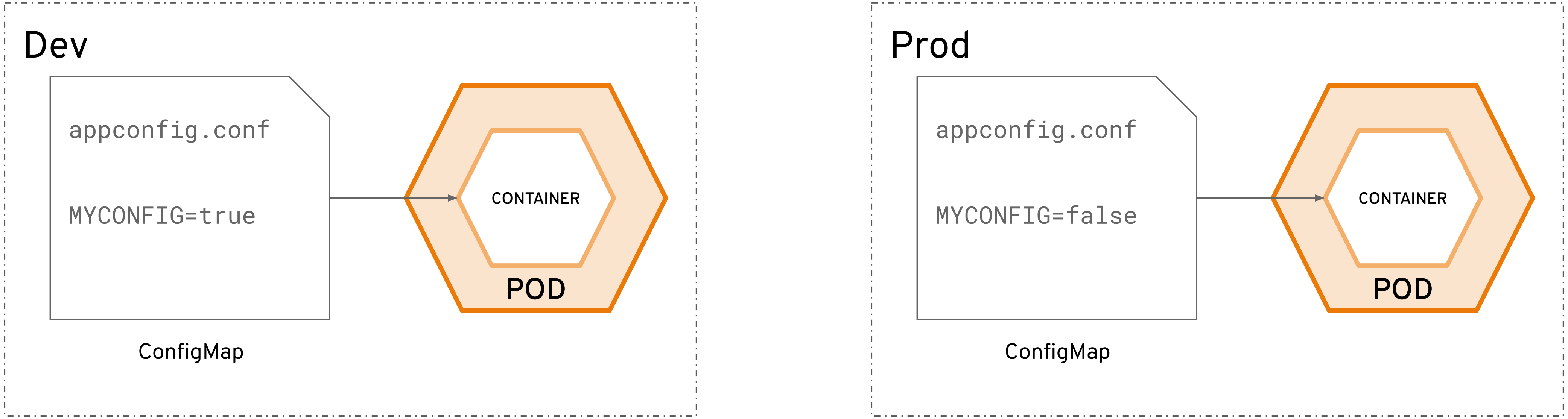
`Deployments` and `DeploymentConfigurations` define how to roll out new versions of Pods



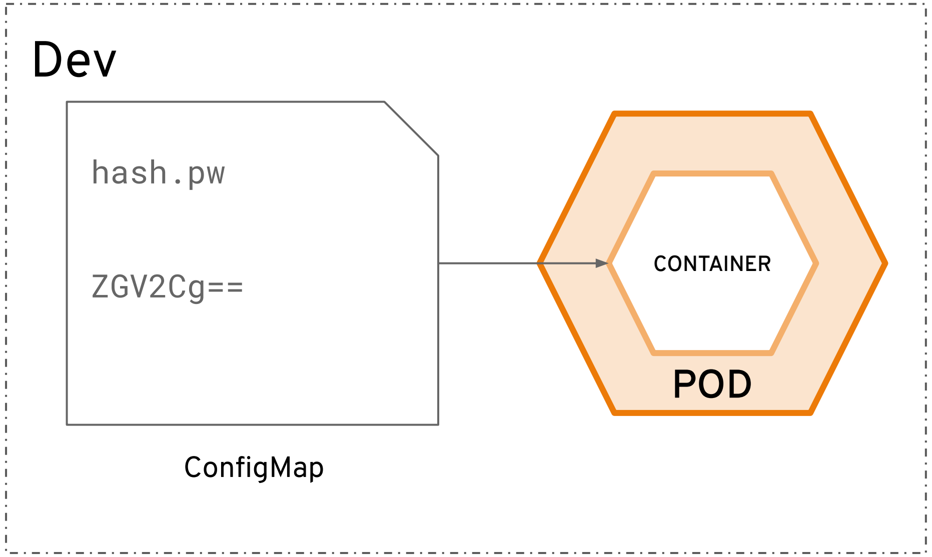
a `daemonset` ensures that all (or some) nodes run a copy of a pod

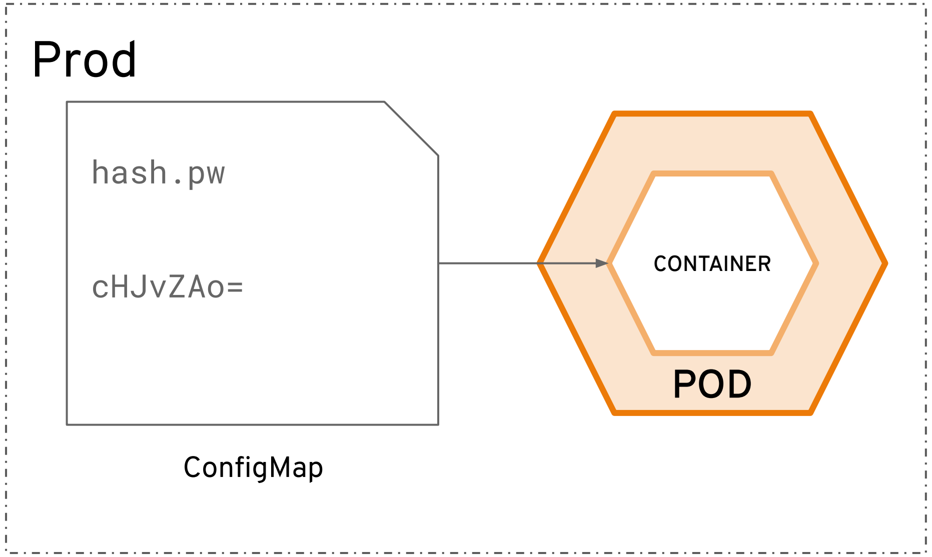


`configmaps` allow you to decouple configuration artifacts from image content

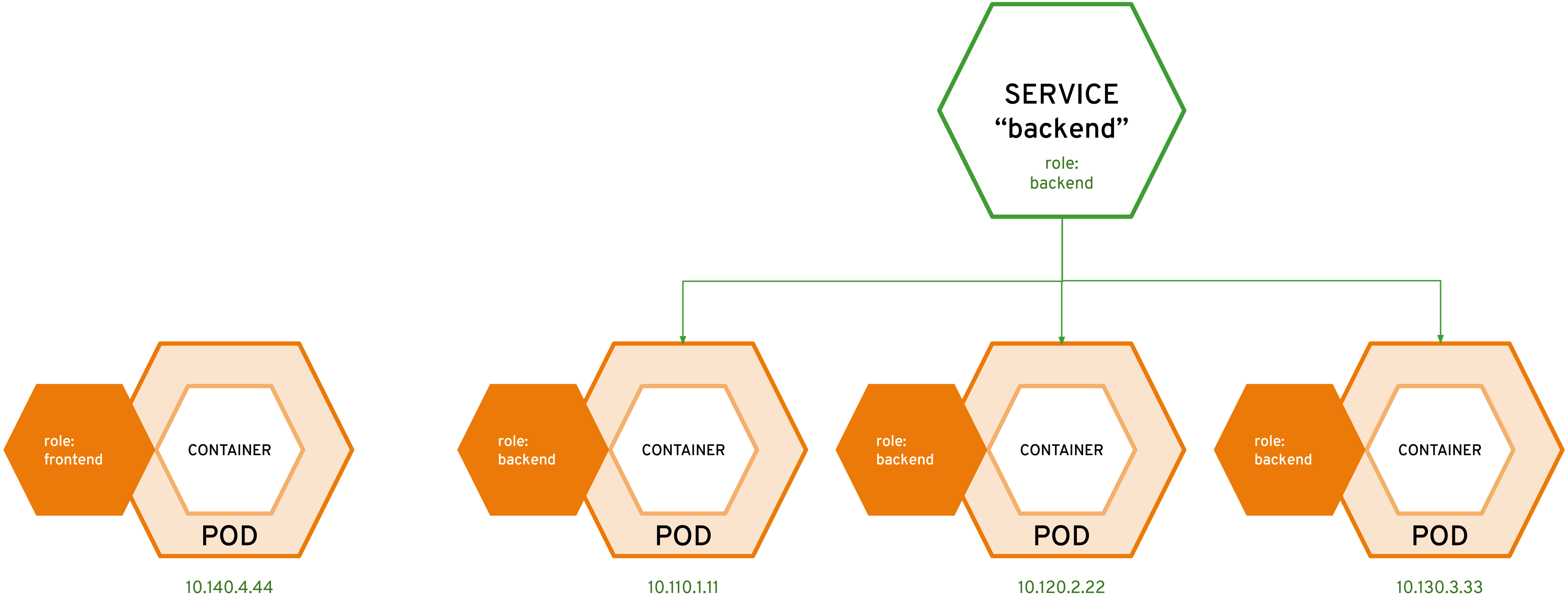


`secrets` provide a mechanism to hold sensitive information such as passwords

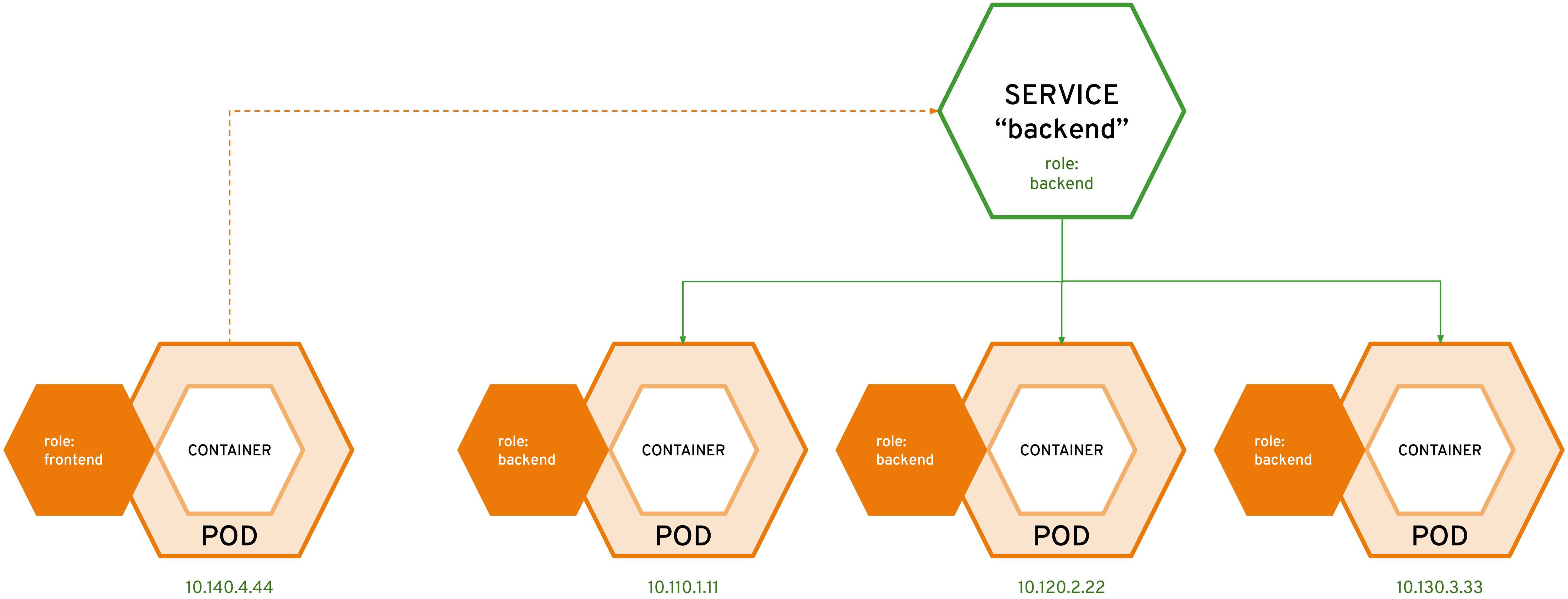




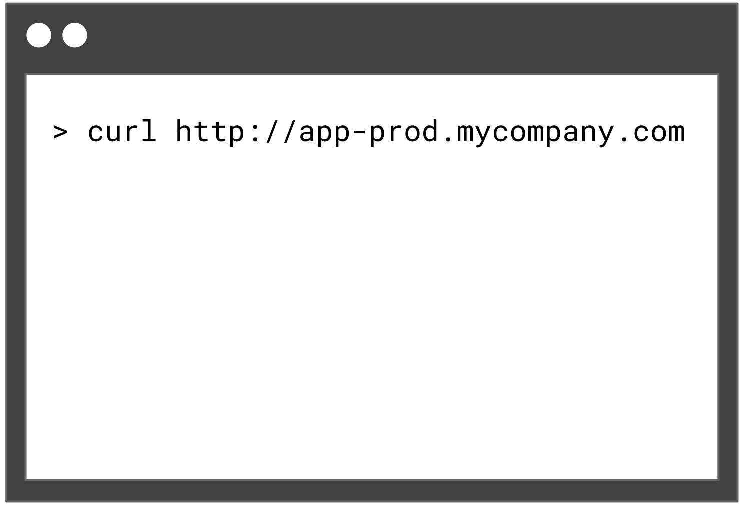
services provide internal load-balancing and service discovery across pods



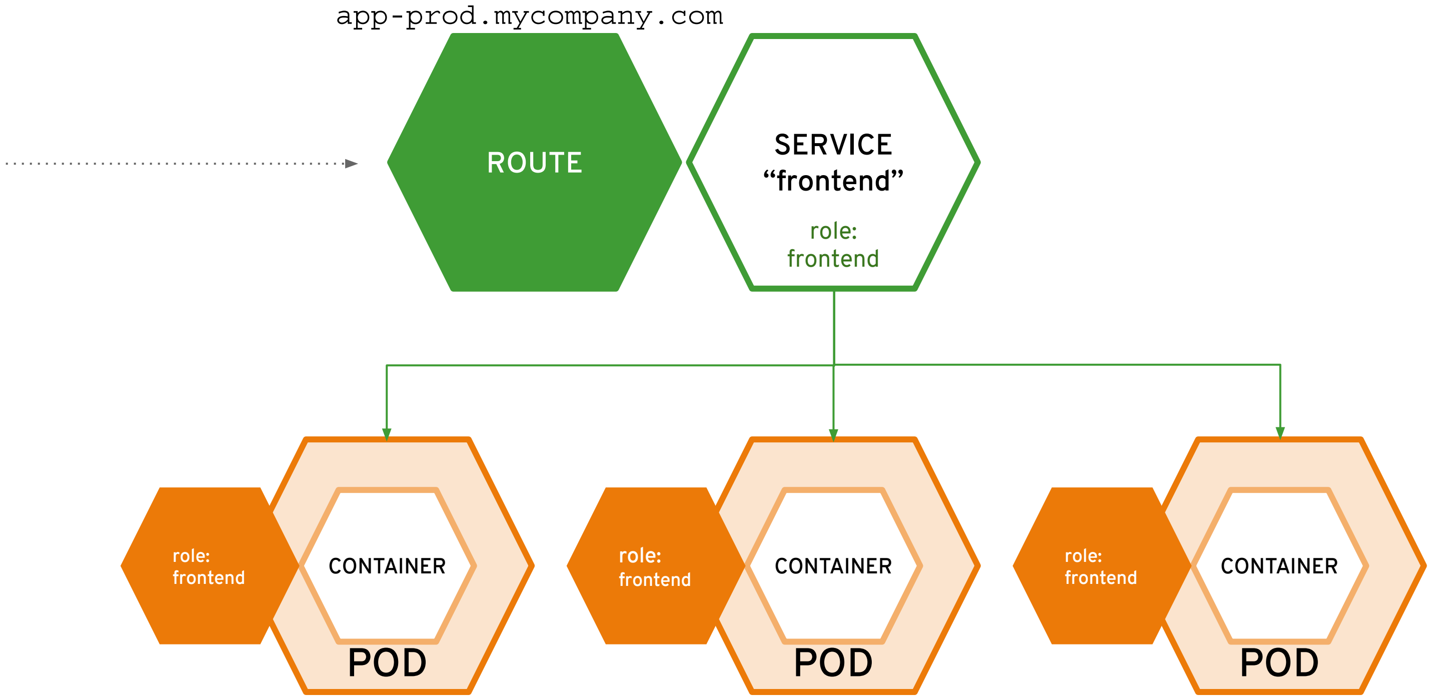
apps can talk to each other via services



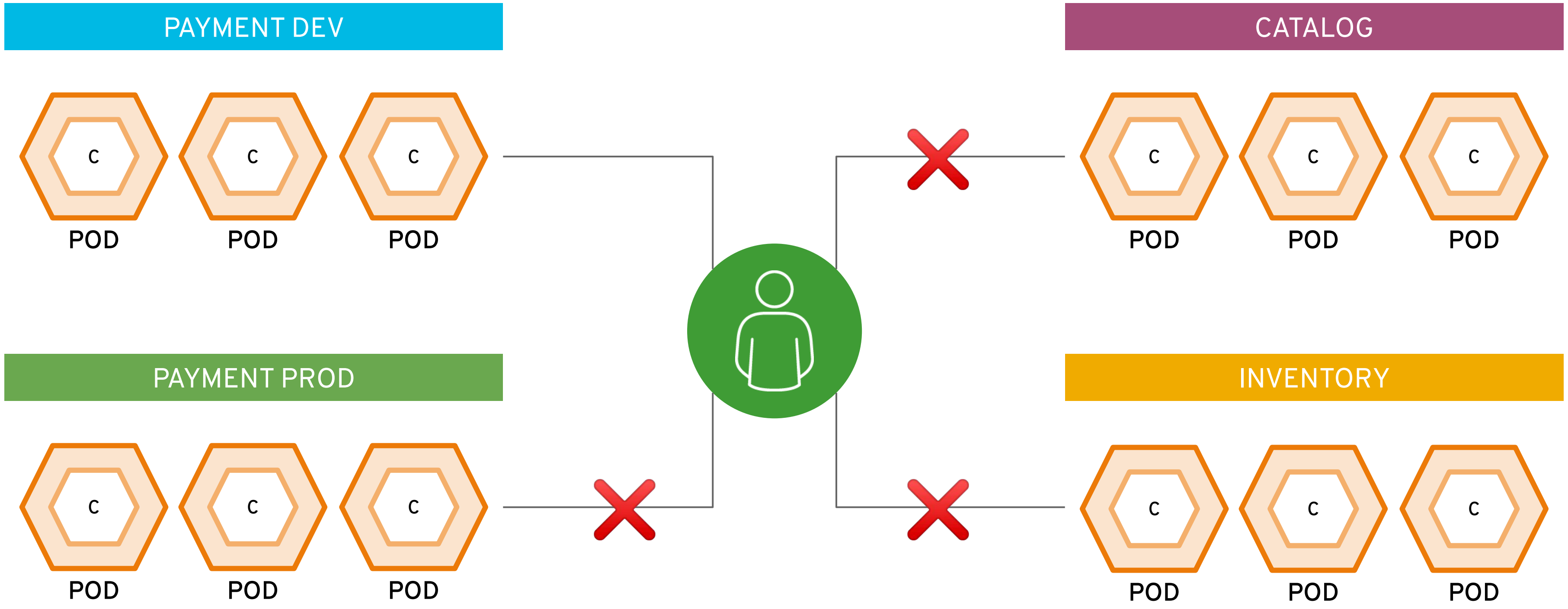
routes make services accessible to clients outside the environment via real-world urls

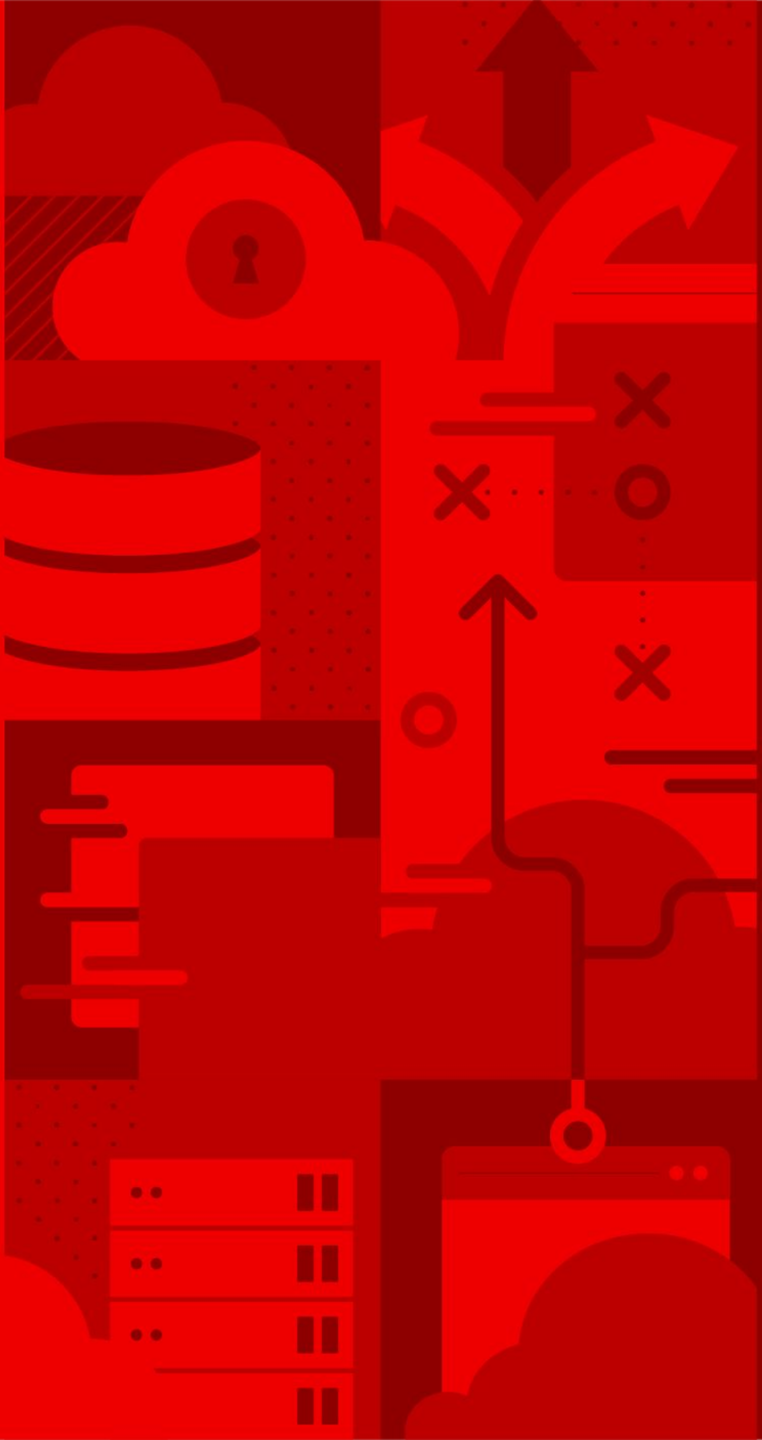


```
> curl http://app-prod.mycompany.com
```



projects isolate apps across environments, teams, groups and departments





OpenShift 4 Architecture

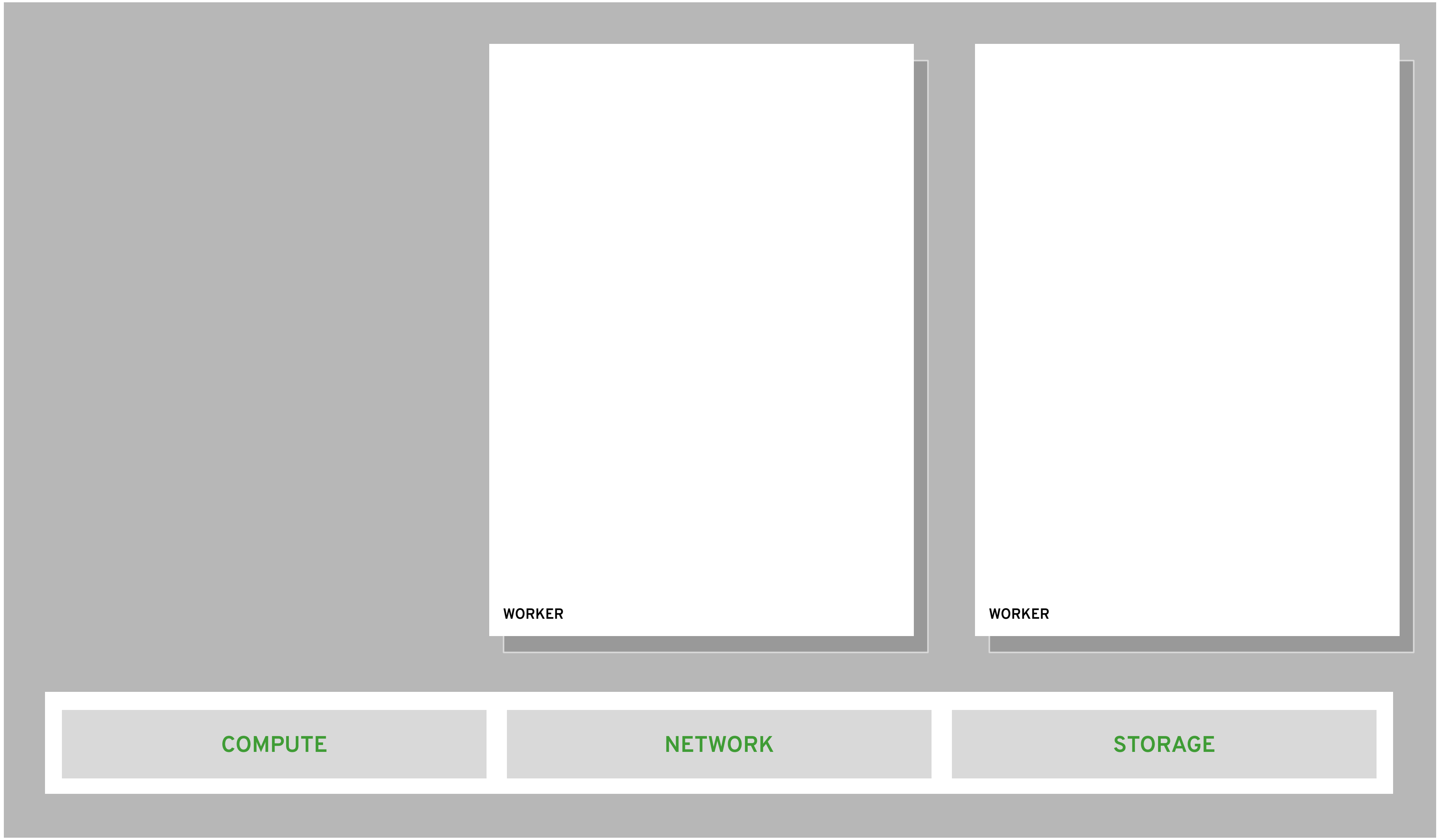
your choice of infrastructure

COMPUTE

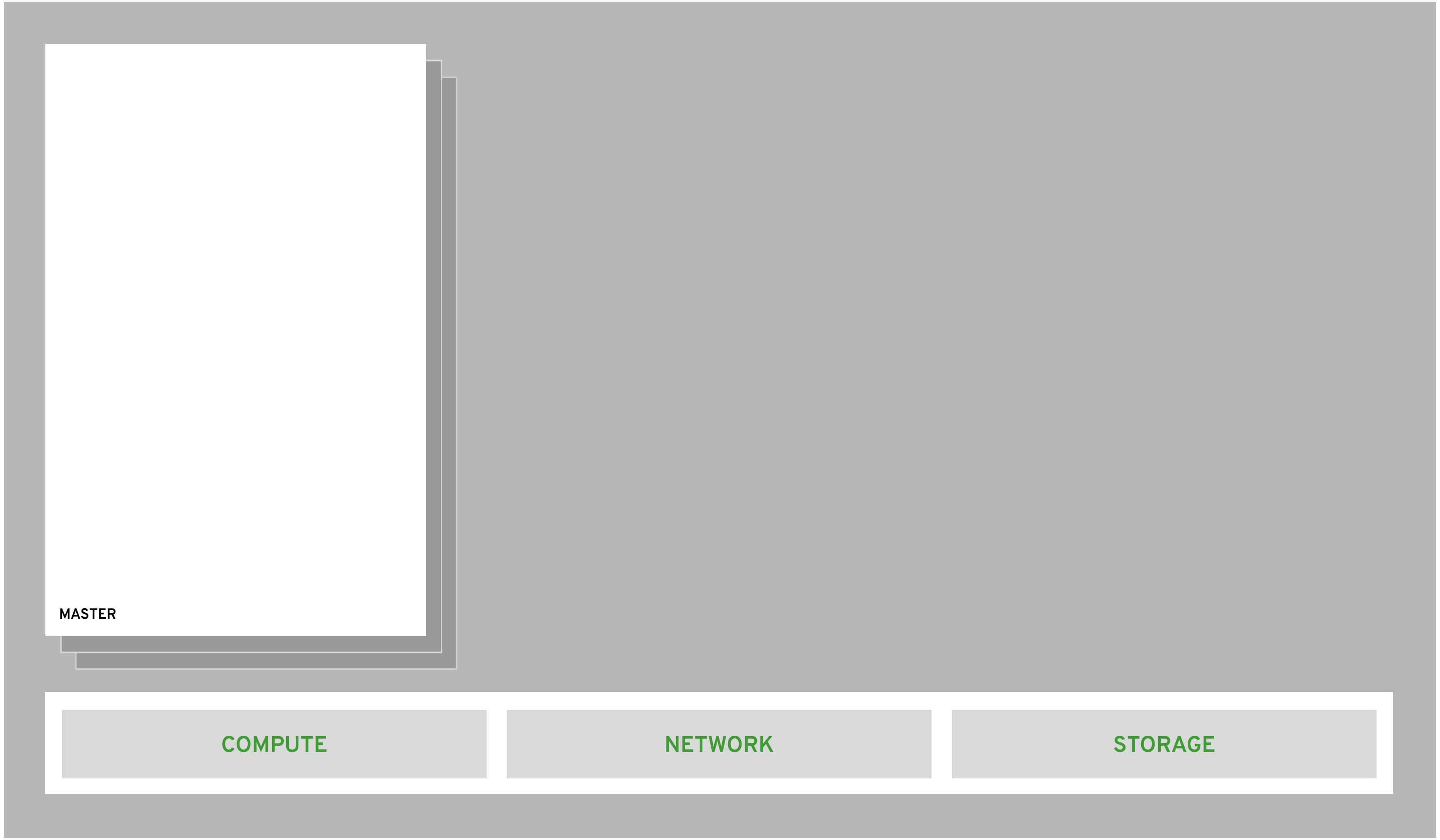
NETWORK

STORAGE

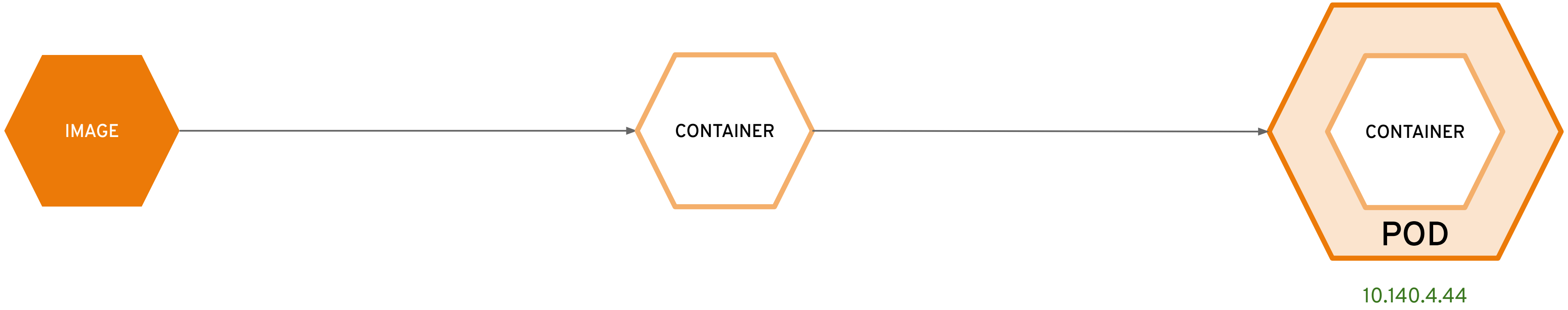
workers run workloads



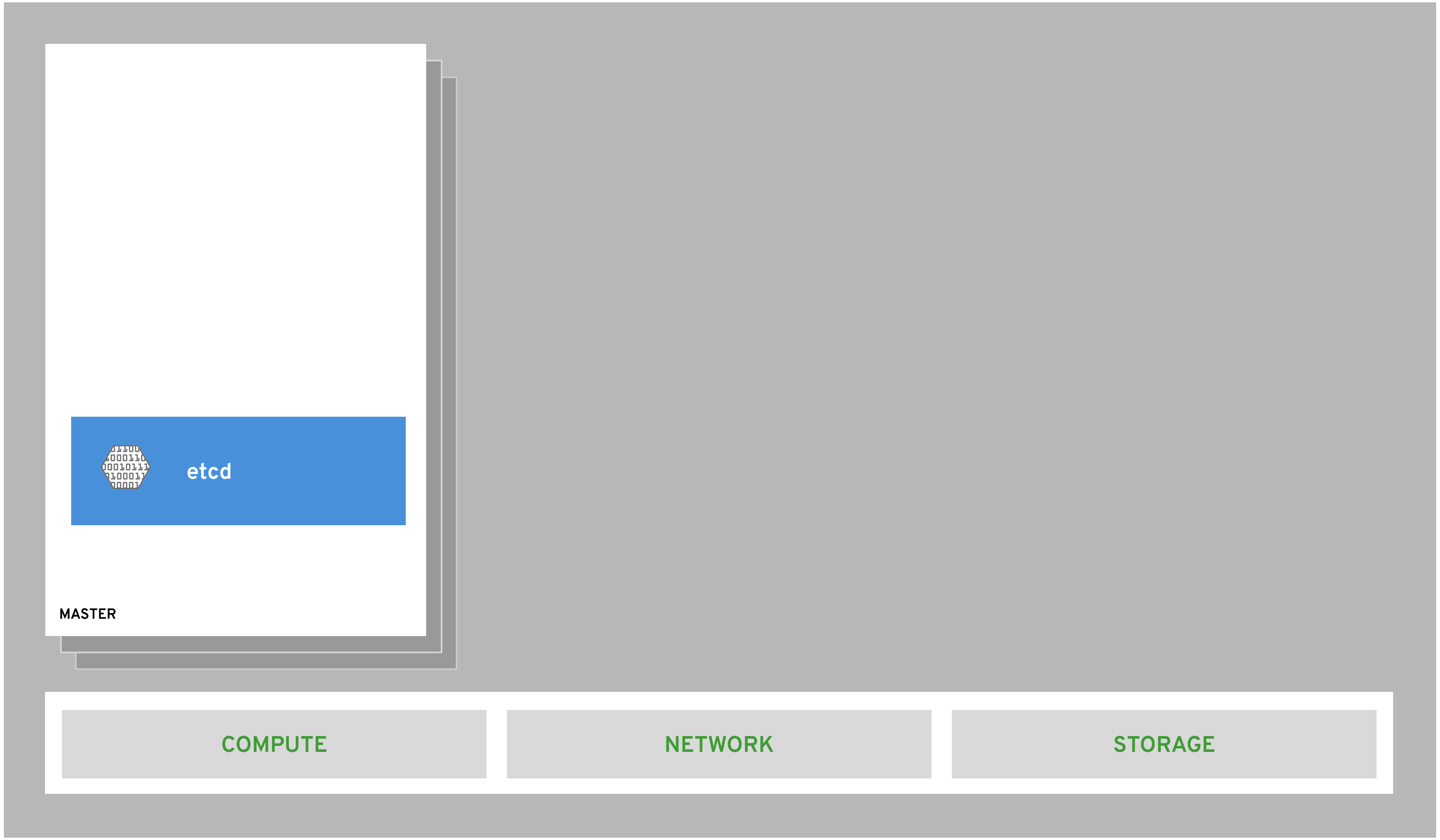
masters are the control plane



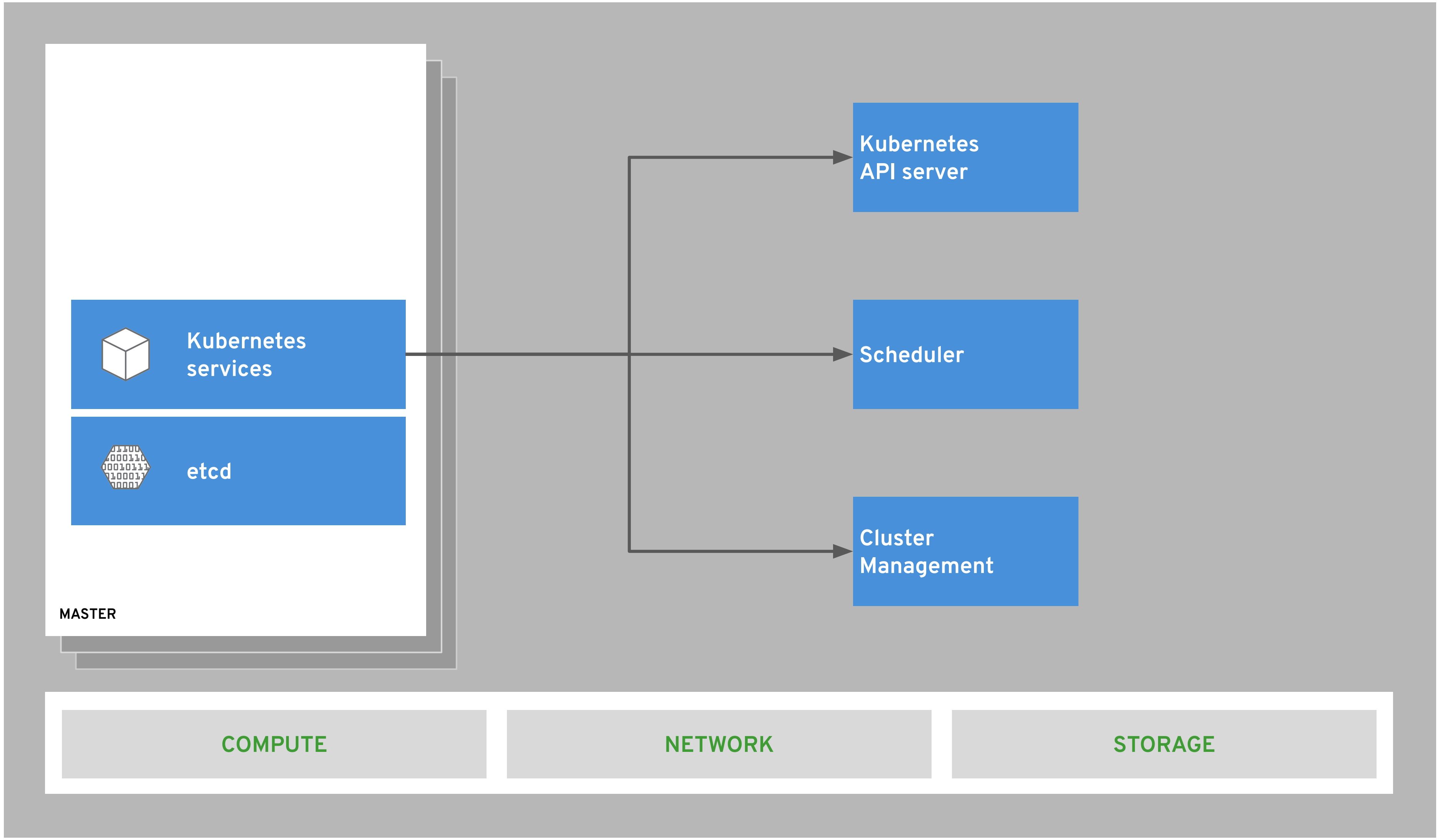
everything runs in pods



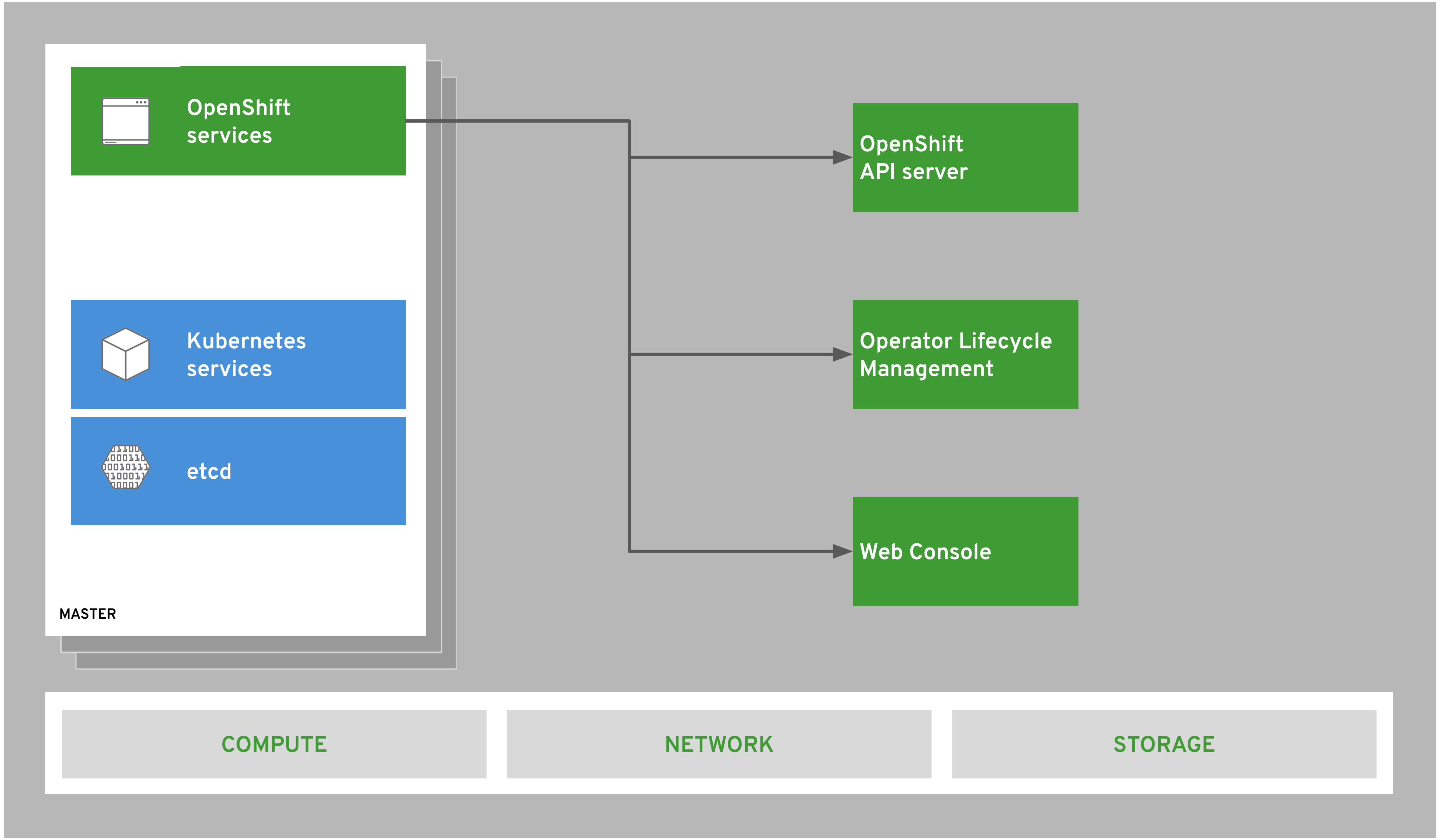
state of everything



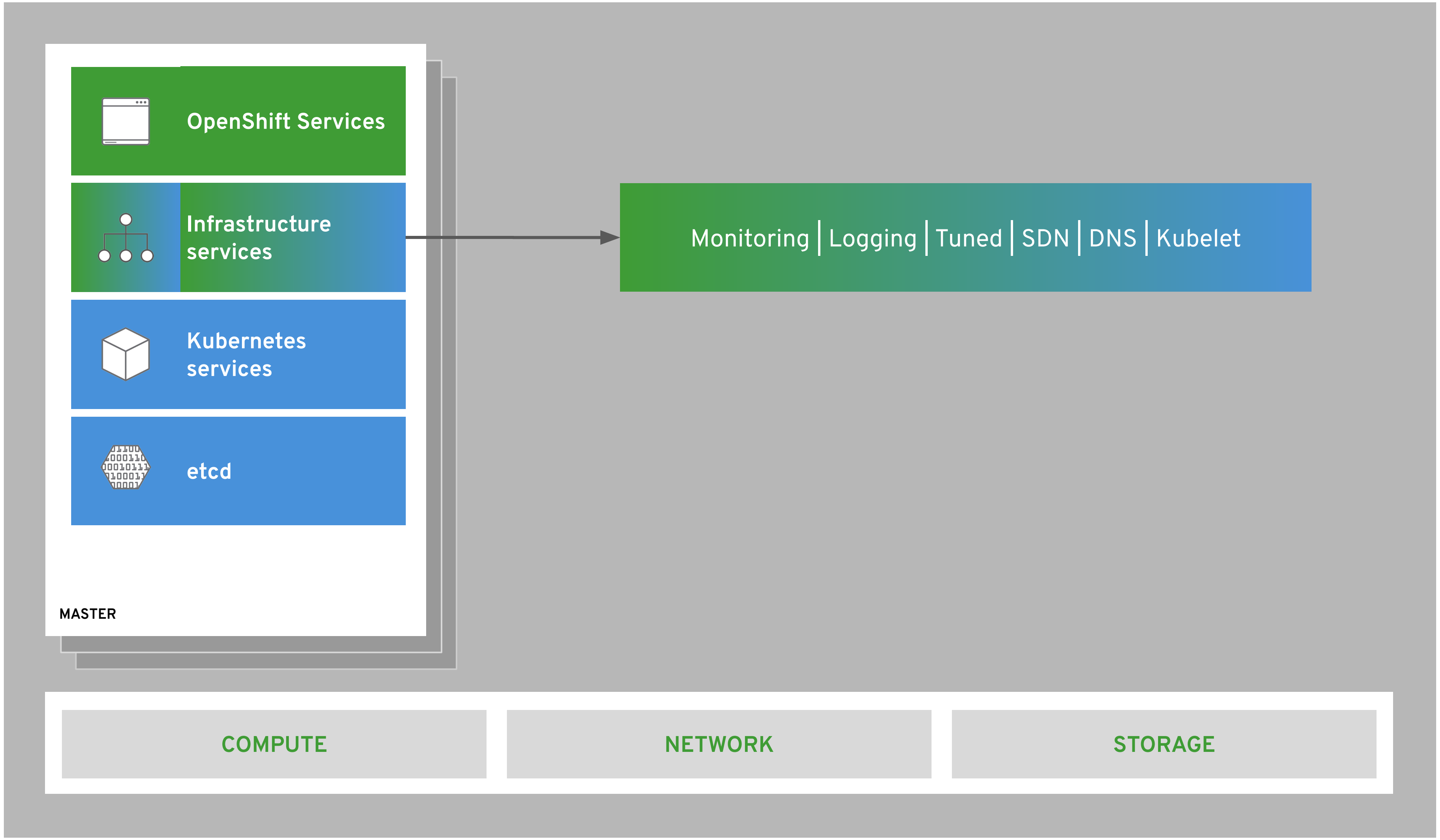
core kubernetes components



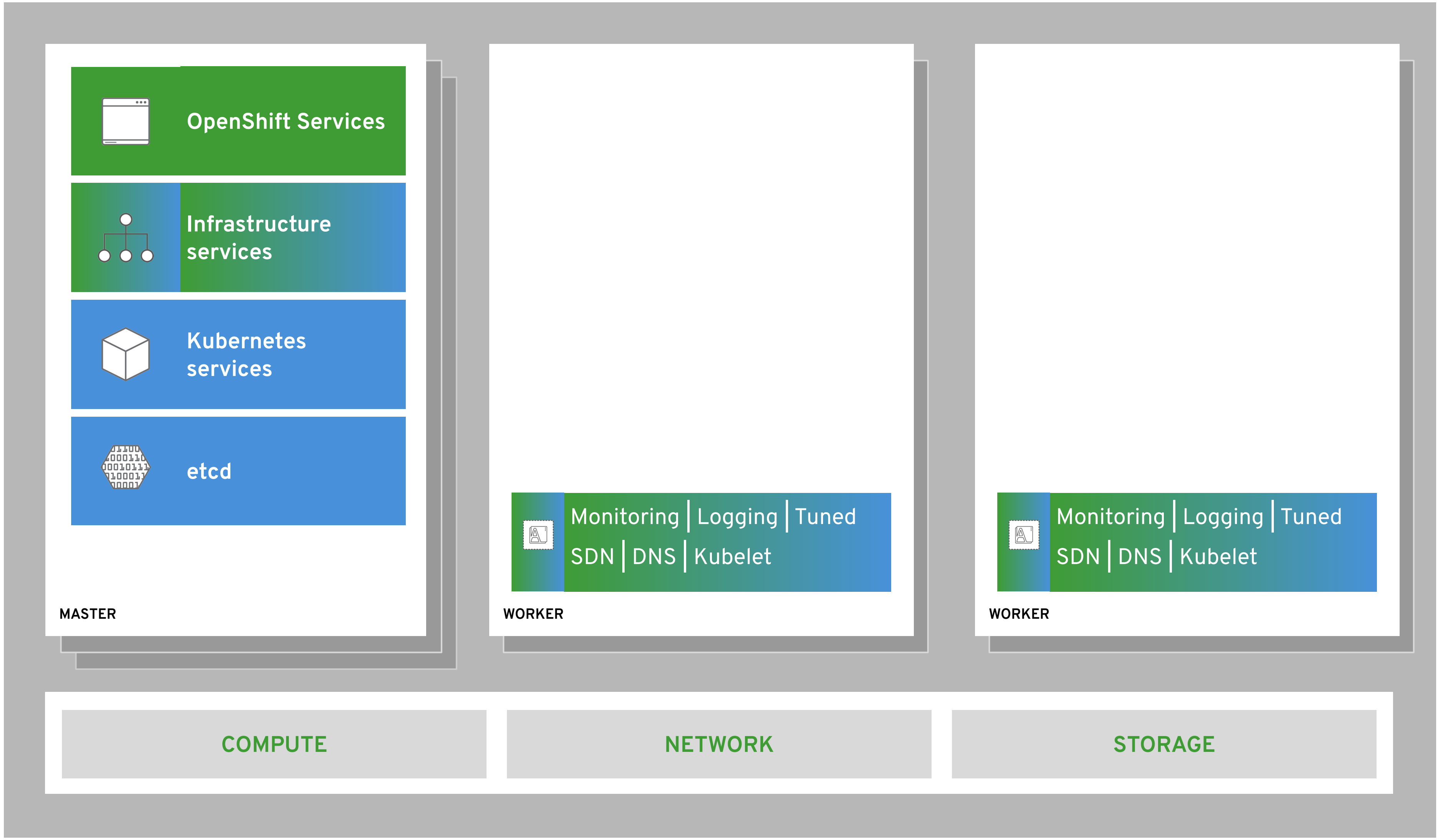
core OpenShift components



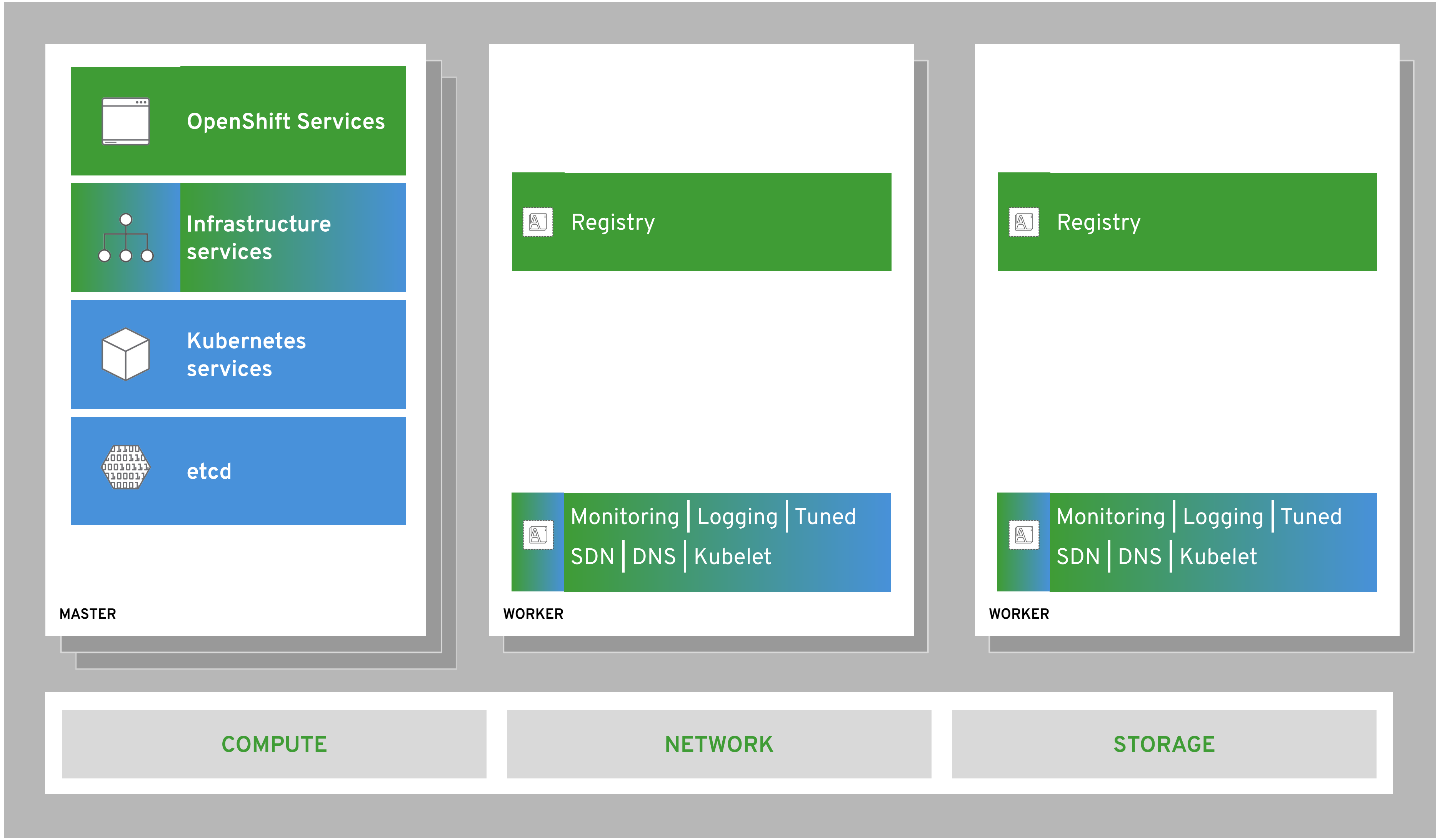
internal and support infrastructure services



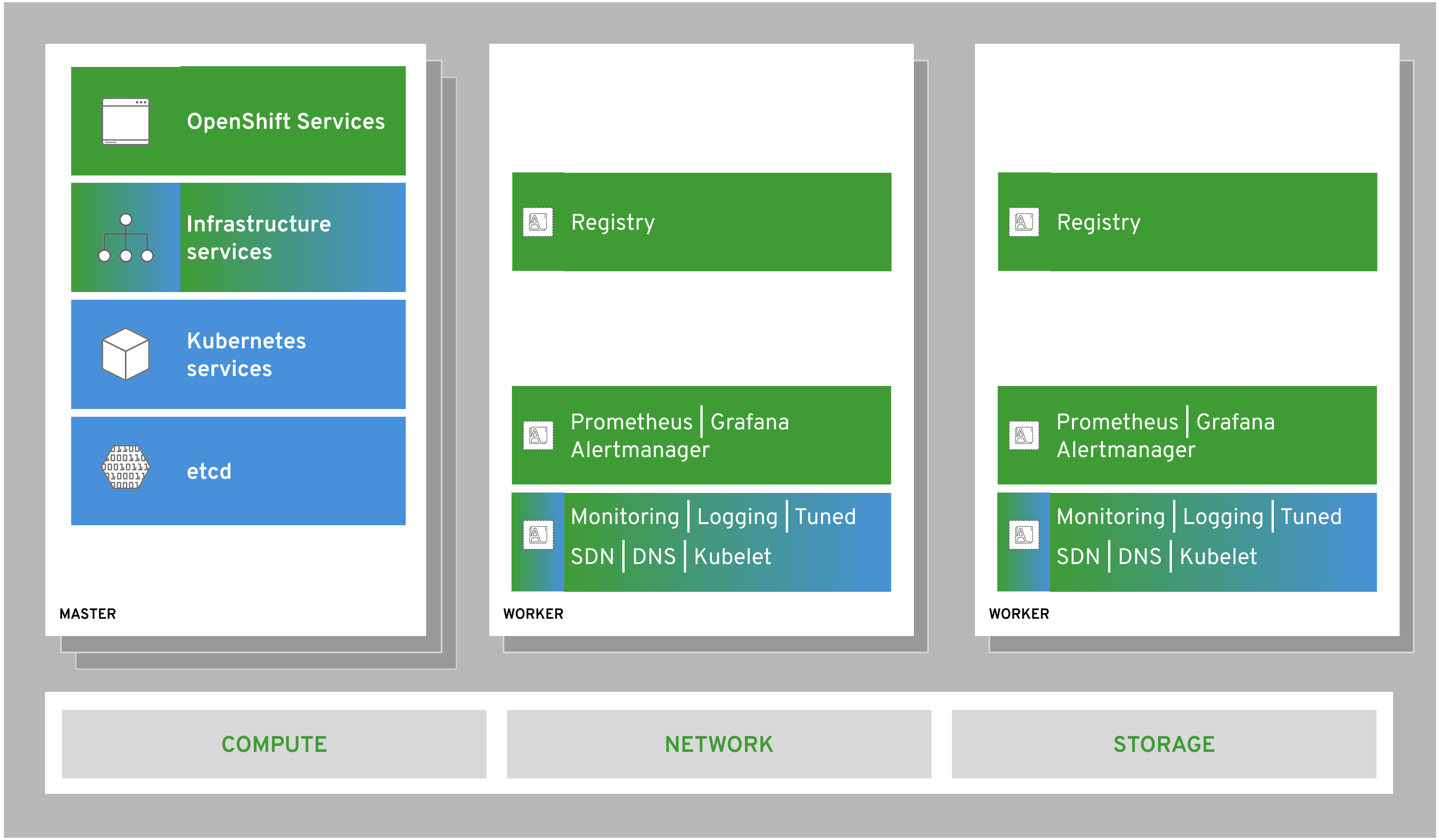
run on all hosts



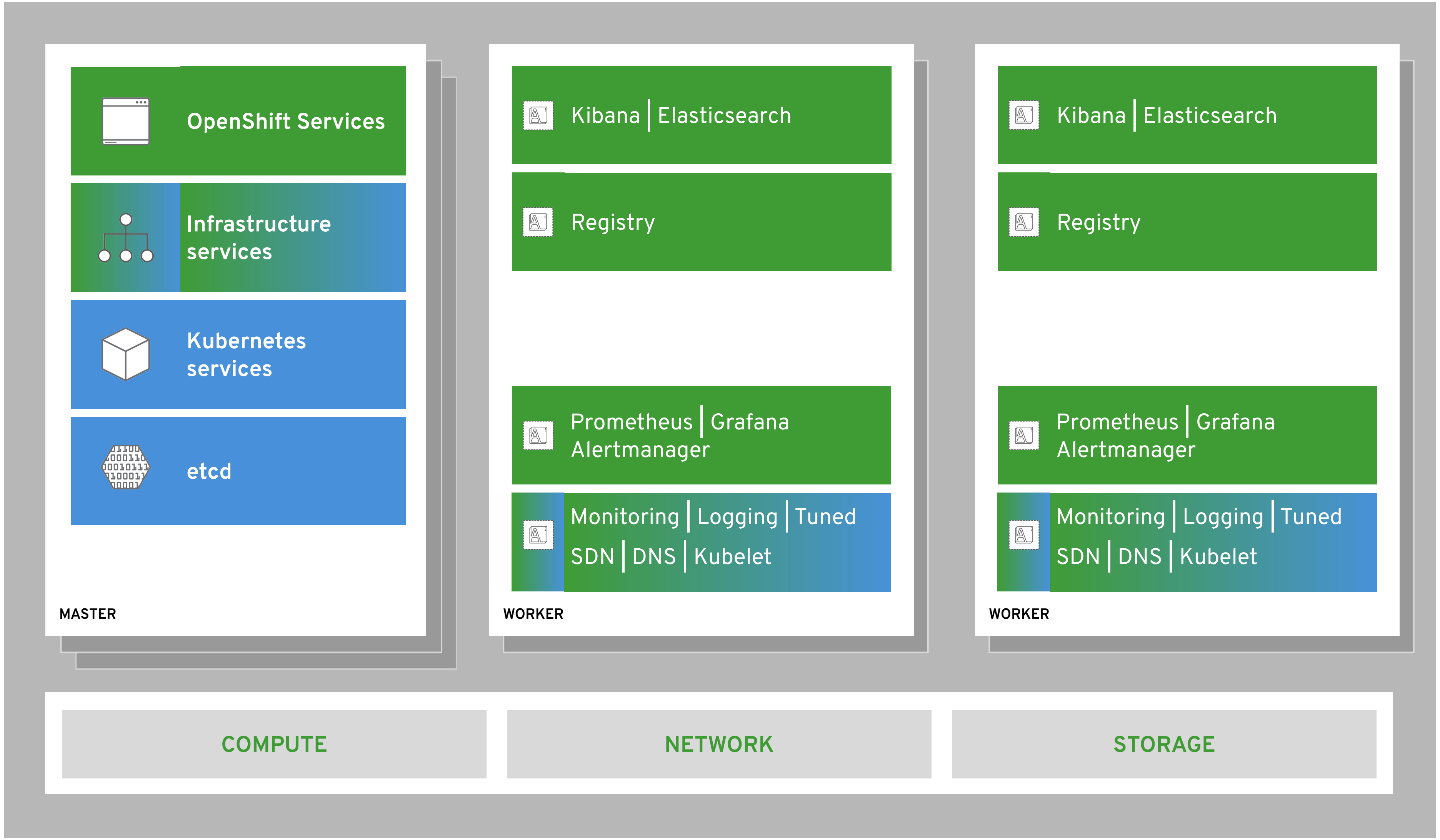
integrated image registry



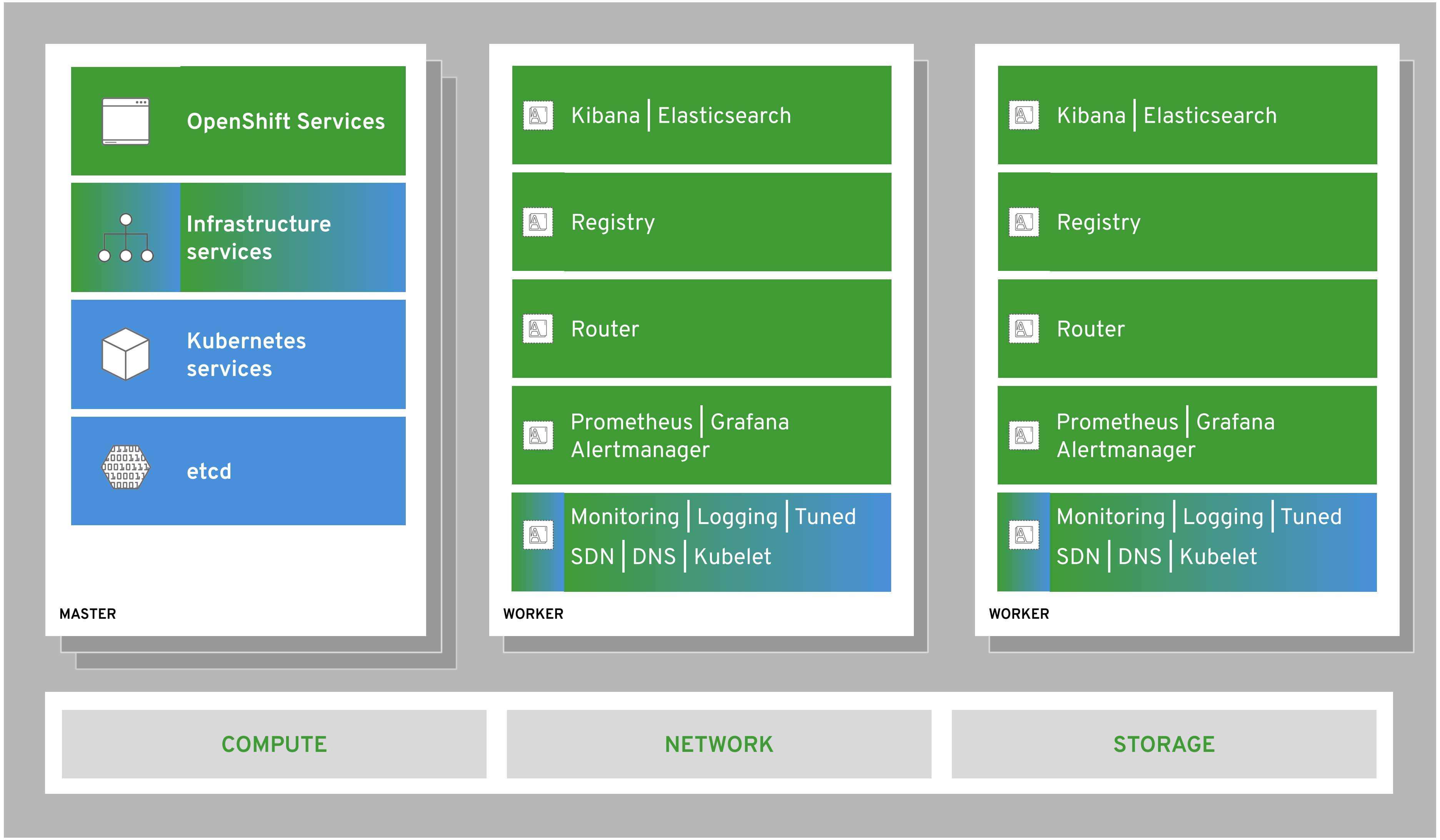
cluster monitoring



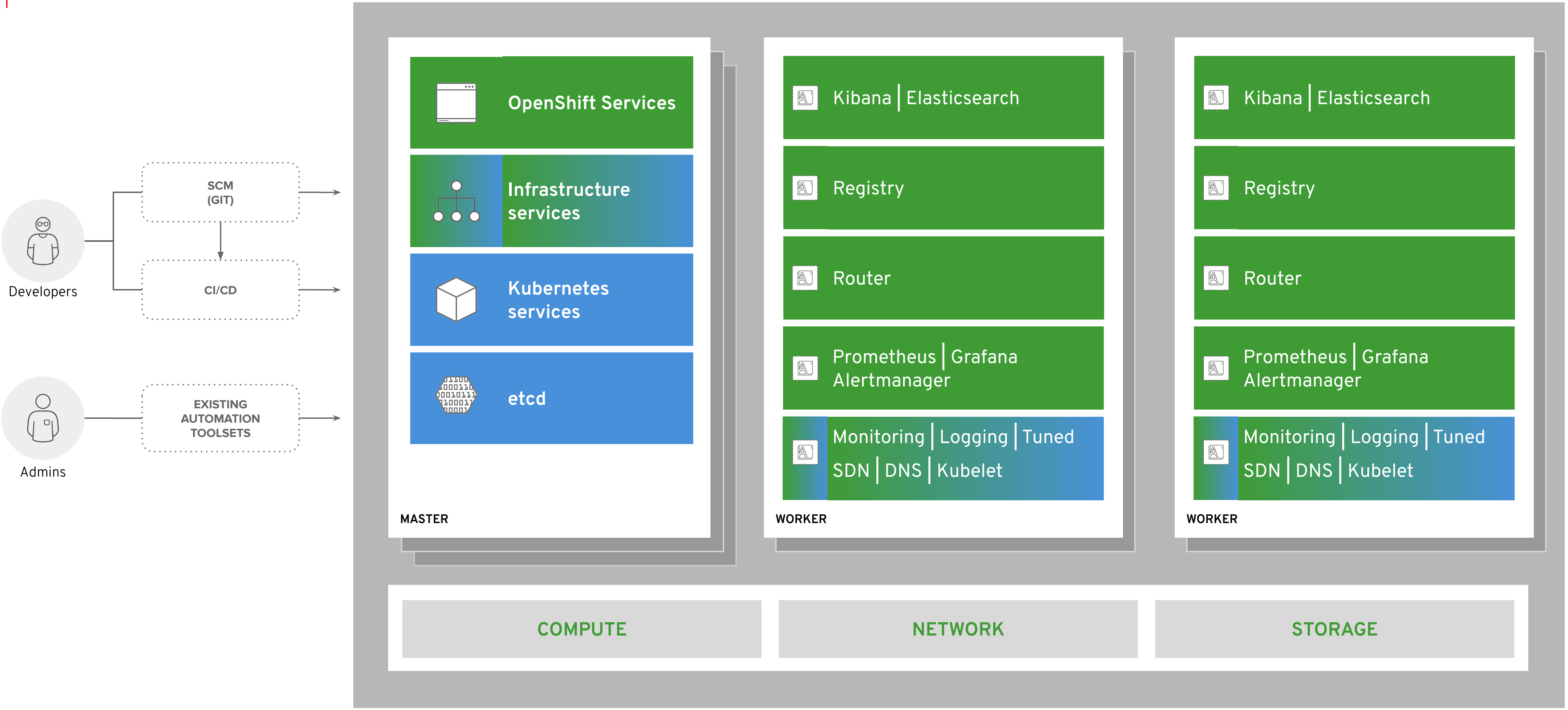
log aggregation

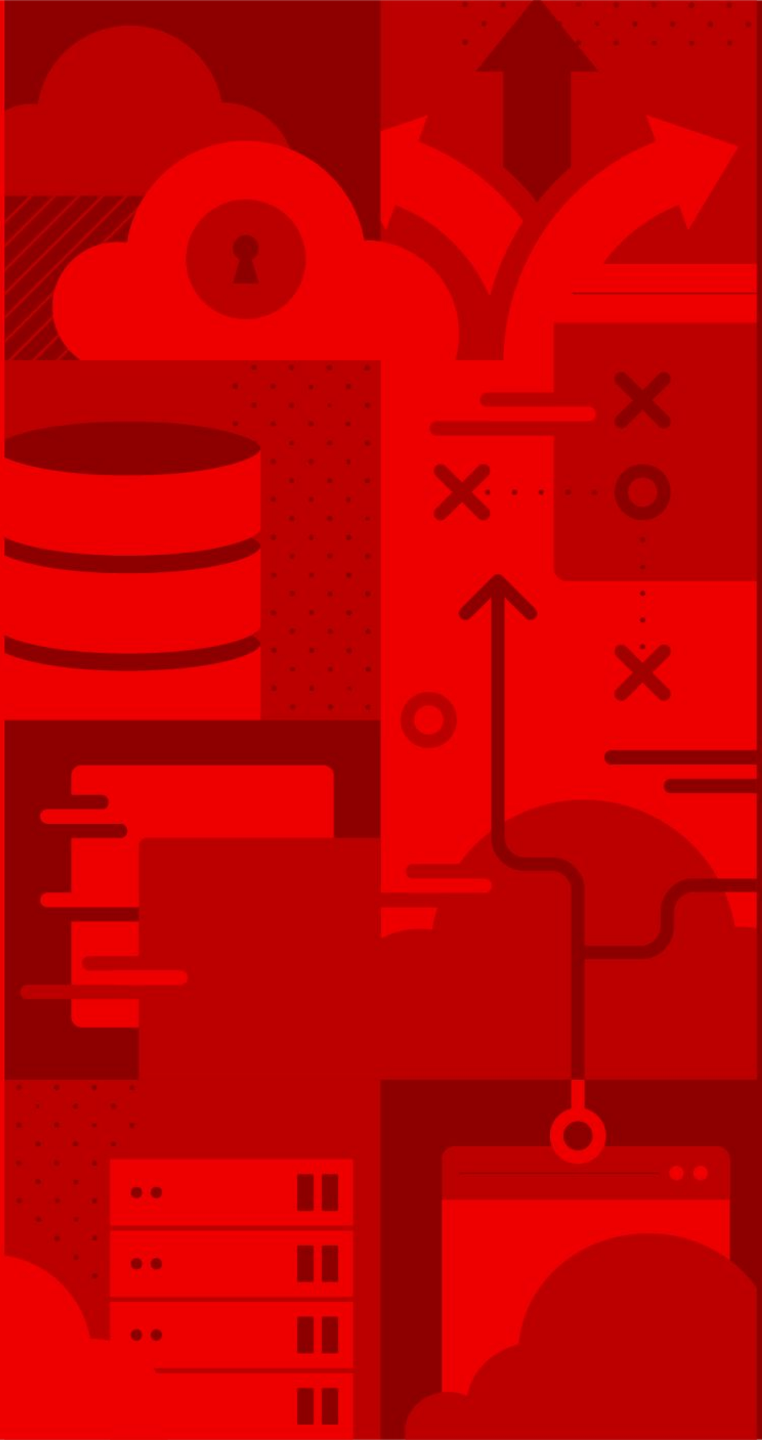


integrated routing



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





OpenShift lifecycle, installation & upgrades

OpenShift 4 Installation

Two new paradigms for

deploying clusters

Installation Paradigms

OPENSHIFT CONTAINER PLATFORM

Full Stack Automated

Simplified opinionated "Best Practices" for cluster provisioning

Fully automated installation and updates including host container OS.

Red Hat Enterprise Linux CoreOS

Pre-existing Infrastructure

Customer managed resources & infrastructure provisioning

Plug into existing DNS and security boundaries

Red Hat Enterprise Linux CoreOS

Red Hat Enterprise Linux

HOSTED OPENSHIFT

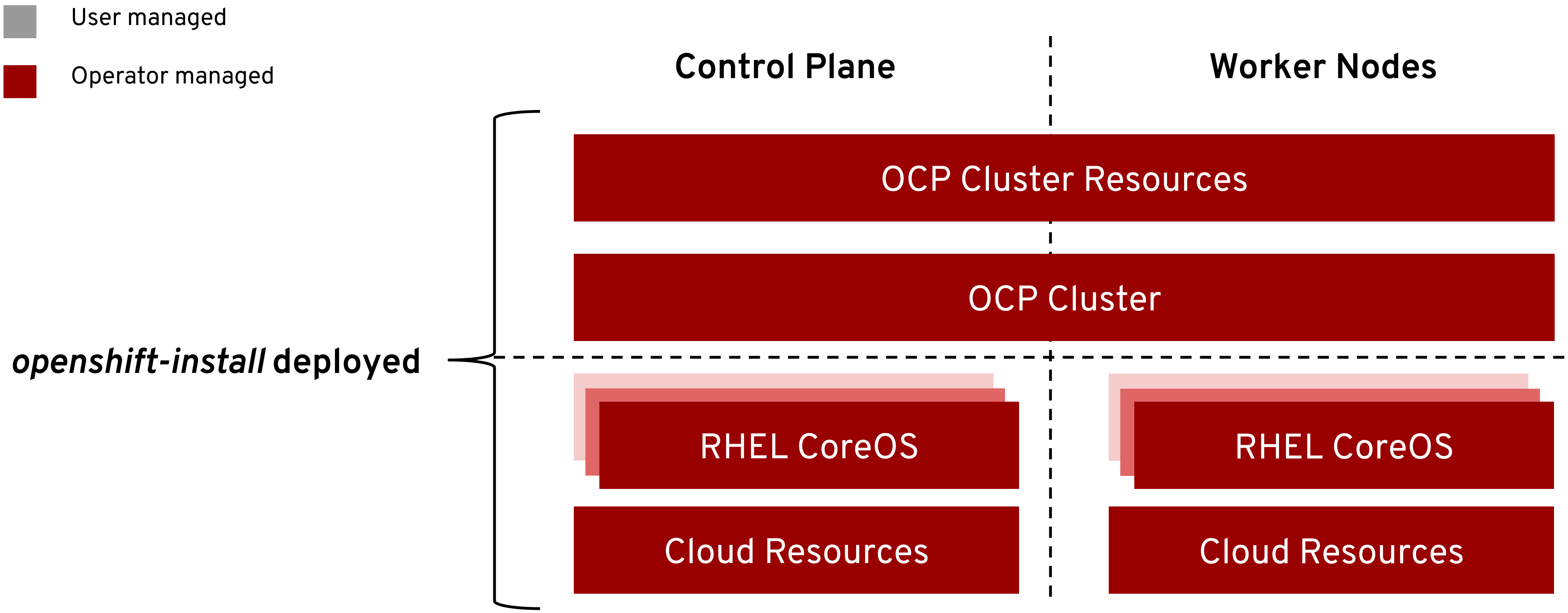
Azure Red Hat OpenShift

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

openshift-install deployed

Control Plane

Worker Nodes

OCP Cluster Resources

OCP Cluster

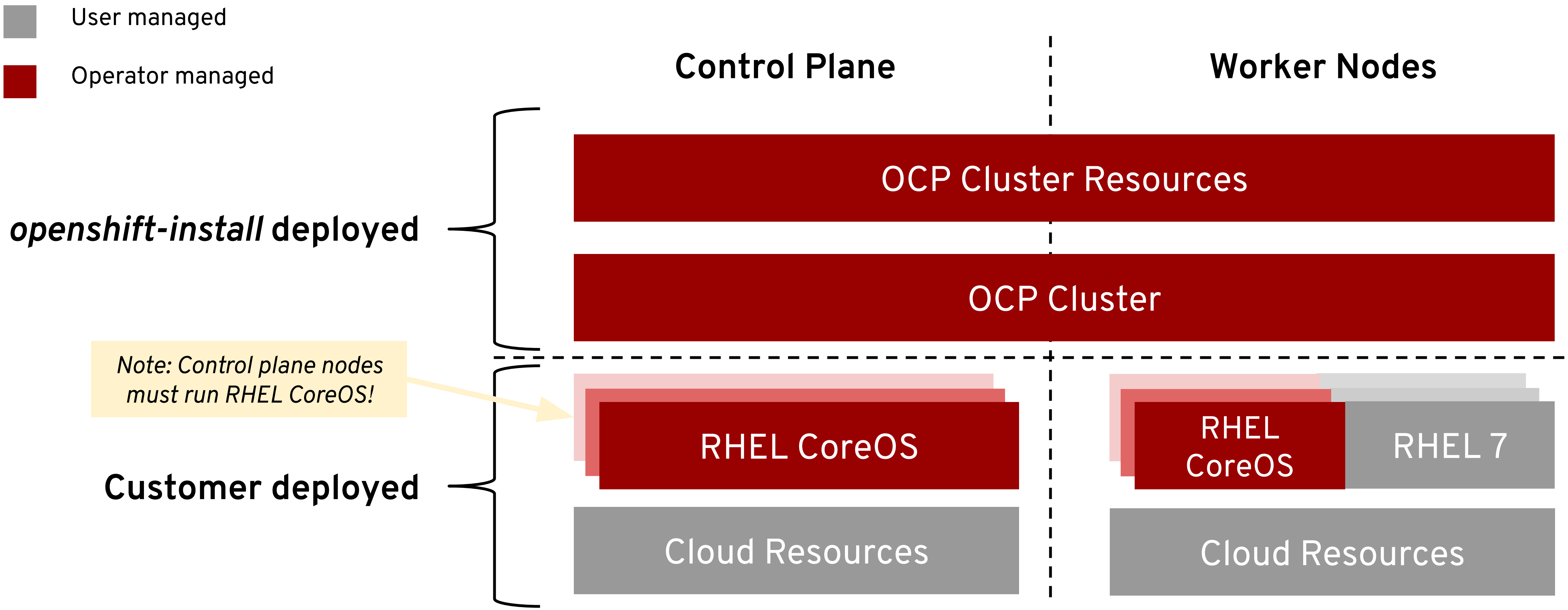
RHEL CoreOS

RHEL CoreOS

Cloud Resources

Cloud Resources

Pre-existing Infrastructure Installation



Note: Control plane nodes must run RHEL CoreOS!

Comparison of Paradigms

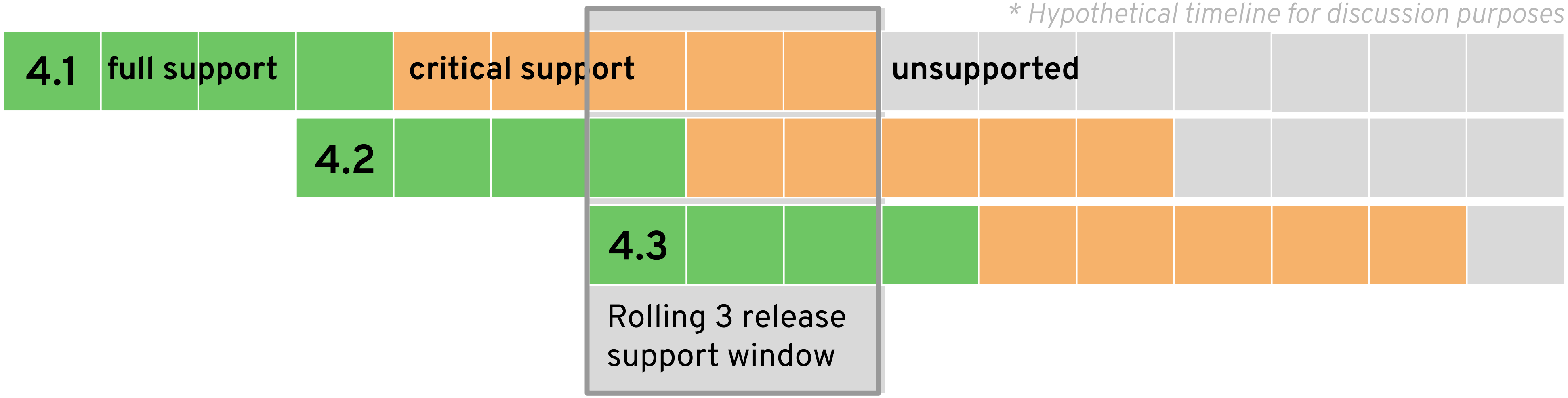
	Full Stack Automation	Pre-existing Infrastructure
Build Network	Installer	User
Setup Load Balancers	Installer	User
Configure DNS	Installer	User
Hardware/VM Provisioning	Installer	User
OS Installation	Installer	User
Generate Ignition Configs	Installer	Installer
OS Support	Installer: RHEL CoreOS	User: 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

Support Timelines



* Hypothetical timeline for discussion purposes

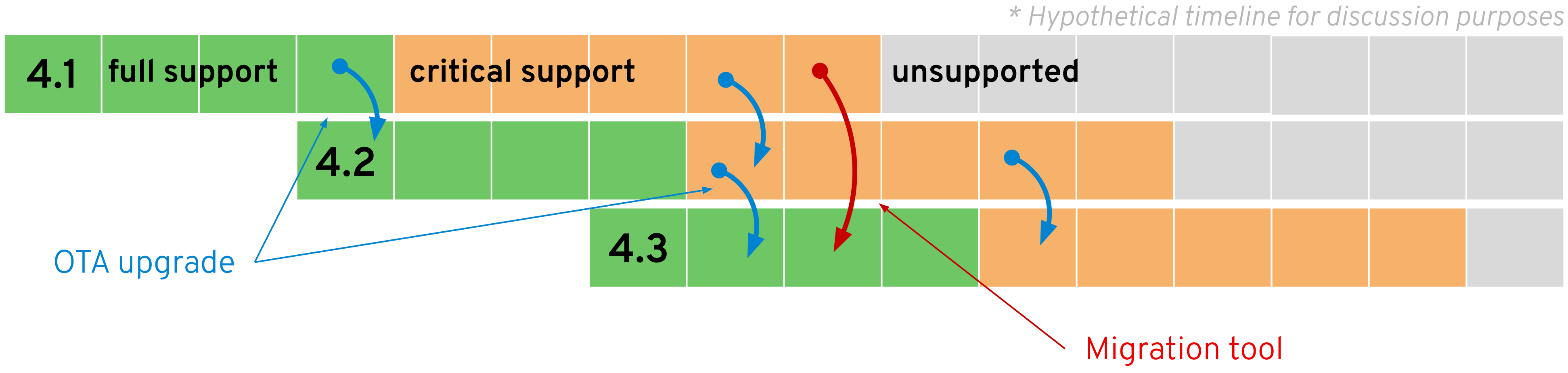
New model
Release based, not date based. Rolling three release window for support.

The overall 4 series will be supported for at least three years

- Minimum two years full support (likely more)
- One year maintenance past the end of full support

EUS release planned
Supported for 14 months of critical bug and critical security fixes instead of the normal 5 months. If you stay on the EUS for its entire life, you must use the application migration tooling to move to a new cluster

Upgrades vs. Migrations



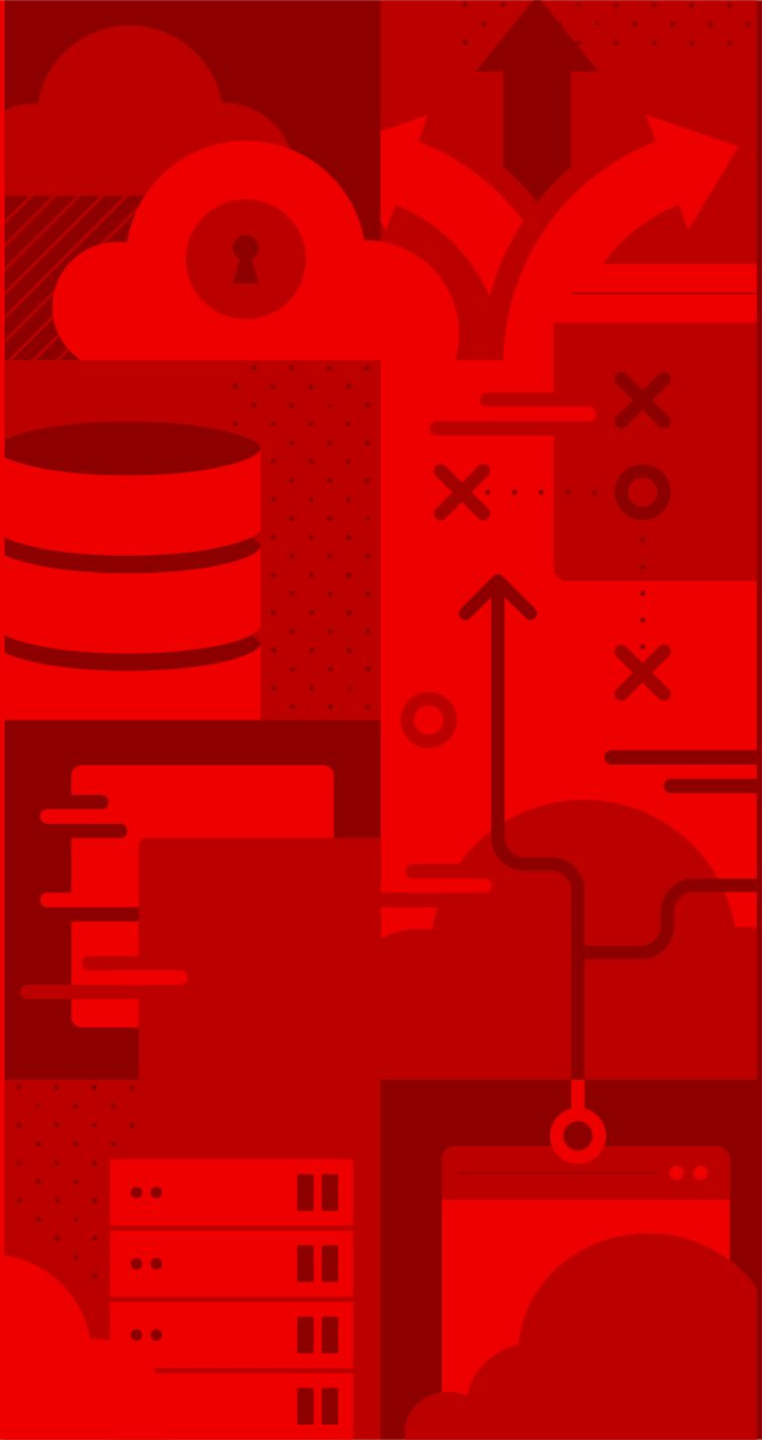
OTA Upgrades
Works between two minor releases in a serial manner.

Happy path = migrate through each version
On a regular cadence, migrate to the next supported version.

Optional path = migration tooling
If you fall more than two releases behind, you must use the application migration tooling to move to a new cluster.

Current minor release
Full support for all bugs and security issues
1 month full support overlap with next release to aid migrations

Previous minor release
Fixes for critical bugs and security issues for 5 months



Operations and infrastructure deep dive

Red Hat Enterprise Linux CoreOS

The OpenShift operating system

Red Hat Enterprise Linux

	RED HAT® ENTERPRISE LINUX® **General Purpose OS**	RED HAT® ENTERPRISE LINUX CoreOS **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

Product Manager: Ben Breard

Generally Available

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.

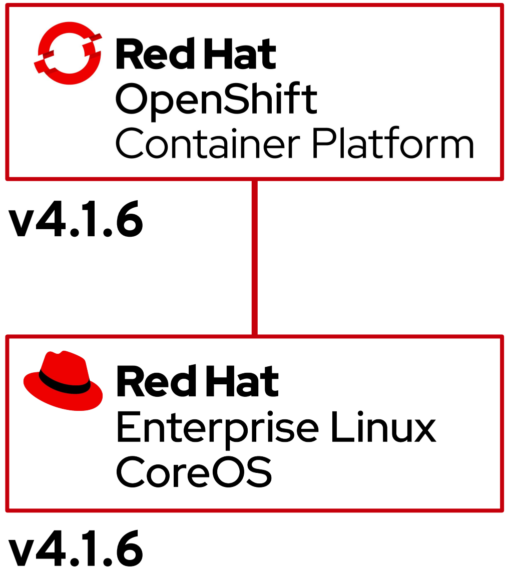
Red Hat Enterprise Linux CoreOS is managed by the cluster

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 OpenShift Container Platform

v4.1.6

Red Hat Enterprise Linux CoreOS

v4.1.6

cri-o

A lightweight, OCI-compliant container runtime

Minimal and Secure Architecture

Optimized for Kubernetes

Runs any OCI-compliant image (including docker)

BROAD ECOSYSTEM OF WORKLOADS

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

podman

A docker-compatible CLI for containers

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

buildah

Secure & flexible OCI container builds

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

OpenShift 4 installation

Installer and user-provisioned infrastructure, bootstrap, and more

OpenShift Bootstrap Process: Self-Managed Kubernetes

How to boot a self-managed cluster:

- OpenShift 4 is unique in that management extends all the way down to the operating system
- Every machine boots with a configuration that references resources hosted in the cluster it joins, enabling cluster to manage itself
- Downside is that every machine looking to join the cluster is waiting on the cluster to be created
- Dependency loop is broken using a bootstrap machine, which acts as a temporary control plane whose sole purpose is bringing up the permanent control plane nodes
- Permanent control plane nodes get booted and join the cluster leveraging the control plane on the bootstrap machine
- Once the pivot to the permanent control plane takes place, the remaining worker nodes can be booted and join the cluster

Bootstrapping process step by step:

1. Bootstrap machine boots and starts hosting the remote resources required for master machines to boot.
2. Master machines fetch the remote resources from the bootstrap machine and finish booting.
3. Master machines use the bootstrap node to form an etcd cluster.
4. Bootstrap node starts a temporary Kubernetes control plane using the newly-created etcd cluster.
5. Temporary control plane schedules the production control plane to the master machines.
6. Temporary control plane shuts down, yielding to the production control plane.
7. Bootstrap node injects OpenShift-specific components into the newly formed control plane.
8. Installer then tears down the bootstrap node or if user-provisioned, this needs to be performed by the administrator.

How everything deployed comes under management

Masters (Special)

- Terraform provisions initial masters*
- Machine API adopts existing masters post-provision
- Each master is a standalone Machine object
- Termination protection (avoid self-destruction)

Workers

- Each Machine Pool corresponds to MachineSet
- Optionally autoscale (min,max) and health check (replace if not ready > X minutes)

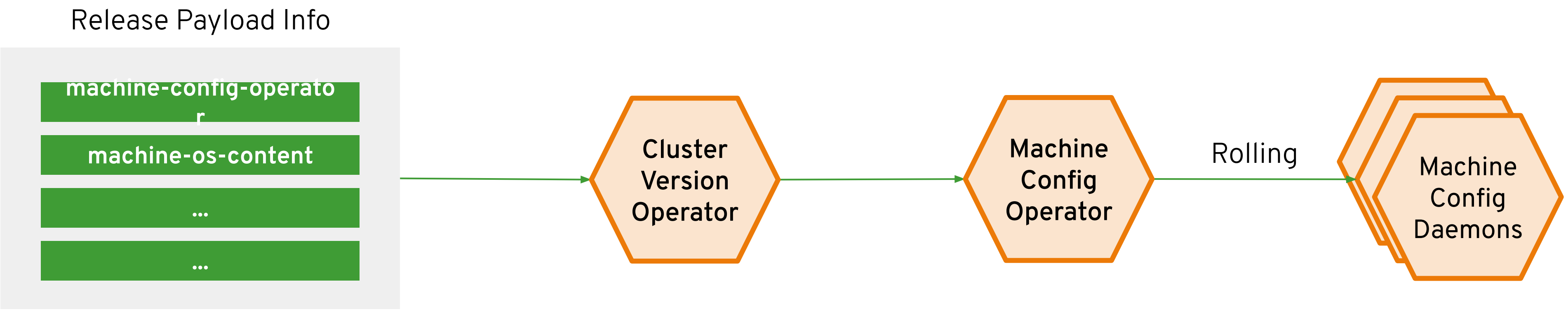
Multi-AZ

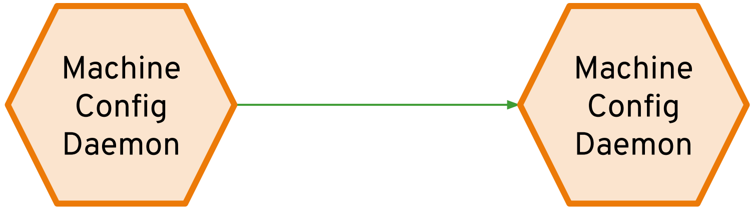
- MachineSets scoped to single AZ
- Installer stripes N machine sets across AZs by default
- Post-install best effort balance via cluster autoscaler

OpenShift 4 Cluster Management

Powered by Operators, OpenShift 4 automates many cluster management activities

Over-the-air updates

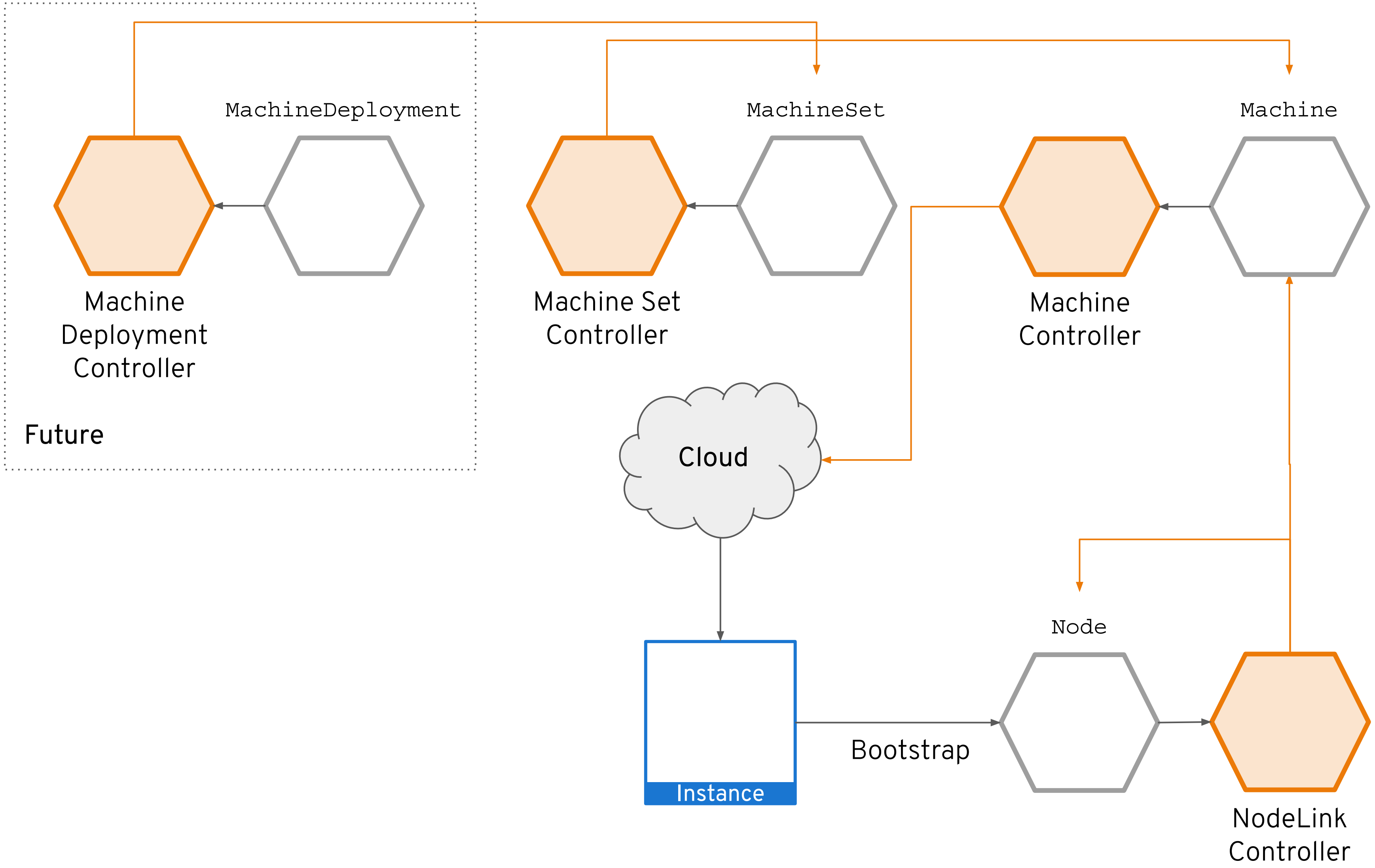




Download and mount update content into host

Update host using mounted content

Cloud API


MachineDeployment
MachineSet
Machine
Machine Deployment Controller
Machine Set Controller
Machine Controller
Future
Cloud
Node
Instance
Bootstrap
NodeLink Controller

OpenShift Security

Features, mechanisms and processes for container and platform isolation

CONTROL
Application Security

- Container Content
- CI/CD Pipeline
- Container Registry
- Deployment Policies

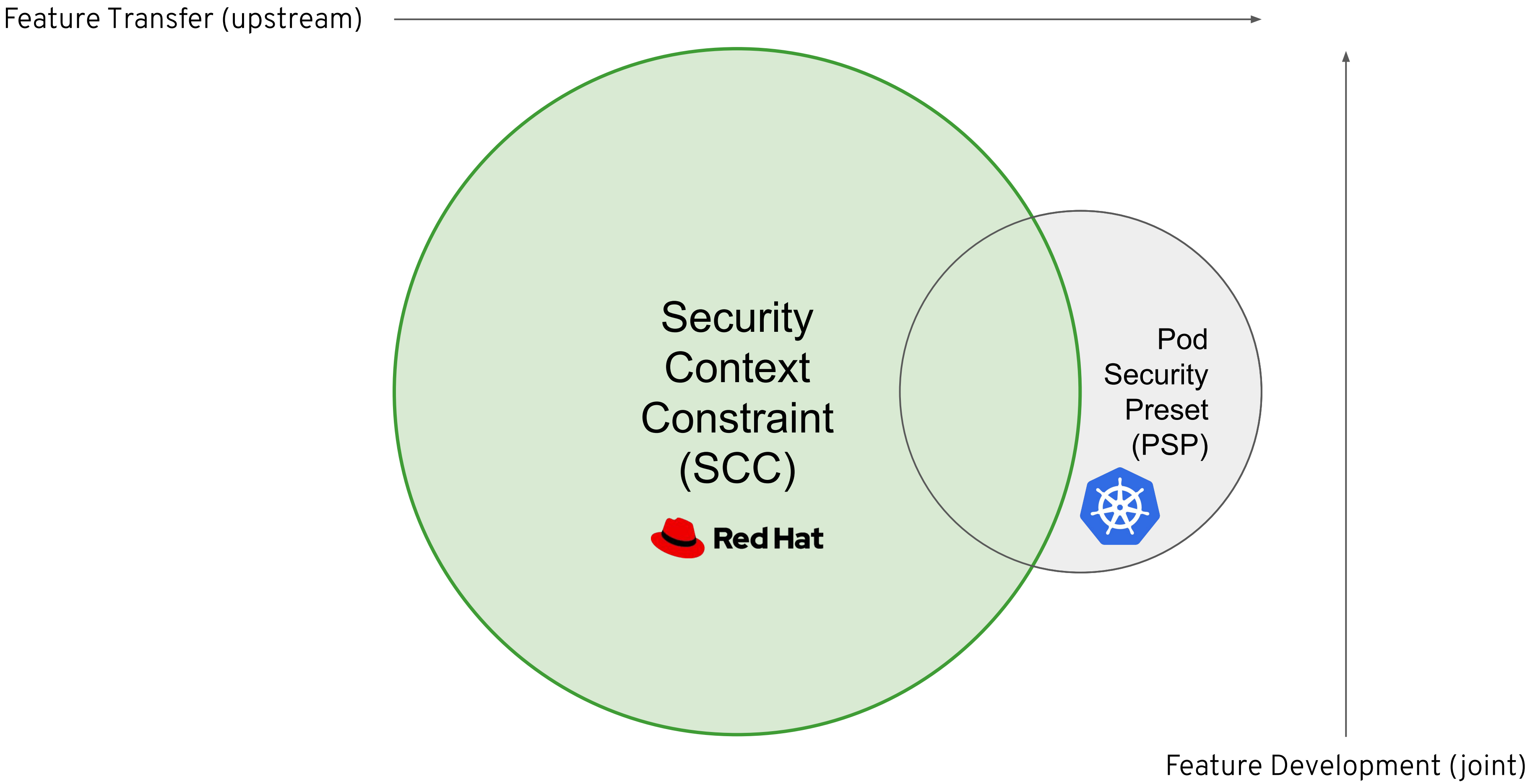
DEFEND
Infrastructure

- Container Platform
- Container Host Multi-tenancy
- Network Isolation
- Storage
- Audit & Logging
- API Management

EXTEND

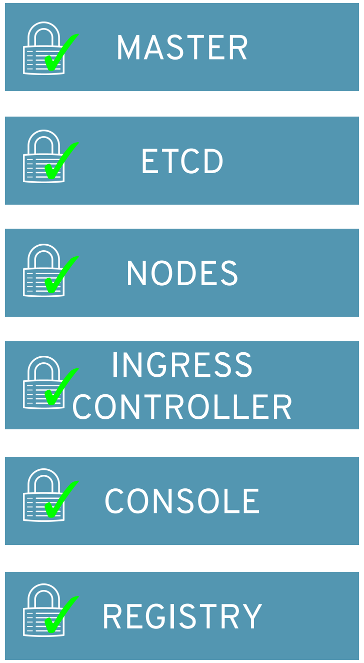
- Security Ecosystem

Extended Depth of Protection


Feature Transfer (upstream)
Security Context Constraint (SCC)
Red Hat
Pod Security Preset (PSP)
Feature Development (joint)

Certificates and Certificate Management

- OpenShift provides its own internal CA
- Certificates are used to provide secure connections to
  - master (APIs) and nodes
  - Ingress controller and registry
  - etcd
- Certificate rotation is automated
- Optionally configure external endpoints to use custom certificates



✓ MASTER

✓ ETCD

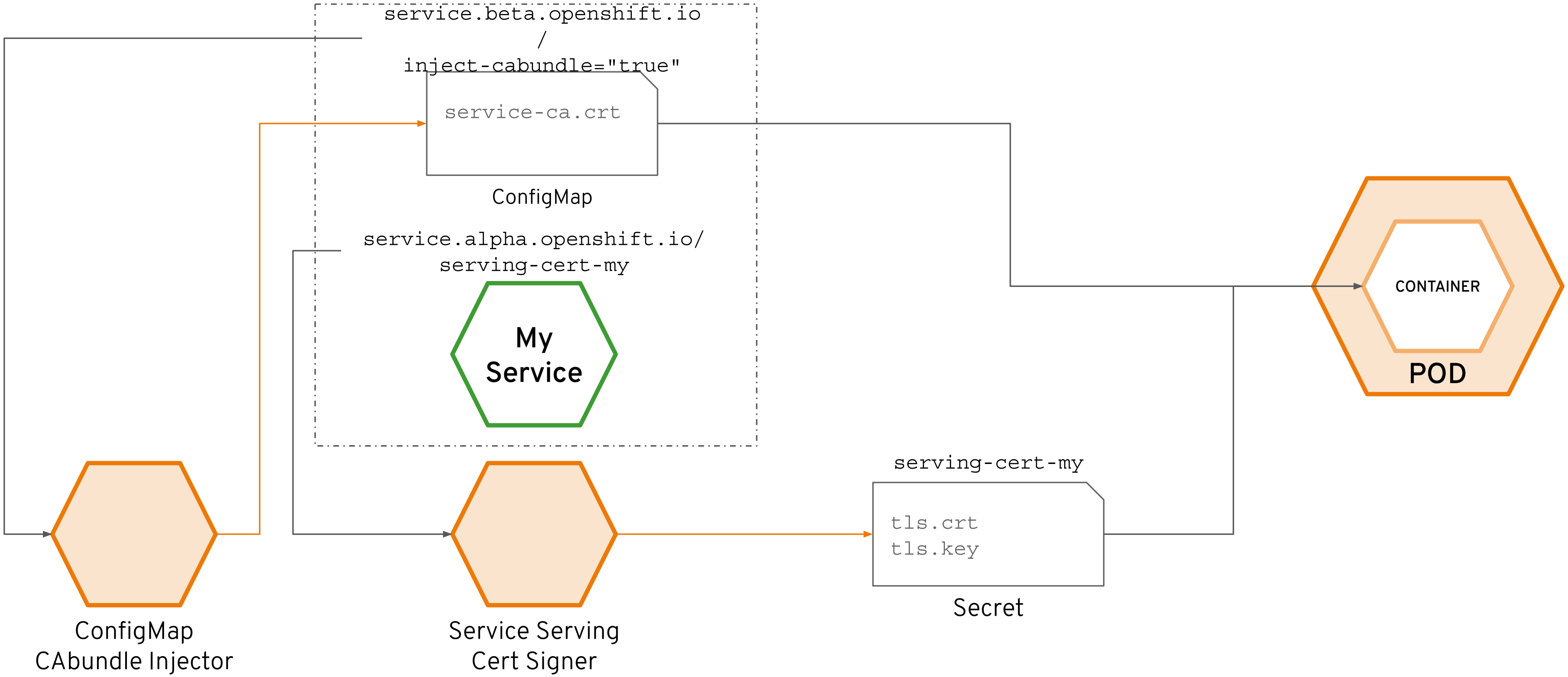
✓ NODES

✓ INGRESS CONTROLLER

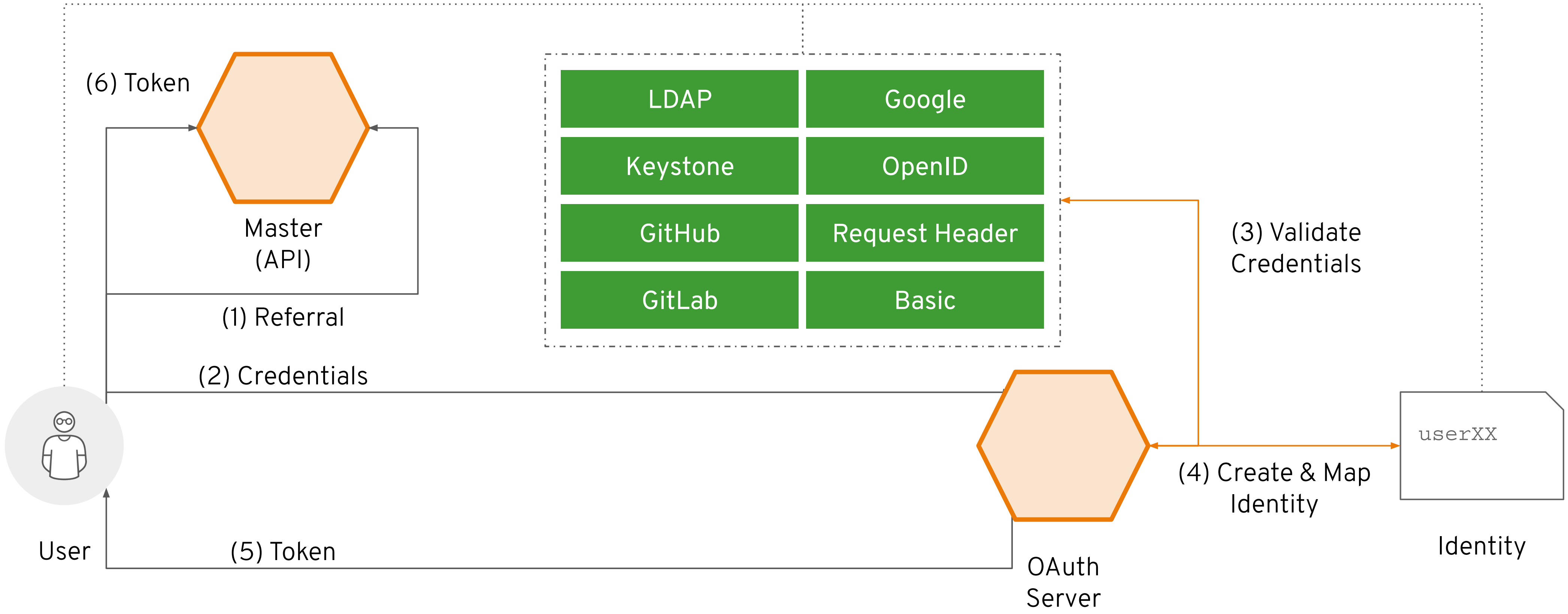
✓ CONSOLE

✓ REGISTRY

Service Certificates



Identity and Access Management



Fine-Grained RBAC

- Project scope & cluster scope available
- Matches request attributes (verb,object,etc)
- If no roles match, request is denied (deny by default)
- Operator- and user-level roles are defined by default
- Custom roles are supported

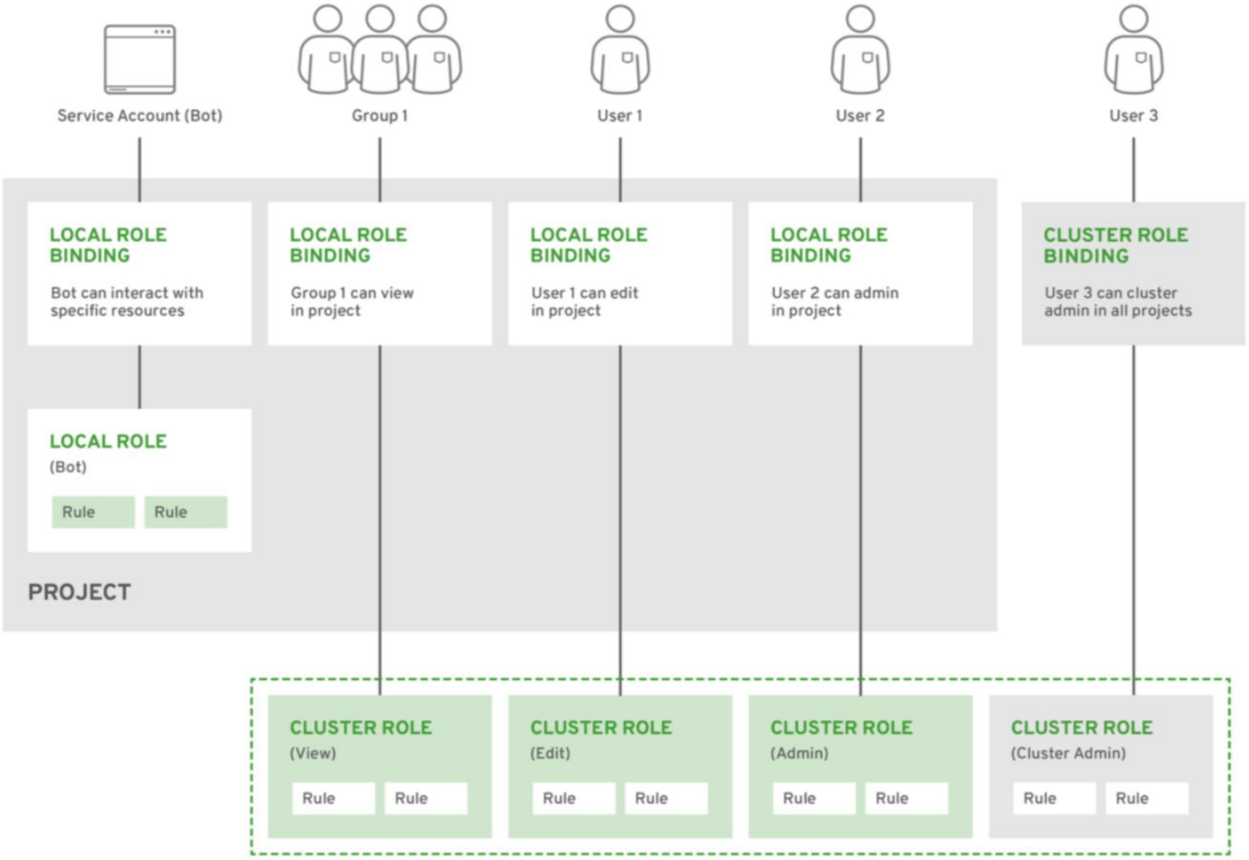


Figure 12 - Authorization Relationships

OpenShift Monitoring

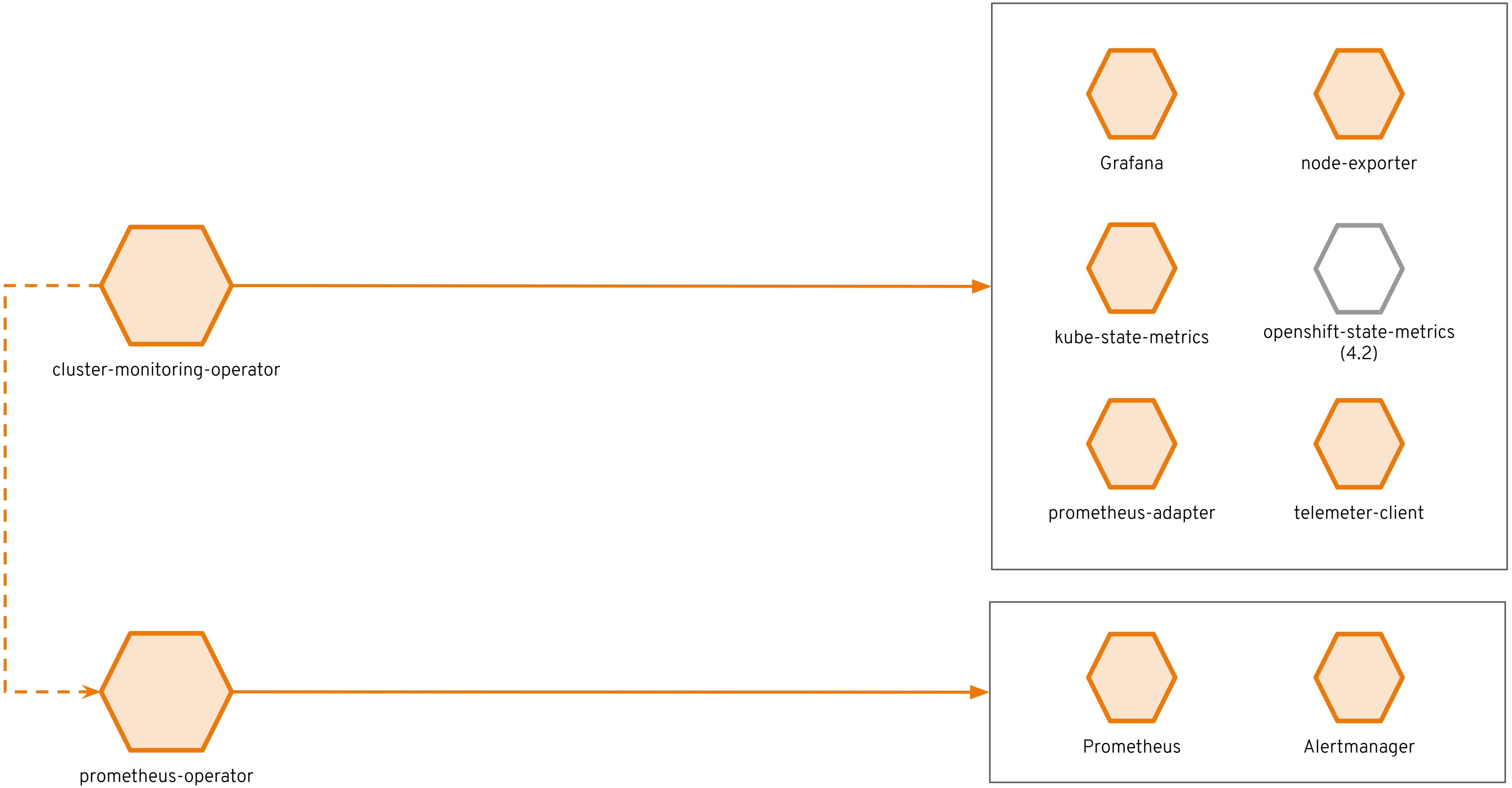
An integrated cluster monitoring and alerting stack

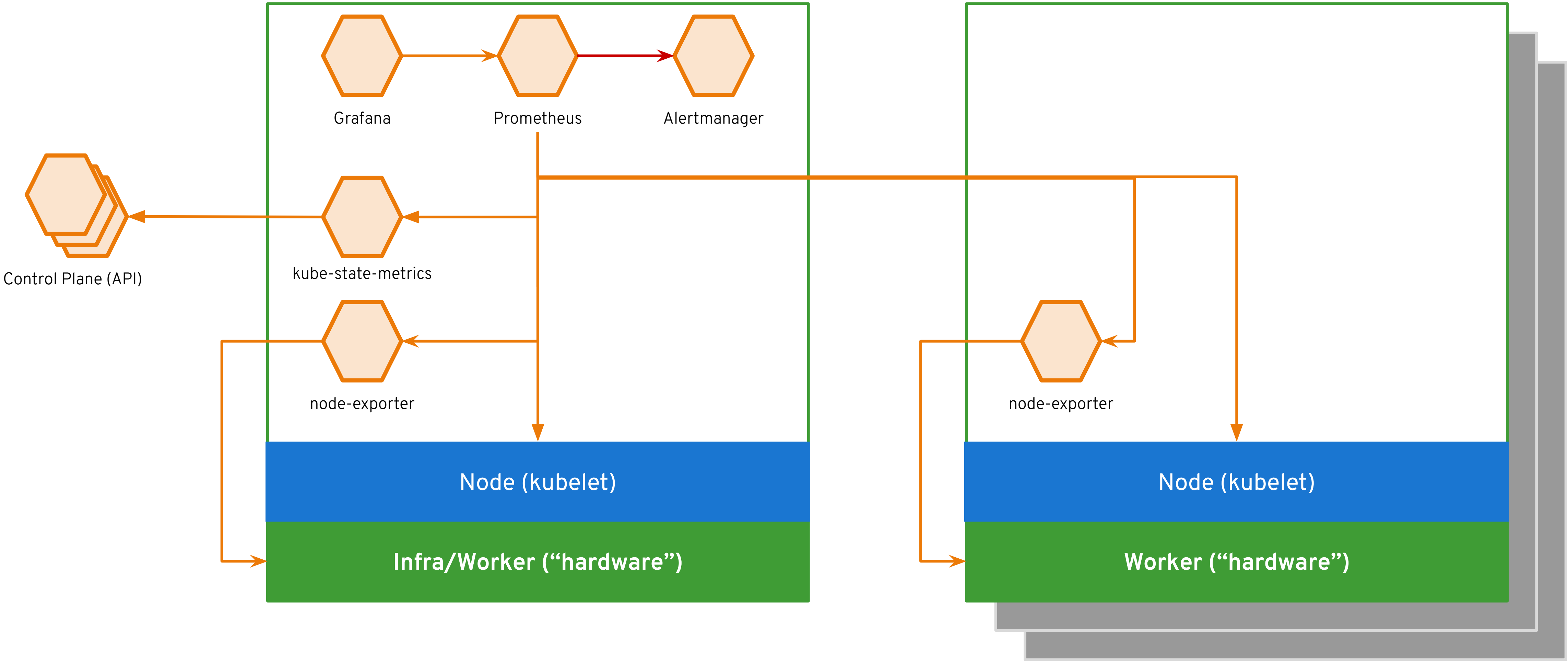
OpenShift Cluster Monitoring

Metrics collection and storage via Prometheus, an open-source monitoring system time series database.

Alerting/notification via Prometheus' Alertmanager, an open-source tool that handles alerts send by Prometheus.

Metrics visualization via Grafana, the leading metrics visualization technology.


cluster-monitoring-operator
prometheus-operator
Grafana
node-exporter
kube-state-metrics
openshift-state-metrics
(4.2)
prometheus-adapter
telemeter-client
Prometheus
Alertmanager


Grafana
Prometheus
Alertmanager
Control Plane (API)
kube-state-metrics
node-exporter
node-exporter
Node (kubelet)
Node (kubelet)
Infra/Worker (“hardware”)
Worker (“hardware”)

OpenShift Logging

An integrated solution for exploring and corroborating application logs

Observability via log exploration and corroboration with EFK

Components

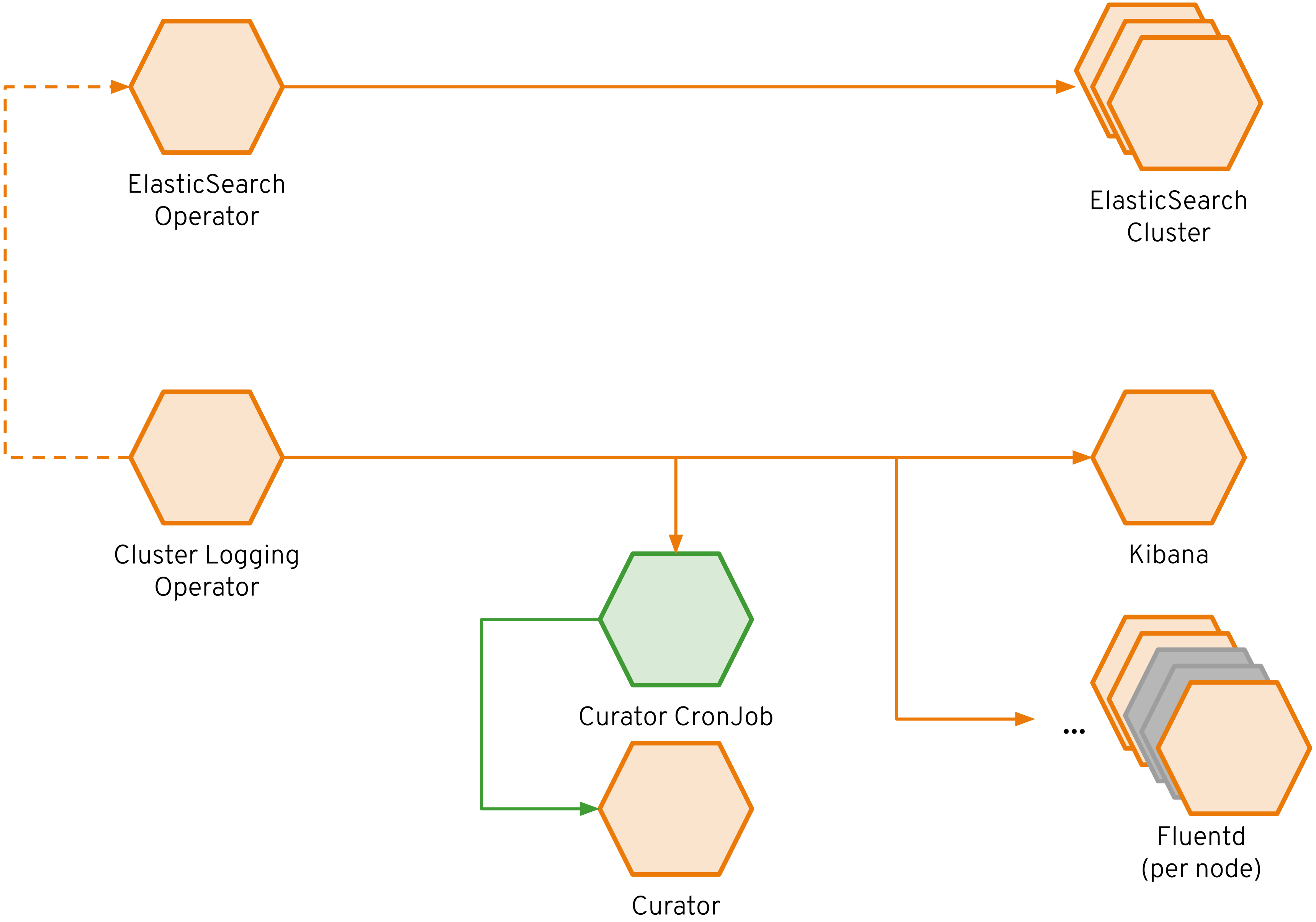
- **Elasticsearch:** a search and analytics engine to store logs
- **Fluentd:** gathers logs and sends to Elasticsearch.
- **Kibana:** A web UI for Elasticsearch.

Access control

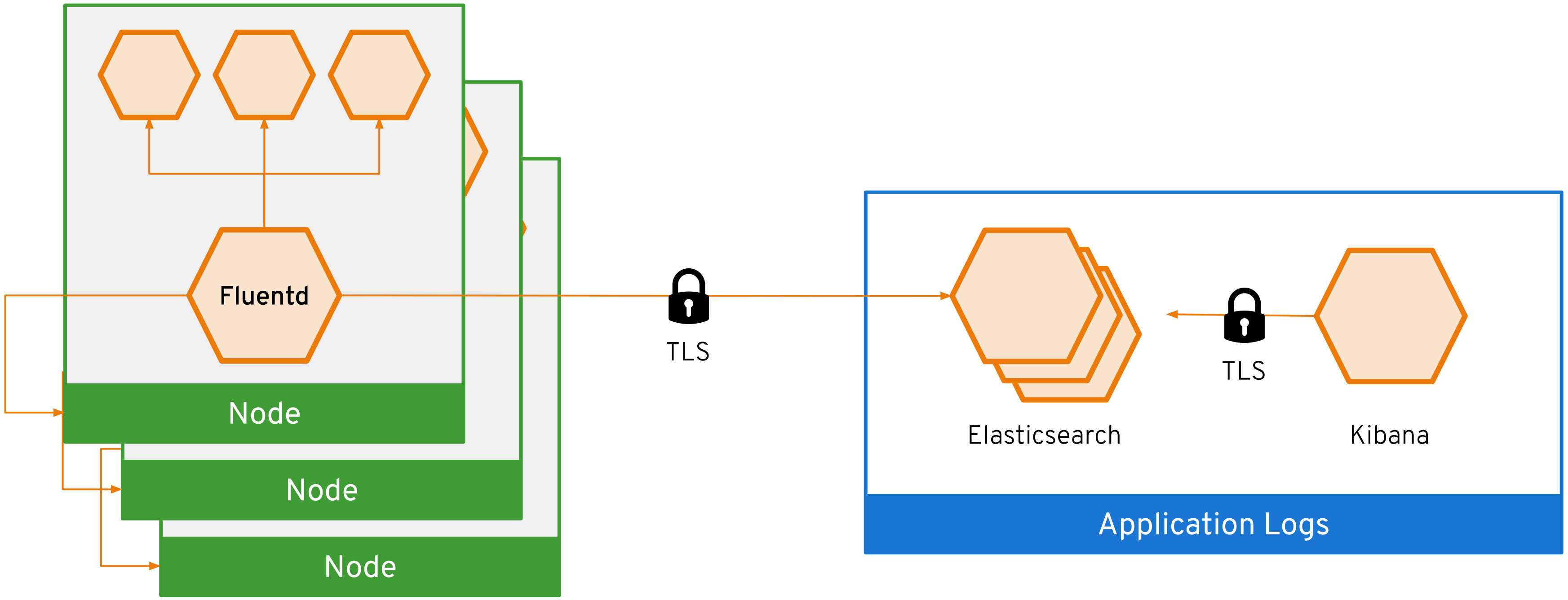
- Cluster administrators can view all logs
- Users can only view logs for their projects

Ability to forward logs elsewhere

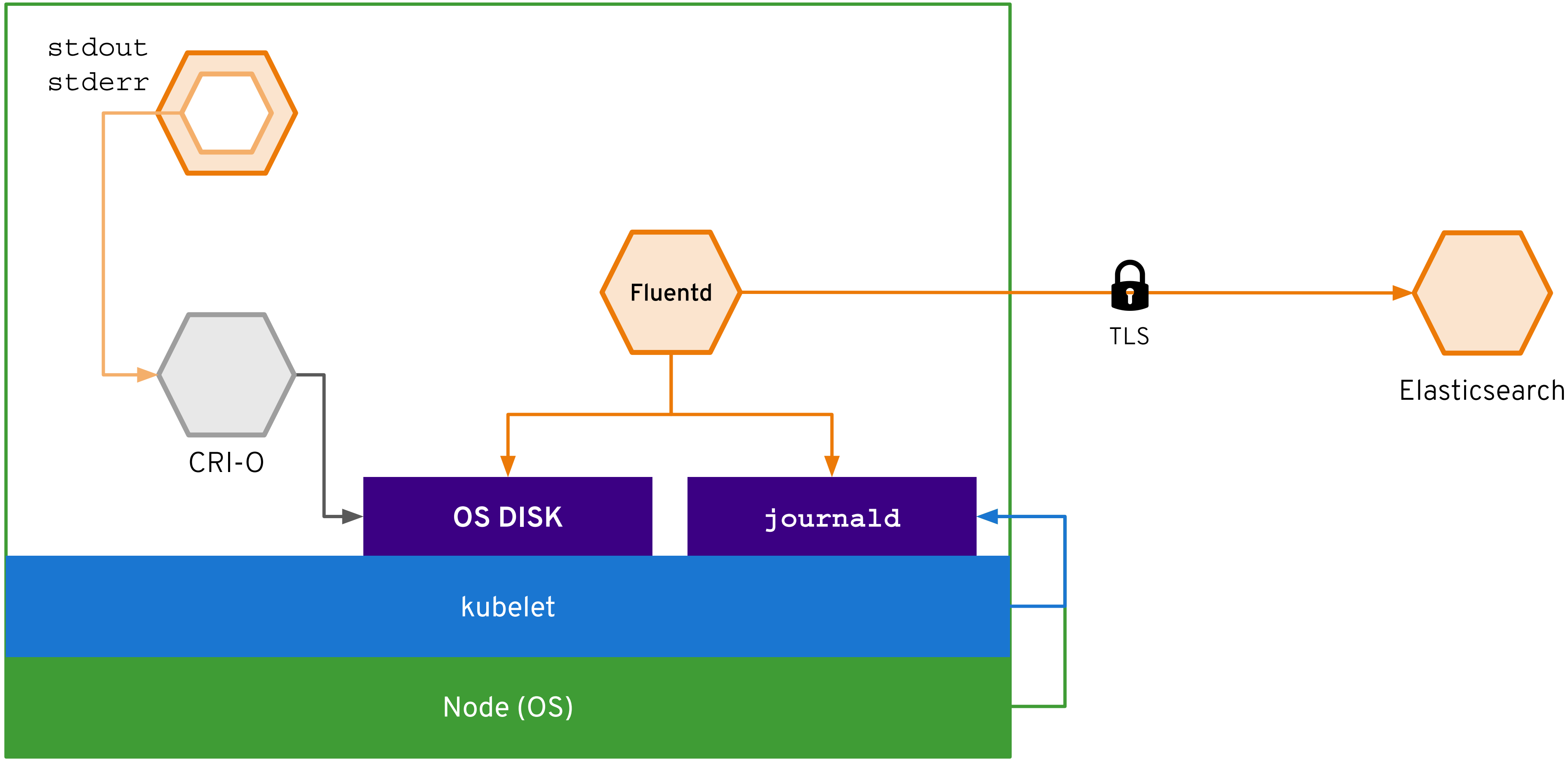
- External elasticsearch, Splunk, etc


ElasticSearch Operator
ElasticSearch Cluster
Cluster Logging Operator
Kibana
Curator CronJob
...
Curator
Fluentd (per node)

Log data flow in OpenShift



Log data flow in OpenShift



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