Linux Container Solutions on IBM Z® & LinuxONE™

IBM® Cloud Paks™ & Red Hat® OpenShift® Container Platform on IBM Z® & LinuxONE™

Workshop ID LXCS1 / ICPOZ1
IBM ATS/WSC/ATG Wildfire Workshop Series

http://www.ibm.com/support/techdocs

Special Notices

This presentation reflects the IBM Advanced Technical Skills organizations' understanding of the technical topic. It was produced and reviewed by the members of the IBM Advanced Technical Skills organization. This document is presented "As-Is" and IBM does not assume responsibility for the statements expressed herein. It reflects the opinions of the IBM Advanced Technical Skills organization. These opinions are based on the author's experiences. If you have questions about the contents of this document, please contact the author at linuxats@us,ibm.com

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

Any and all customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. References in this document to IBM products or services do not imply that IBM intends to make them available in every country. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.

The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk

IBM retains the title to the copyright in this paper, as well as the copyright in all underlying works. IBM retains the right to make derivative works and to republish and distribute this paper to whomever it chooses in any way it chooses.

Trademarks

The following are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

IBM, the IBM logo, DB2, Redbooks, Tivoli Enterprise Console, WebSphere, z/OS, System z, z/VM.

A full list of U.S. trademarks owned by IBM may be found at http://www.ibm.com/legal/copytrade.shtml.

Microsoft, Windows, Windows NT, Internet Explorer, and the Windows logo are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

Intel and Pentium are registered trademarks and MMX, Pentium II Xeon and Pentium III Xeon are trademarks of Intel Corporation in the United States and/or other countries.

Other company, product and service names may be trademarks or service marks of others.

Special Notices and Trademarks

IBM Cloud Paks & Red Hat OpenShift Container Platform on IBM Z & LinuxONE

Rest of today: Hands-on labs

Let us know if you have any questions!

Schedule for the day

Start 10:00 AM Eastern

Presentation

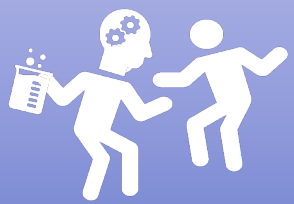
- Overview of OpenShift
- Overview of IBM Cloud Paks

Presentation

- OpenShift on Z technical deep dive

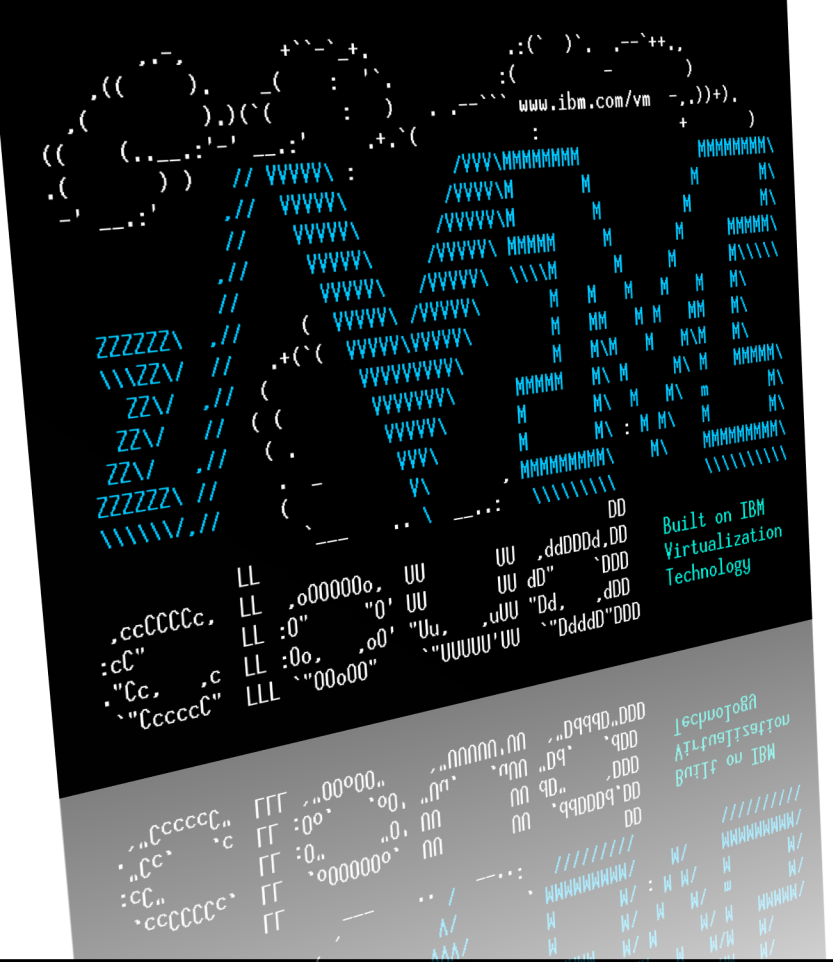
Hands-on Labs

- Self-paced
- Non-sequential



End 4:00 PM Eastern

Red Hat® OpenShift® Deep Dive Good Practices & Lessons Learned



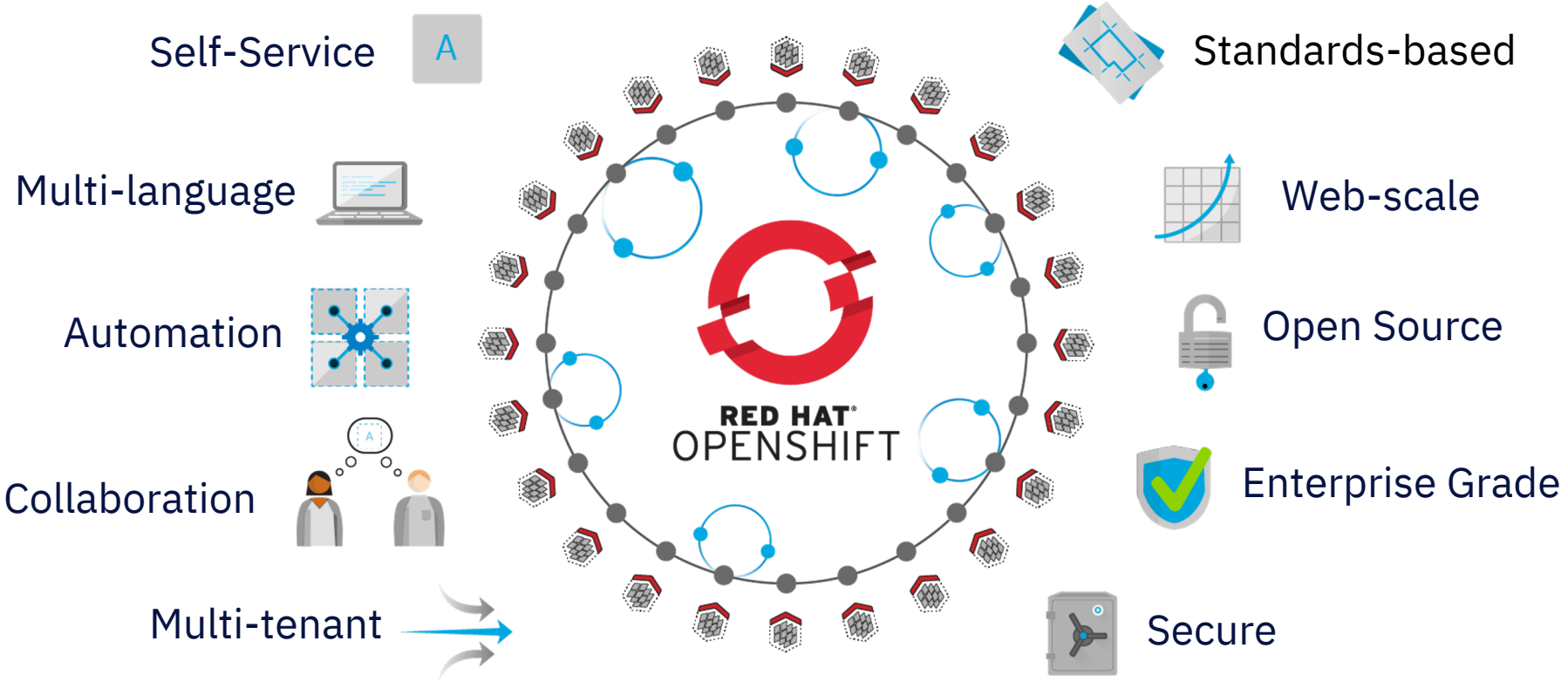
z/VM

KVM

LPAR

Virtualization

Functional overview

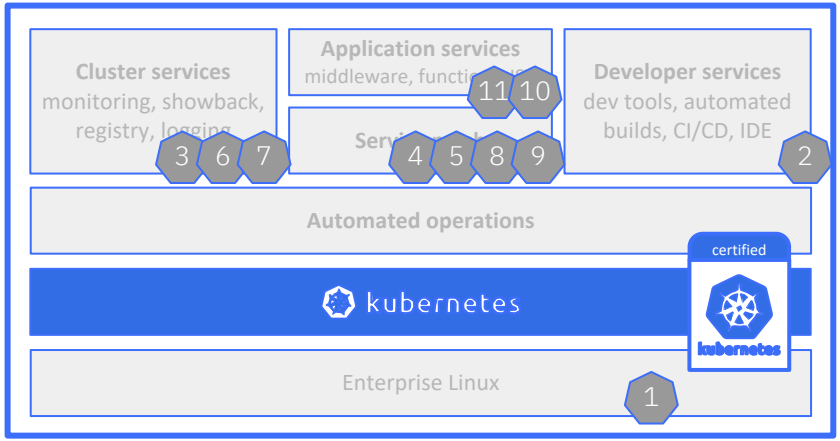


kubernetes

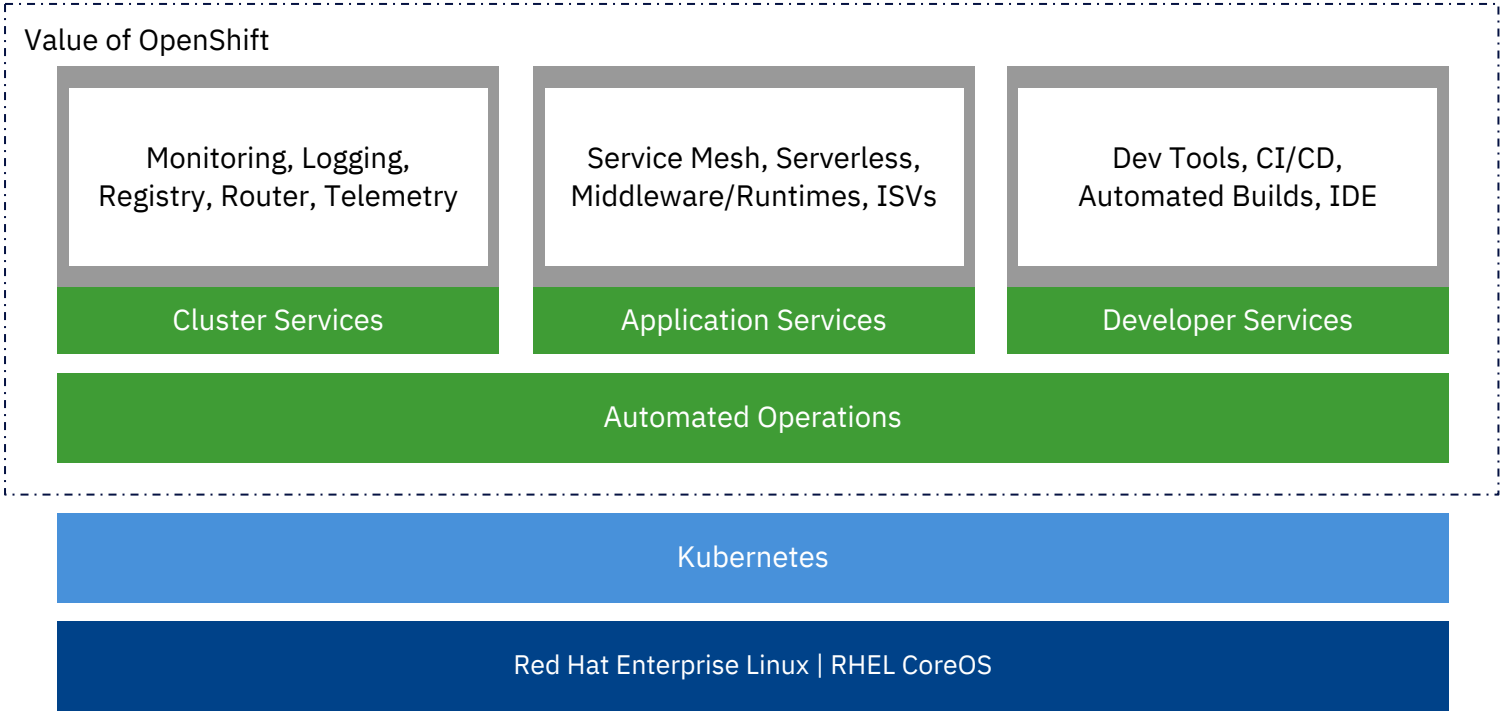
Lacks many essential components

1. Operating system
2. Container runtime (CRI-O, Containerd, Docker, etc).
3. Image registry
4. Software-defined networking
5. Load-balancer and routing
6. Log management
7. Container metrics and monitoring
8. DNS
9. Load balancing
10. Ingress
11. RBAC

The customer (or third-party) must configure, integrate, operate and support additional components to be fully operational.



What's needed to put Kubernetes into production?

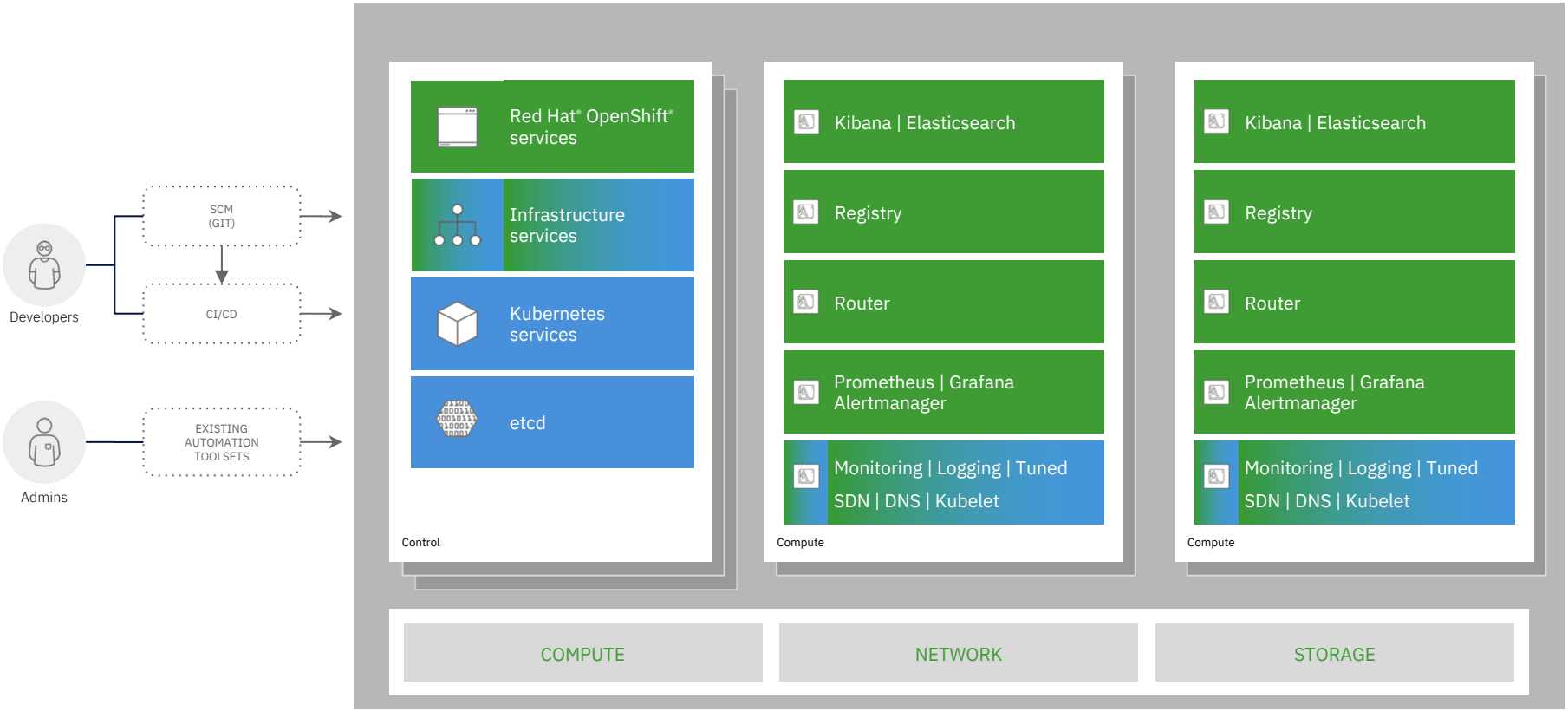


Best IT Ops Experience

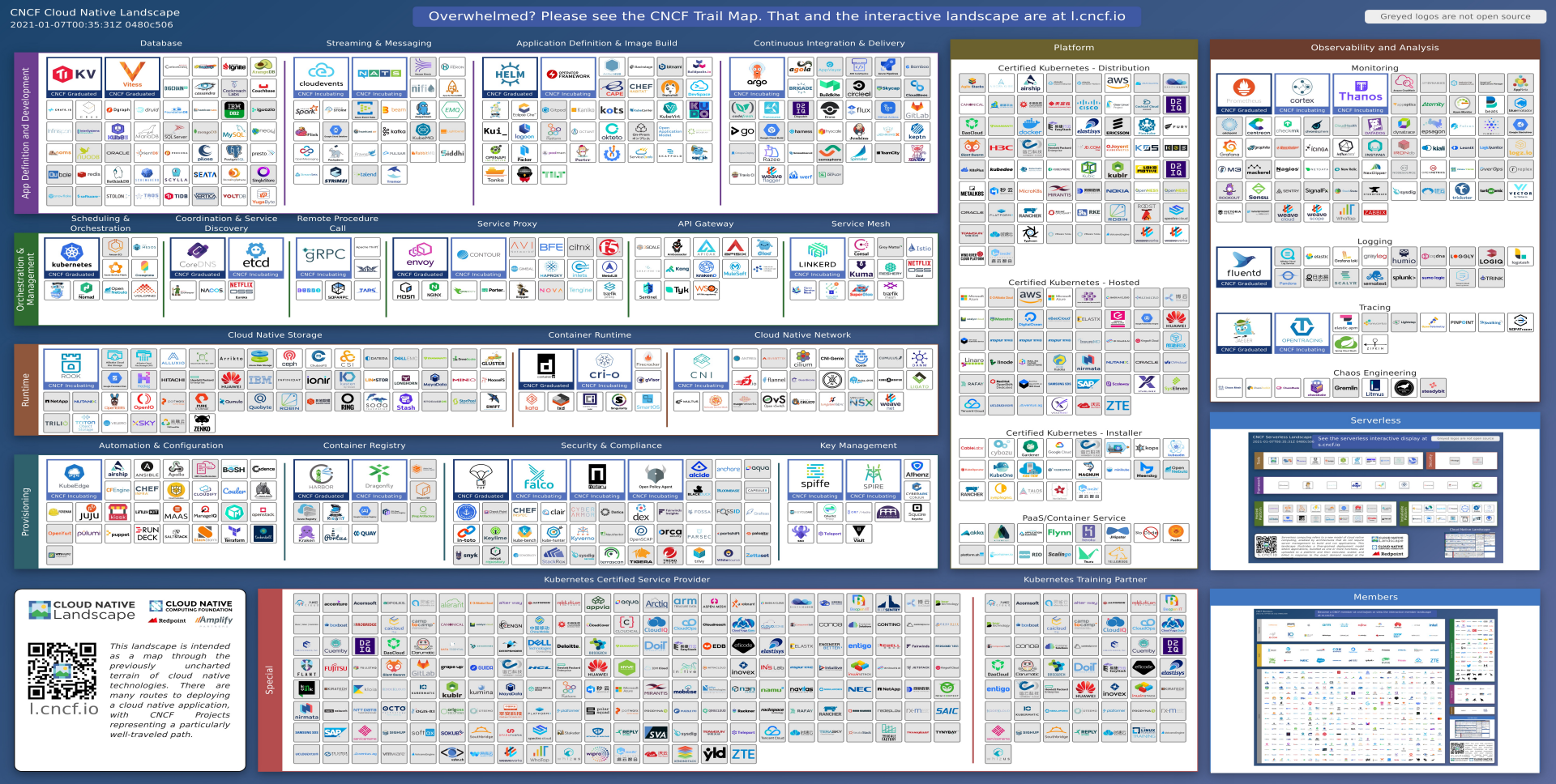
CaaS ⟷ PaaS ⟷ FaaS

Best Developer Experience

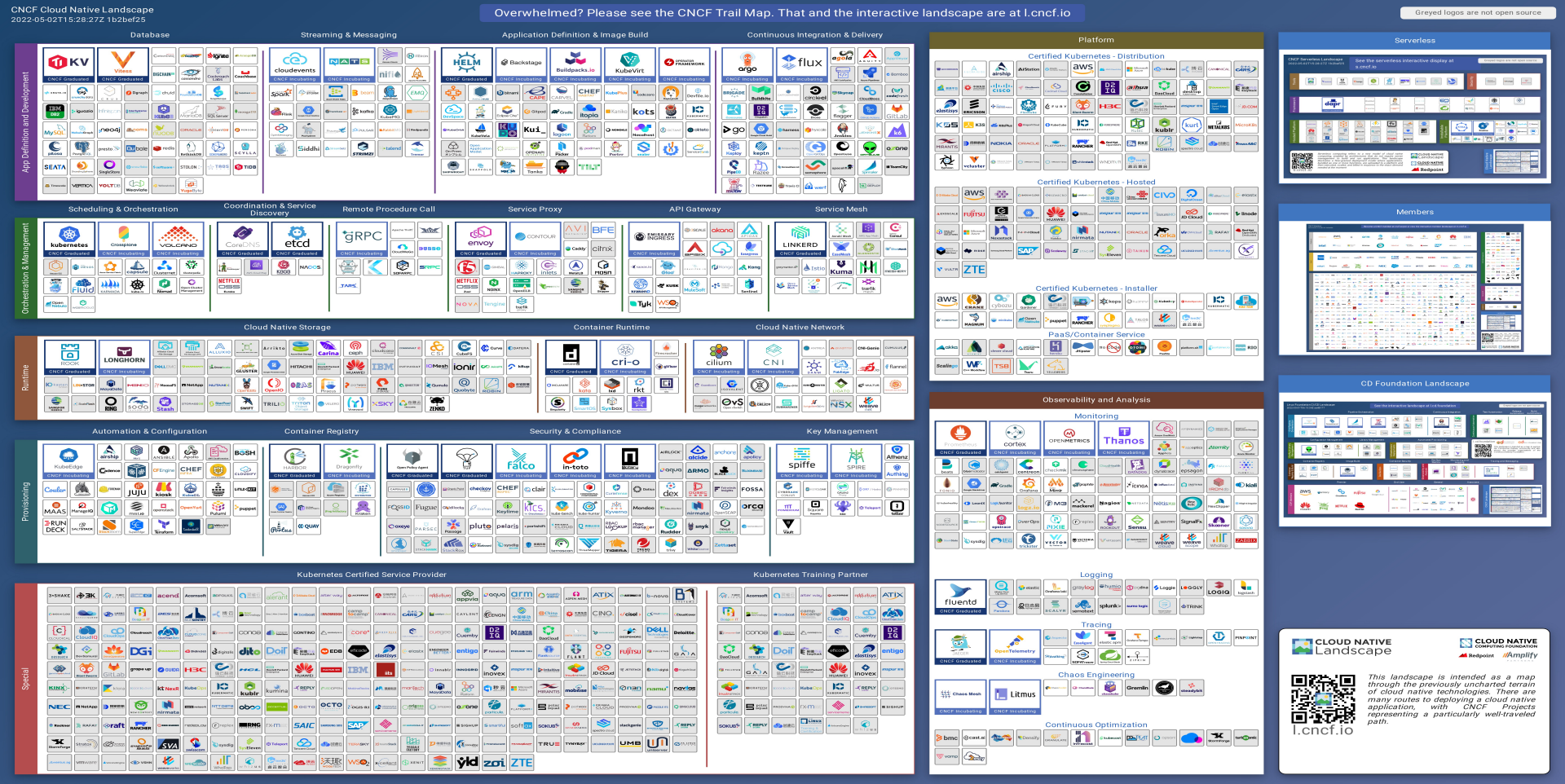
A best-of-breed concept



Architectural overview



CNCF Cloud Native Landscape – January 2021



CNCF Cloud Native Landscape – May 2022

IBM WSC washington systems center

CLOUD NATIVE COMPUTING FOUNDATION

CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape *l.cncf.io* has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer

cncf.io/training

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider

cncf.io/kcsp

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally

cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

l.cncf.io

v20200501

1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing
- Argo is a set of Kubernetes-native tools for deploying and running jobs, applications, workflows, and events using GitOps paradigms such as continuous and progressive delivery and MLops

argo — CNCF Incubating

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ck
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

kubernetes — CNCF Graduated

HELM — CNCF Graduated

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

Prometheus — CNCF Graduated

fluentd — CNCF Graduated

JAEGER — CNCF Graduated

OPENTRACING — CNCF Incubating

5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

envoy — CNCF Graduated

CoreDNS — CNCF Graduated

LINKERD — CNCF Incubating

6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general-purpose policy engine with uses ranging from authorization and admission control to data filtering. Falco is an anomaly detection engine for cloud native.

CNI — CNCF Incubating

Open Policy Agent — CNCF Incubating

Falco — CNCF Incubating

7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high performant distributed transactional key-value store written in Rust.

Vitess — CNCF Graduated

ROOK — CNCF Incubating

etcd — CNCF Incubating

TiKV — CNCF Incubating

8. STREAMING & MESSAGING

When you need higher performance than JSON-REST, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues. CloudEvents is a specification for describing event data in common ways.

gRPC — CNCF Incubating

NATS — CNCF Incubating

cloudevents — CNCF Incubating

9. CONTAINER REGISTRY & RUNTIME

Harbor is a registry that stores, signs, and scans content. You can use alternative container runtimes. The most common, both of which are OCI-compliant, are containerd and CRI-O.

containerd — CNCF Graduated

HARBOR — CNCF Incubating

cri-o — CNCF Incubating

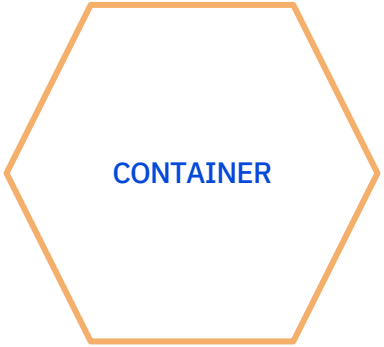
10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

TUF — CNCF Graduated

Notary — CNCF Incubating

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

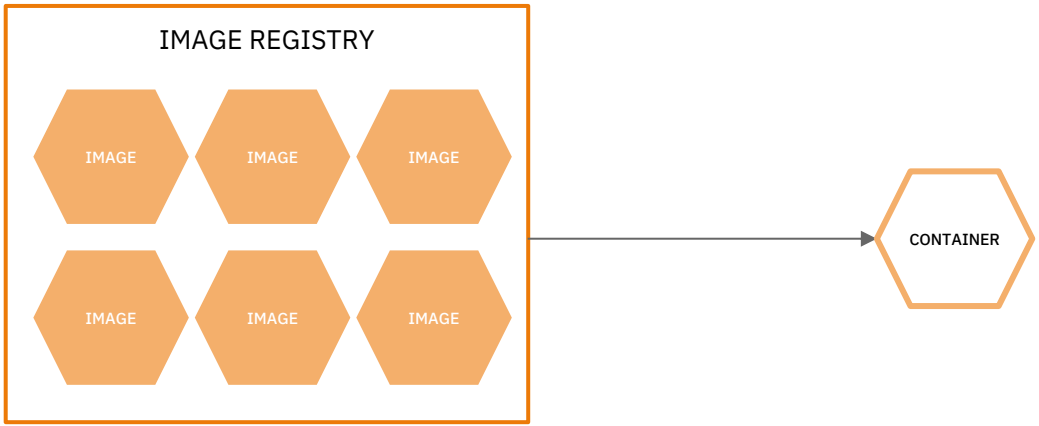
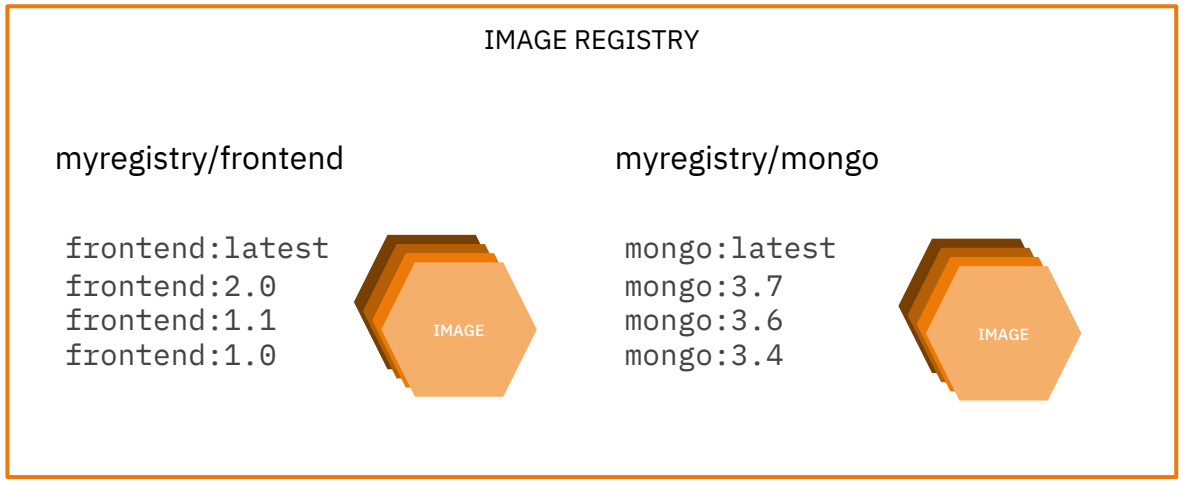


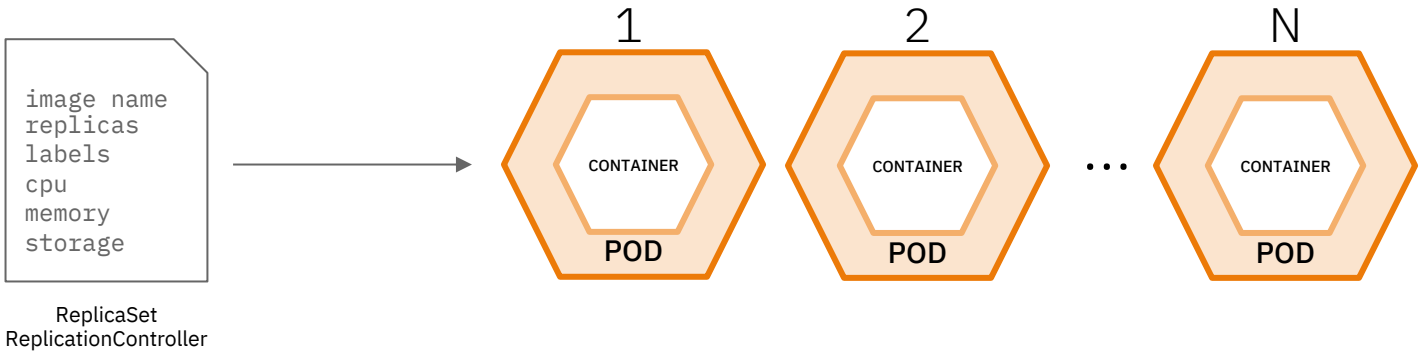
Image registry

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

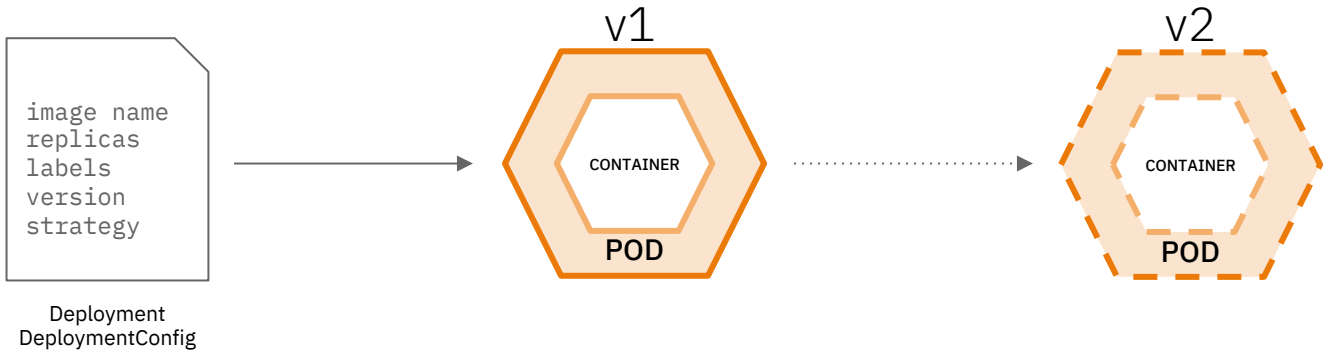


containers are wrapped in pods which are units of deployment and management

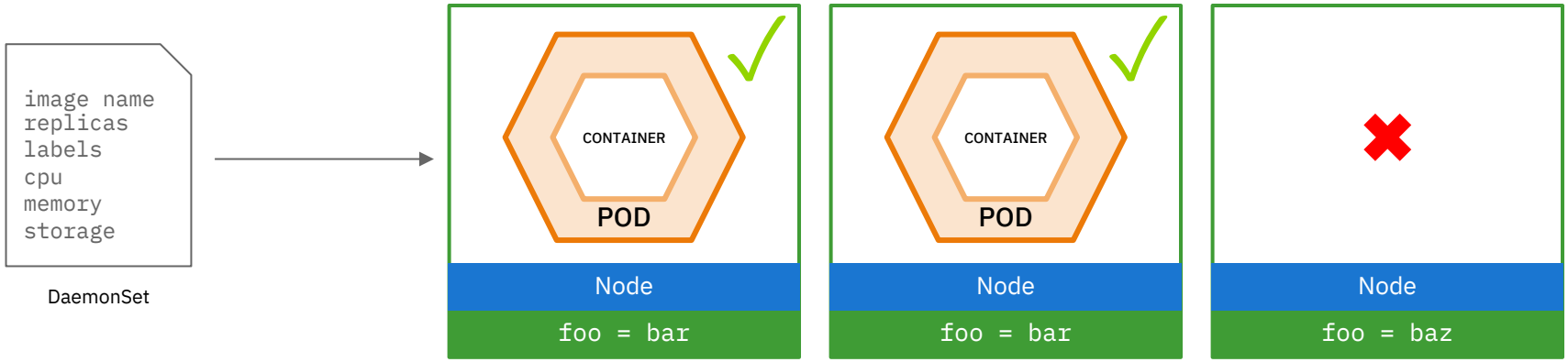
ReplicationControllers & ReplicaSets ensure a specified number of pods are running at any given time



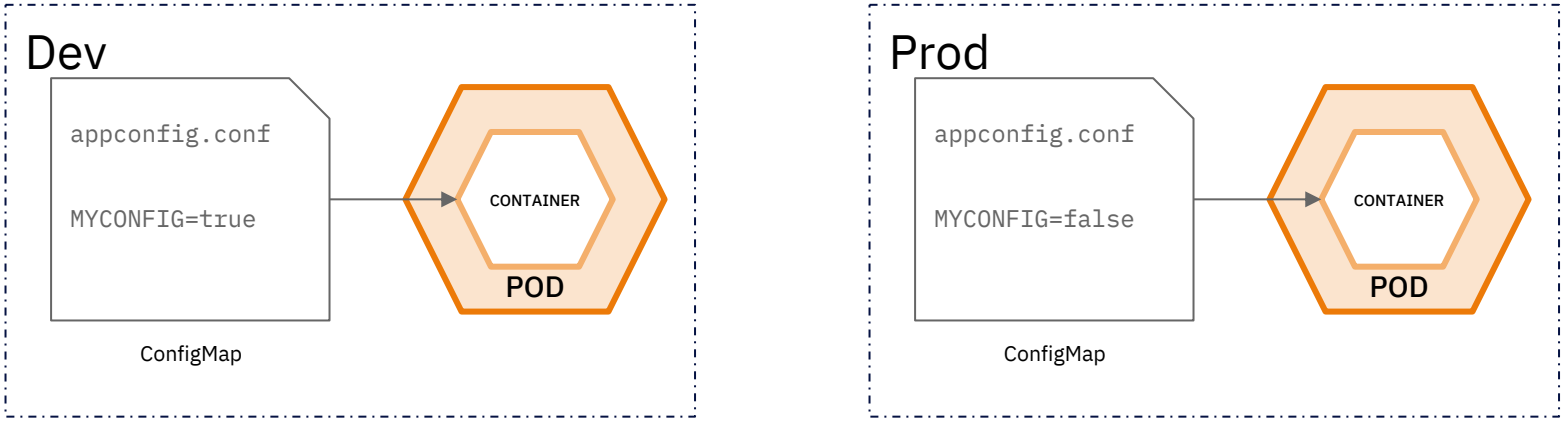
Deployments and DeploymentConfigurations define how to roll out new versions of Pods



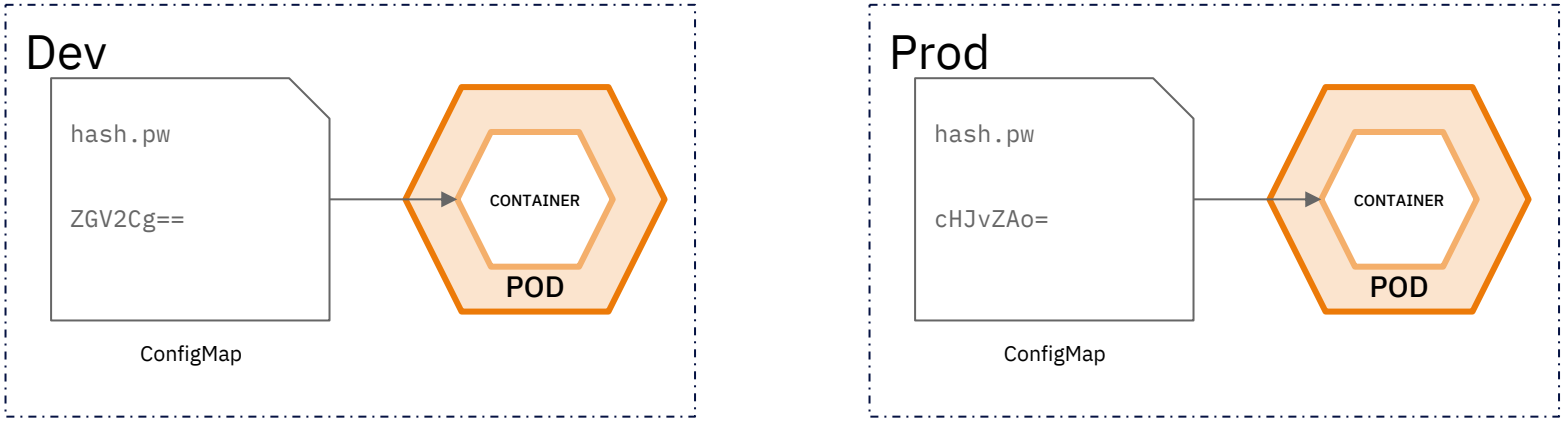
a `daemonset` ensures that all (or some) nodes run a copy of a pod

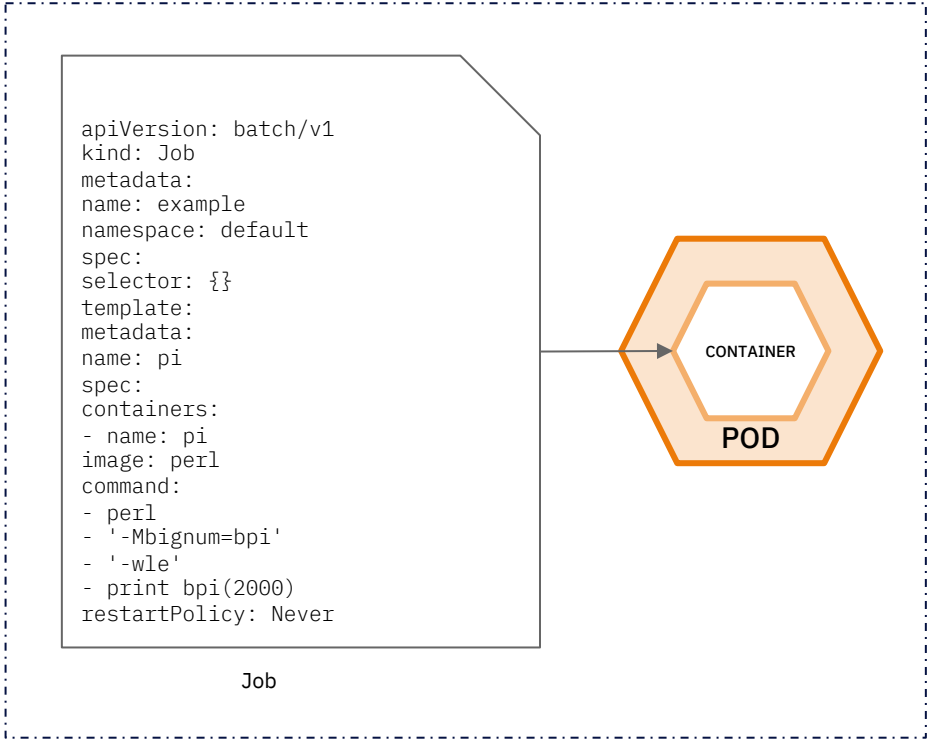


`configmaps` allow you to decouple configuration artifacts from image content

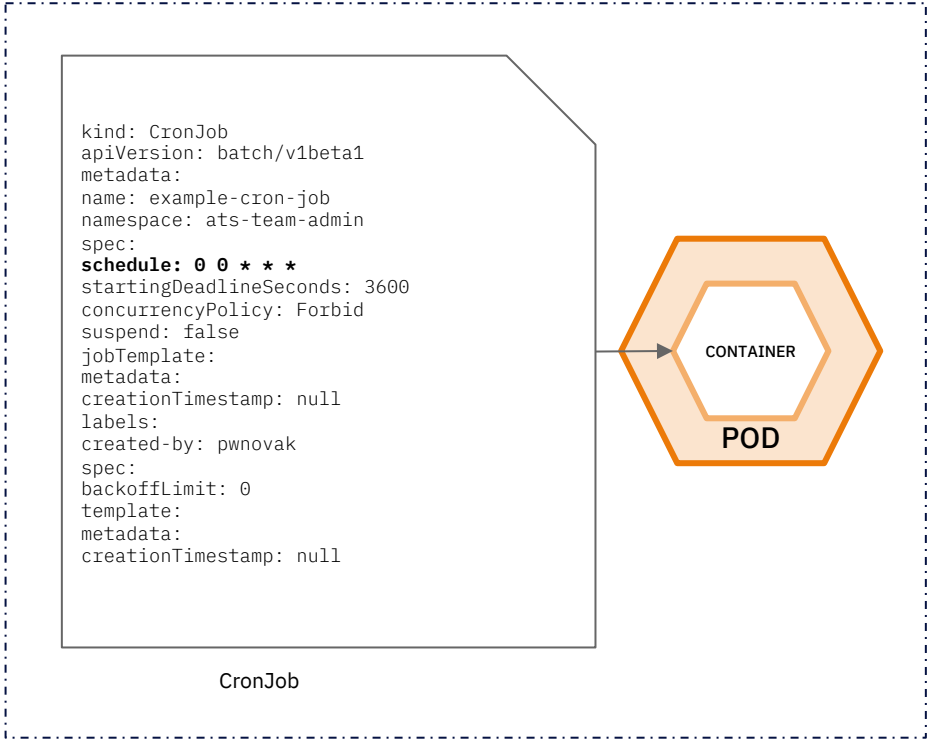


`secrets` provide a mechanism to hold sensitive information such as passwords





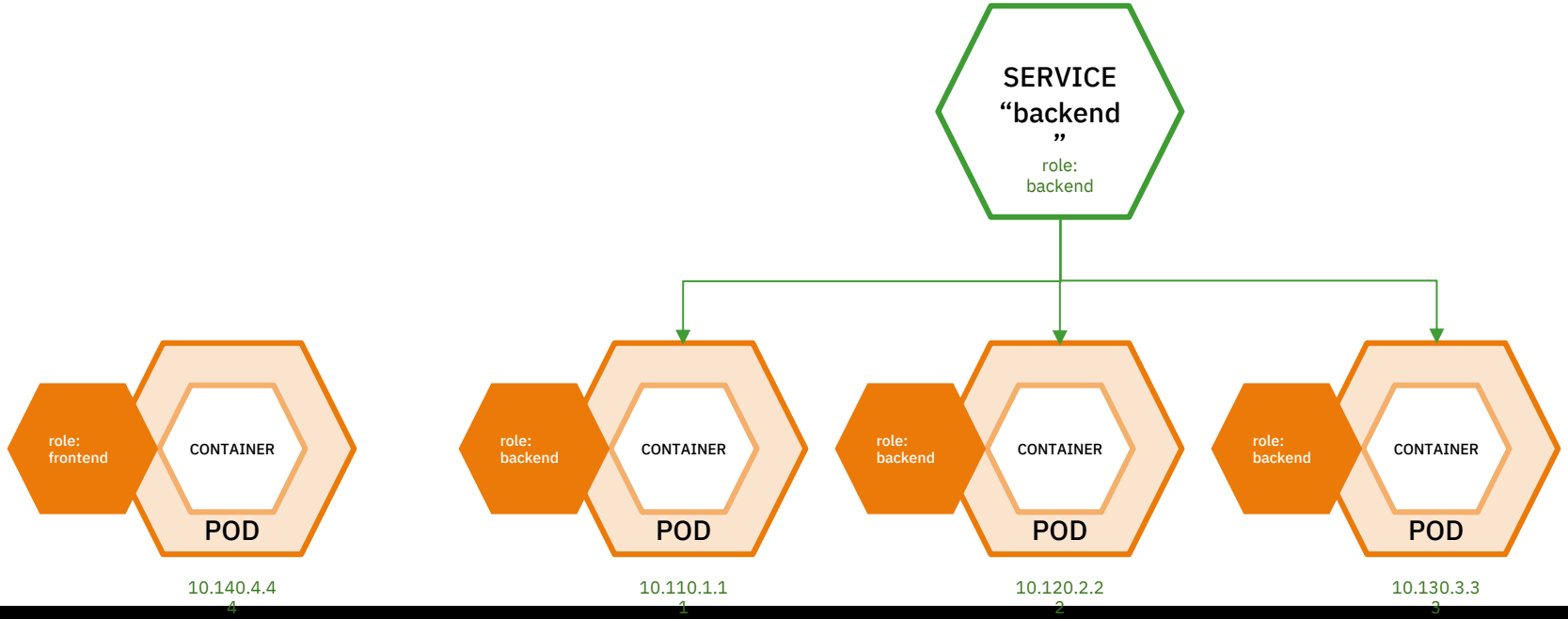
jobs are batch tasks that can be run either manually or via the cluster crontab.



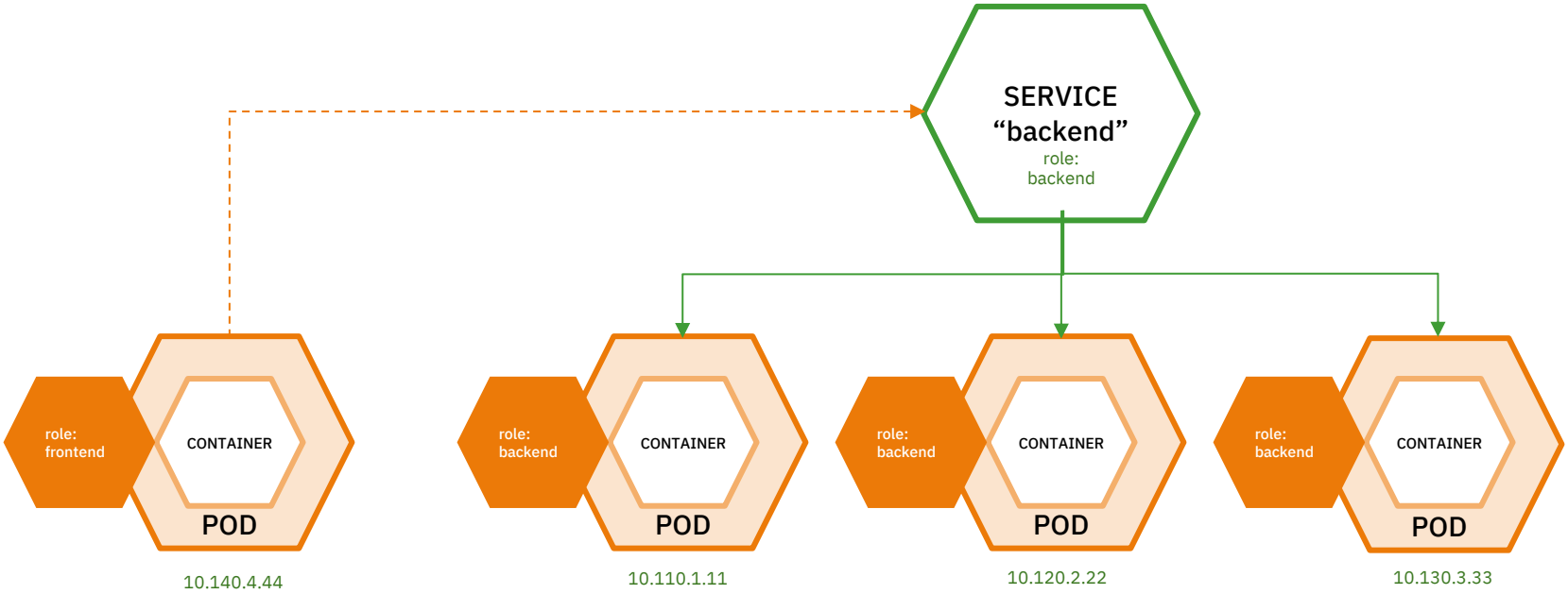
cronjobs are batch tasks run on a defined schedule via the cluster crontab.

Tip: You MUST stagger your scheduling!

services provide internal load-balancing and service discovery across pods

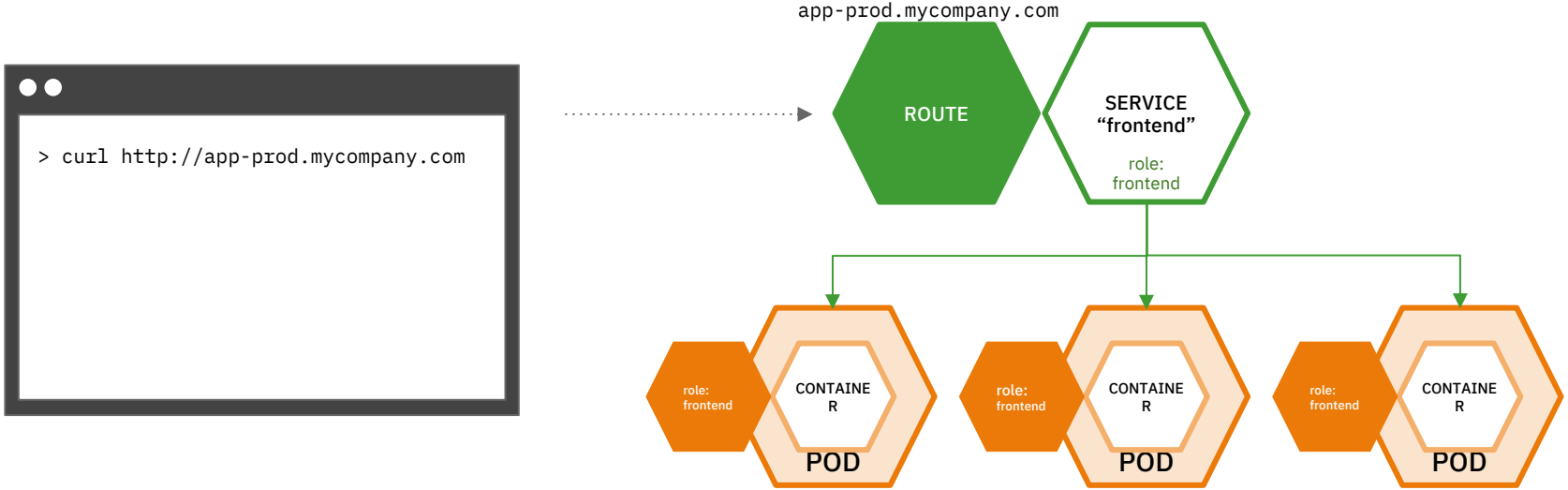


apps can talk to each other via services

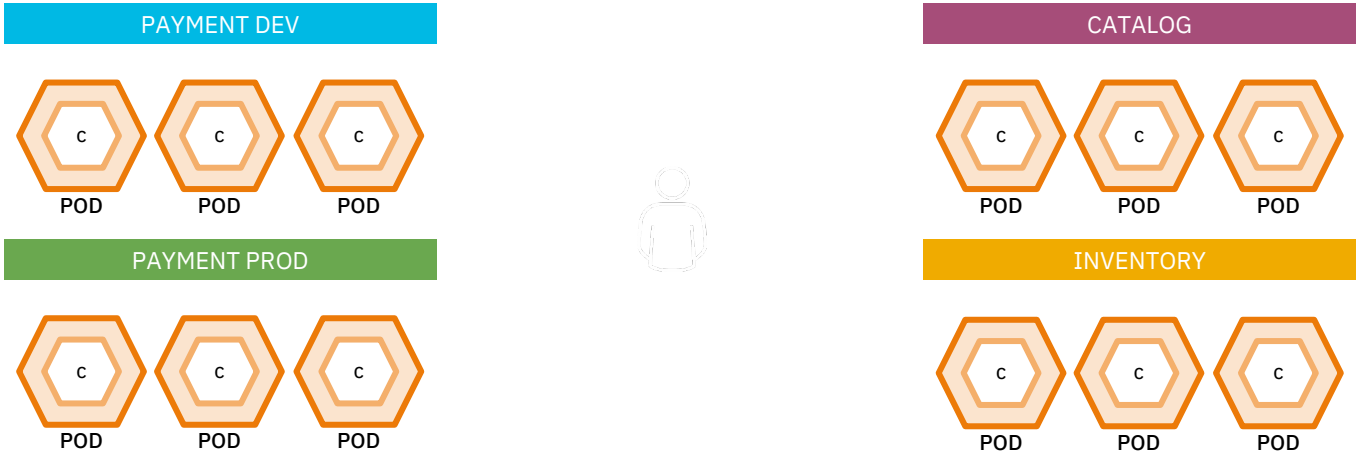


Service chatter

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



Namespaces collate resources and isolate apps across environments, teams, groups and departments.



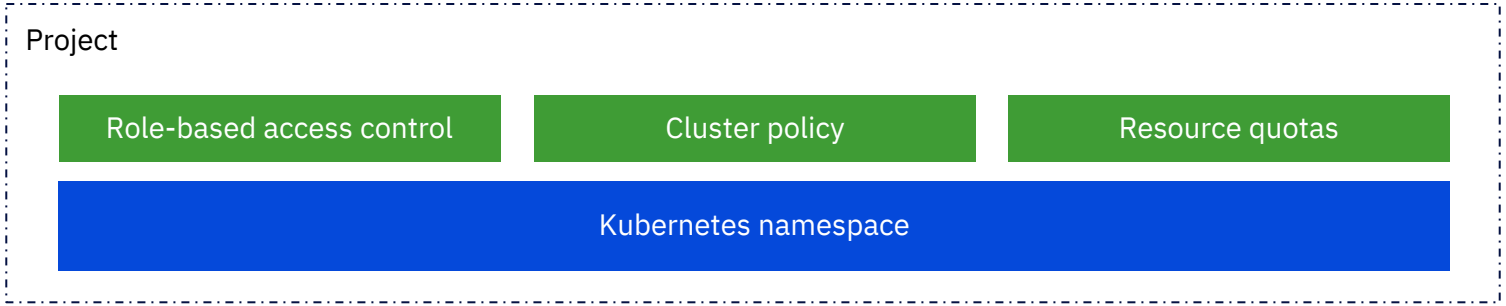
Kubernetes namespaces

Namespaces were designed as a construct for cluster resource management, not security.

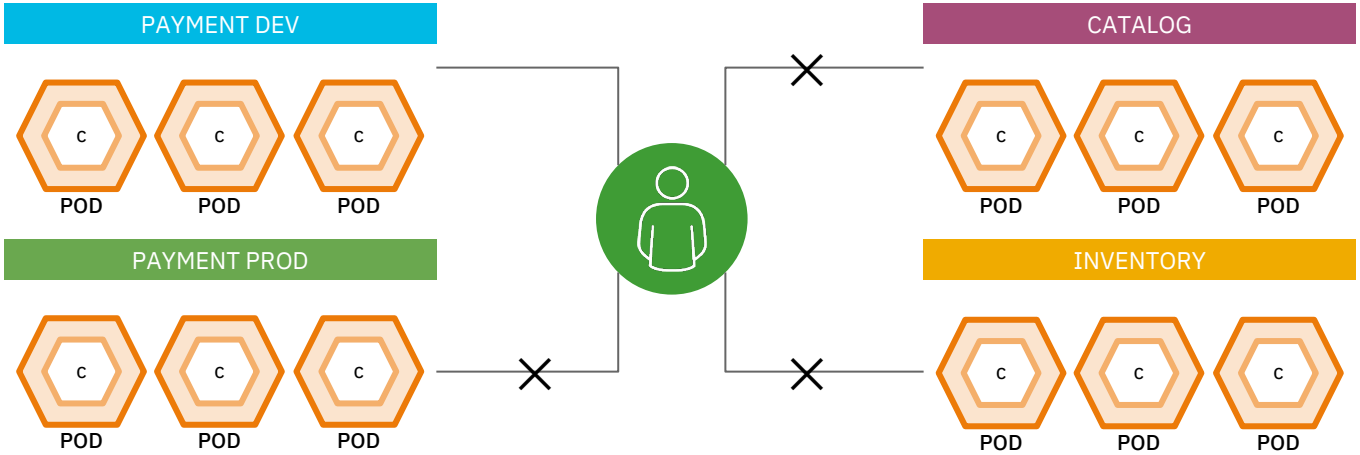
Do not rely on namespaces as a security feature outside of cluster internals within trusted domains.

Do not rely on namespaces to deny a cluster user access to resources in other namespaces.

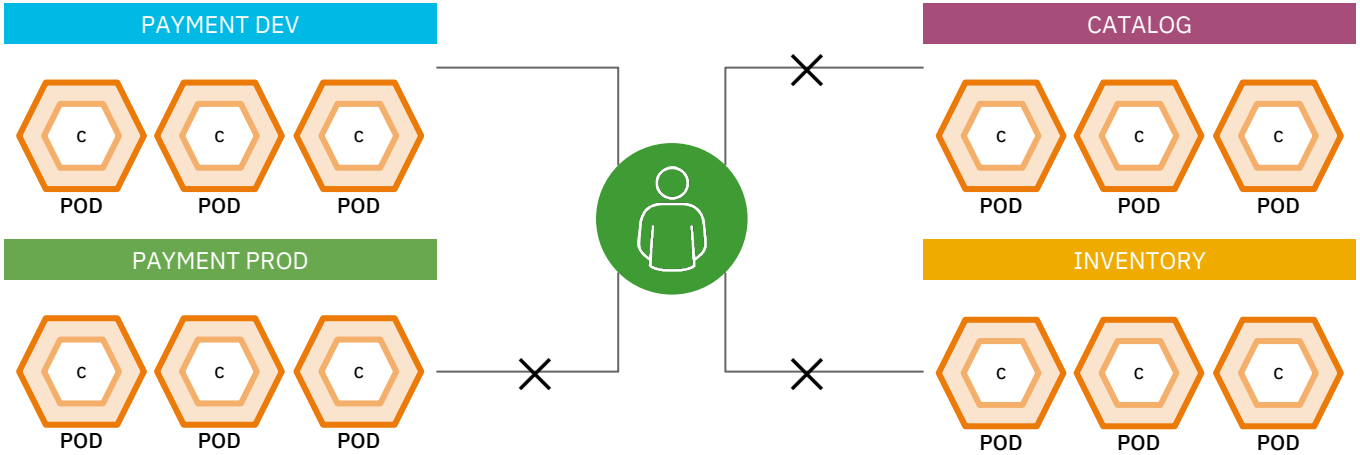
A namespace plus the RBAC layer and some other enhancements is a project



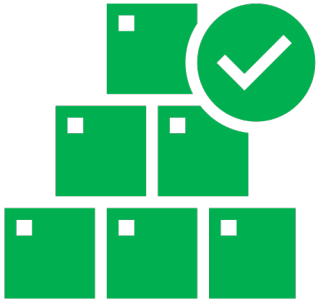
Projects isolate apps across environments, teams, groups and departments

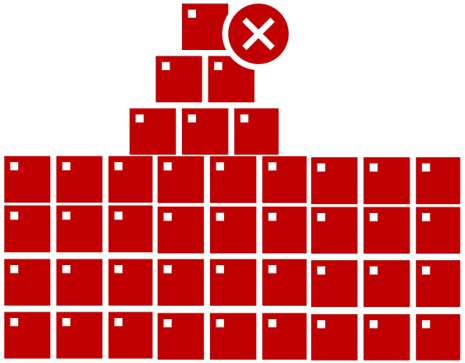


IBM Z and LinuxONE are the only platform where SECURE multi-tenant usage is possible



Embrace projects and use them on a sensible scale. Balance their performance enhancement against operational complexity.





OpenShift 4 Architecture

COMPUTE

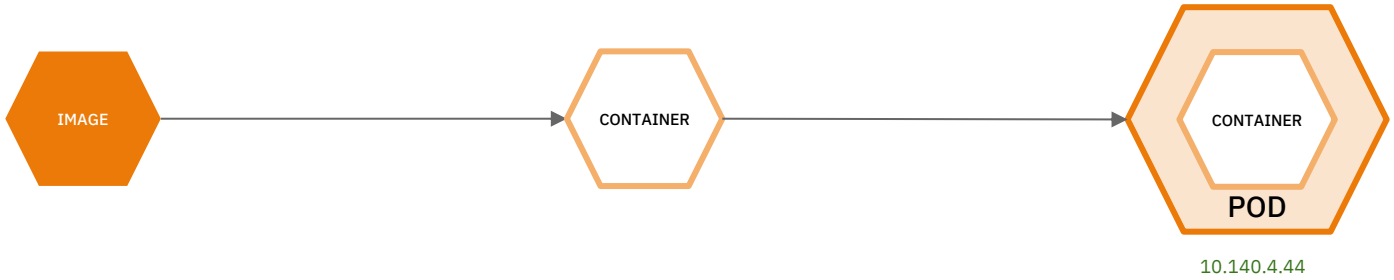
NETWORK

STORAGE

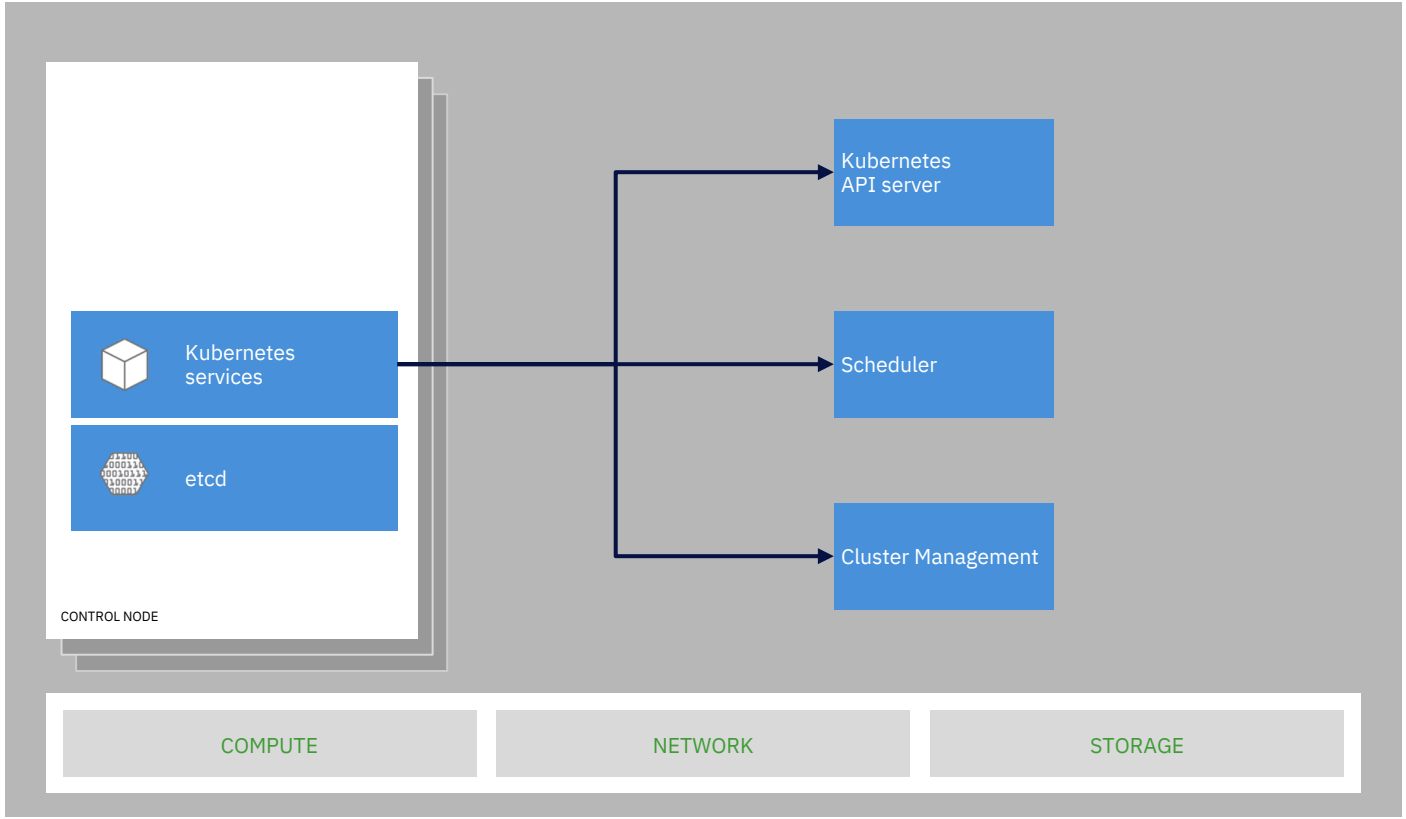
your choice of infrastructure

Compute nodes run workloads

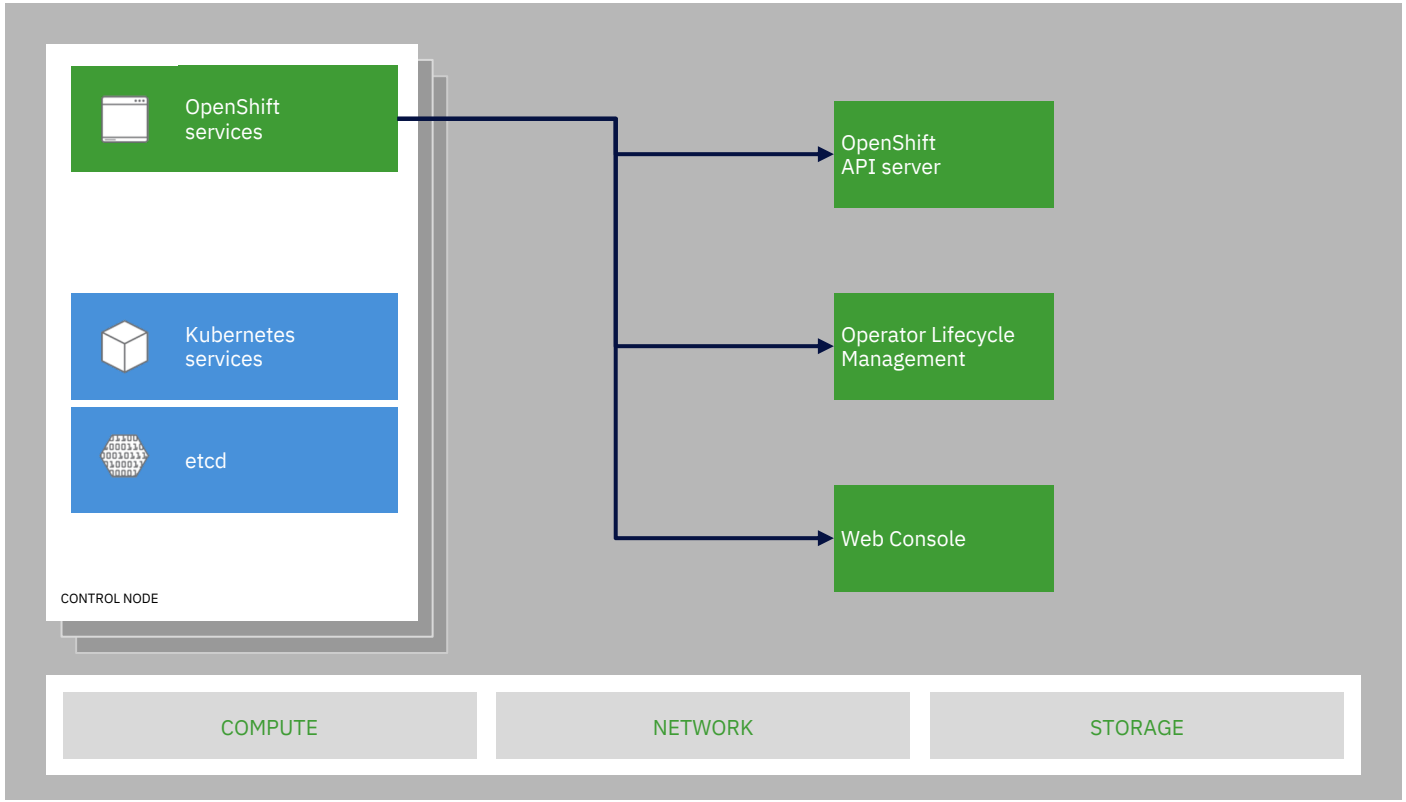
Control nodes



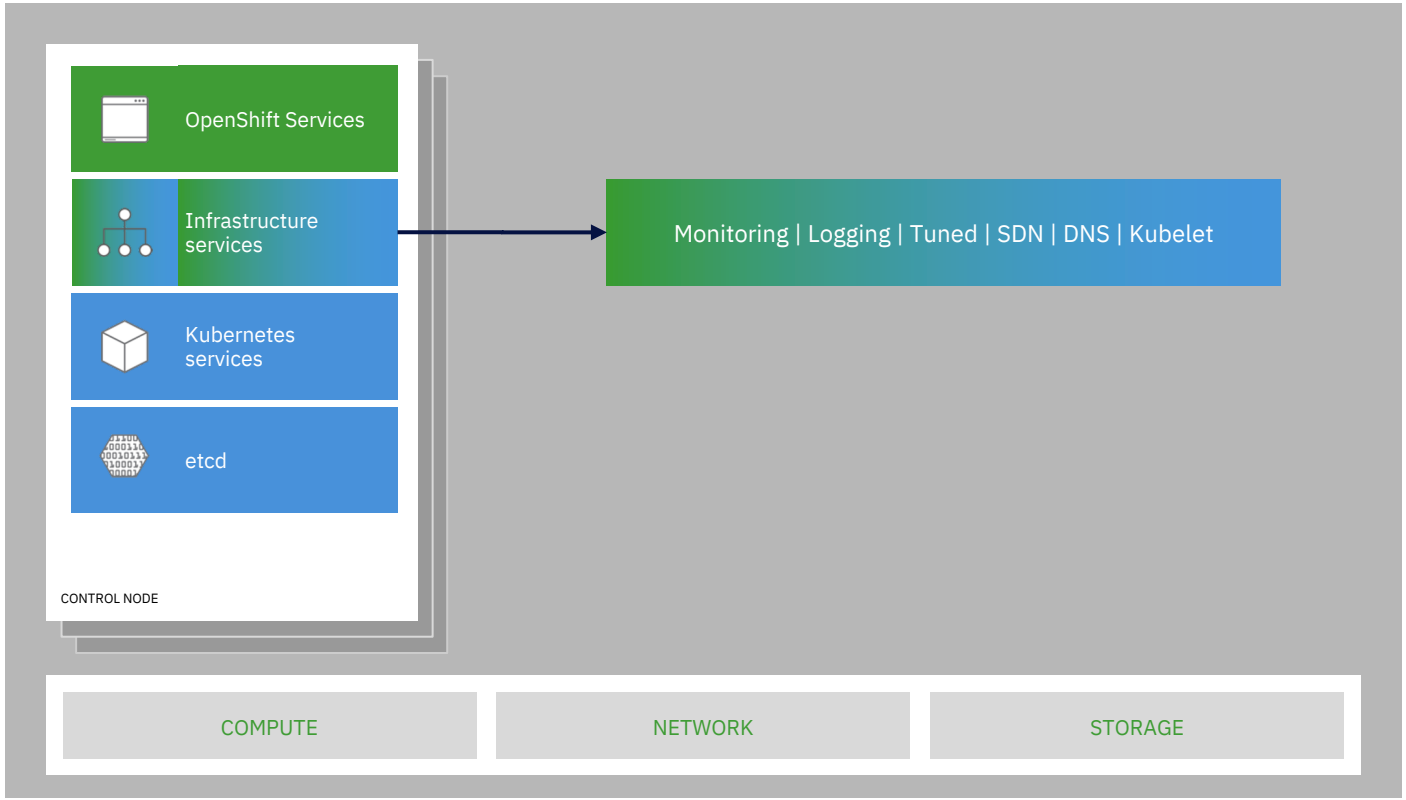
everything runs in pods



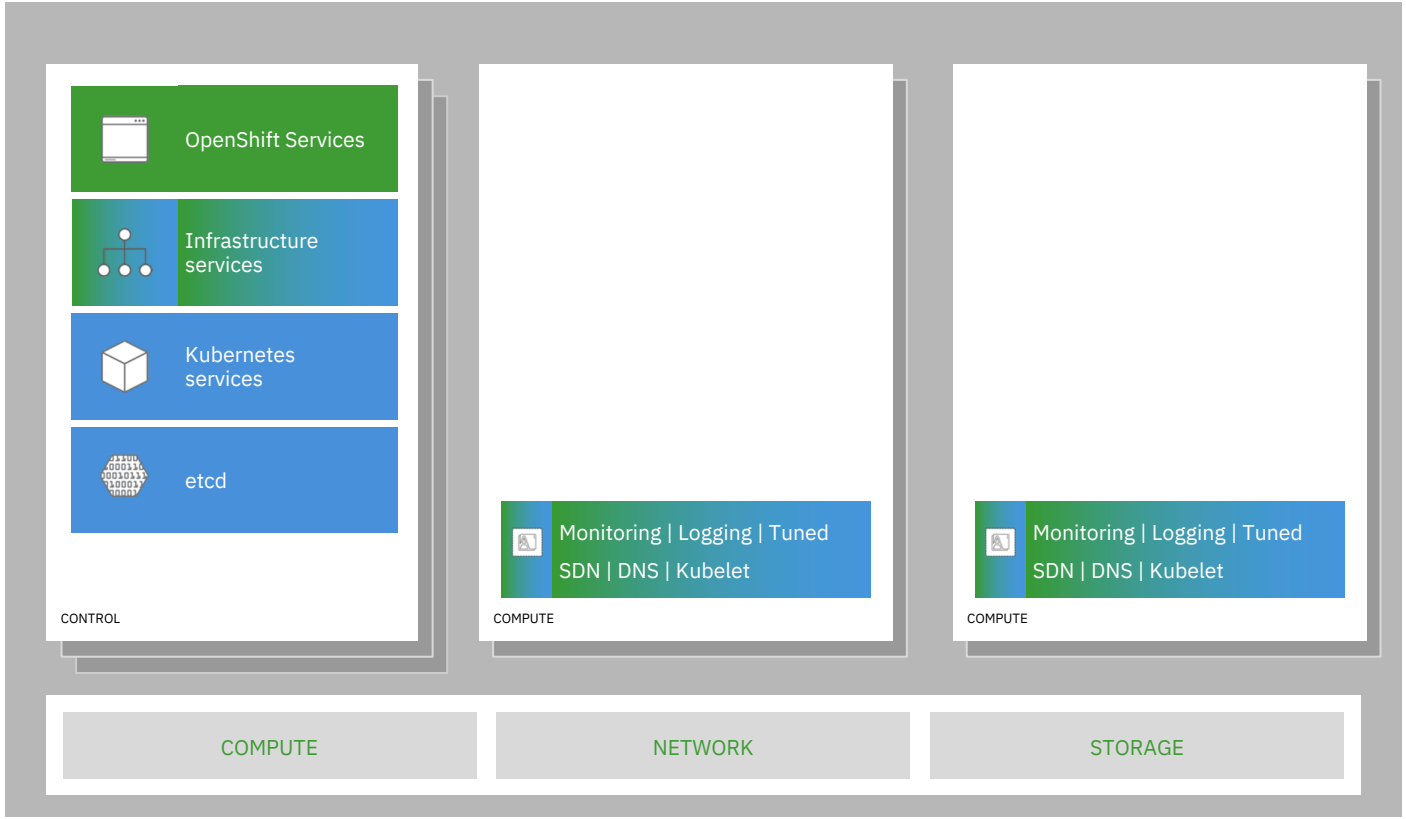
core kubernetes components



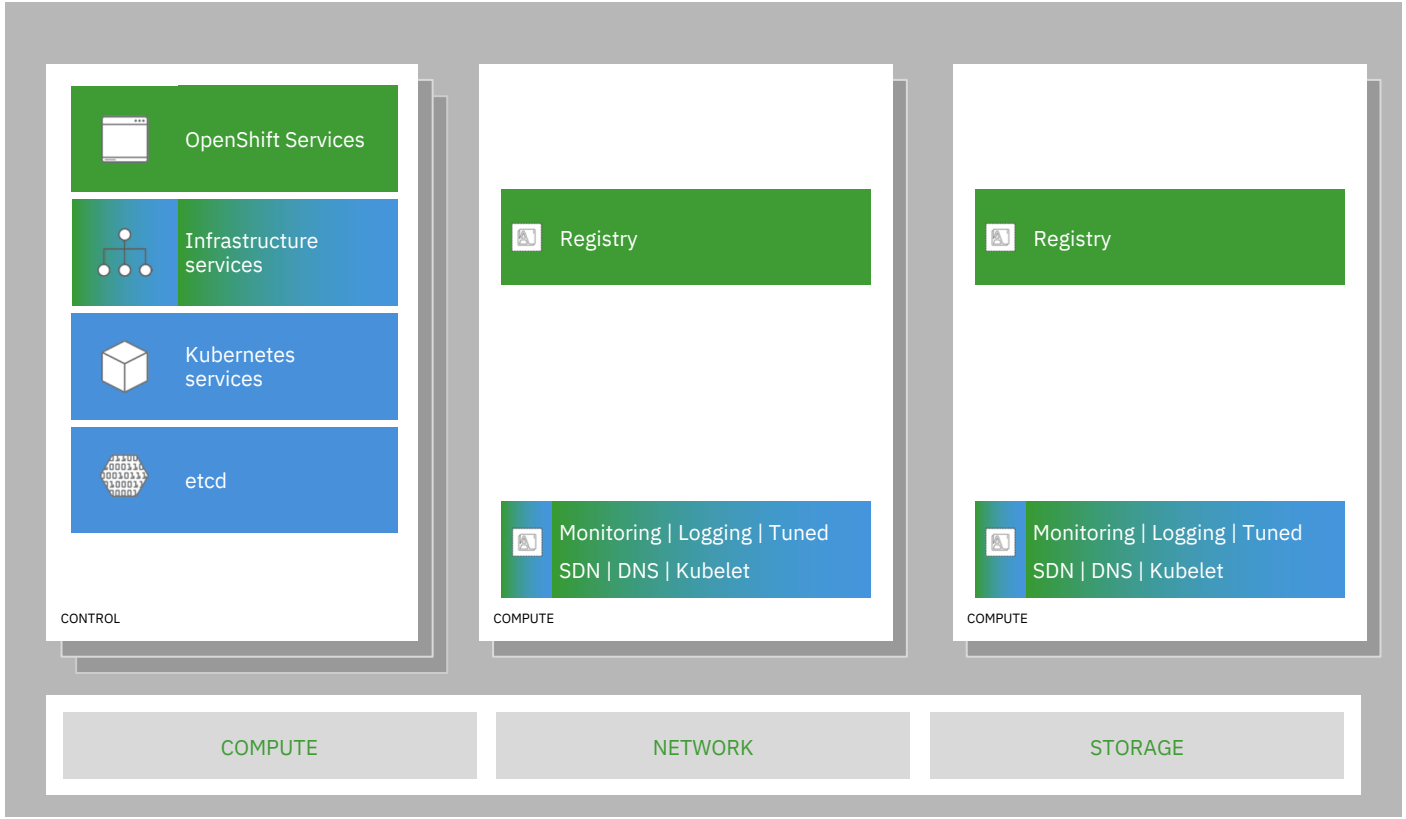
core OpenShift components



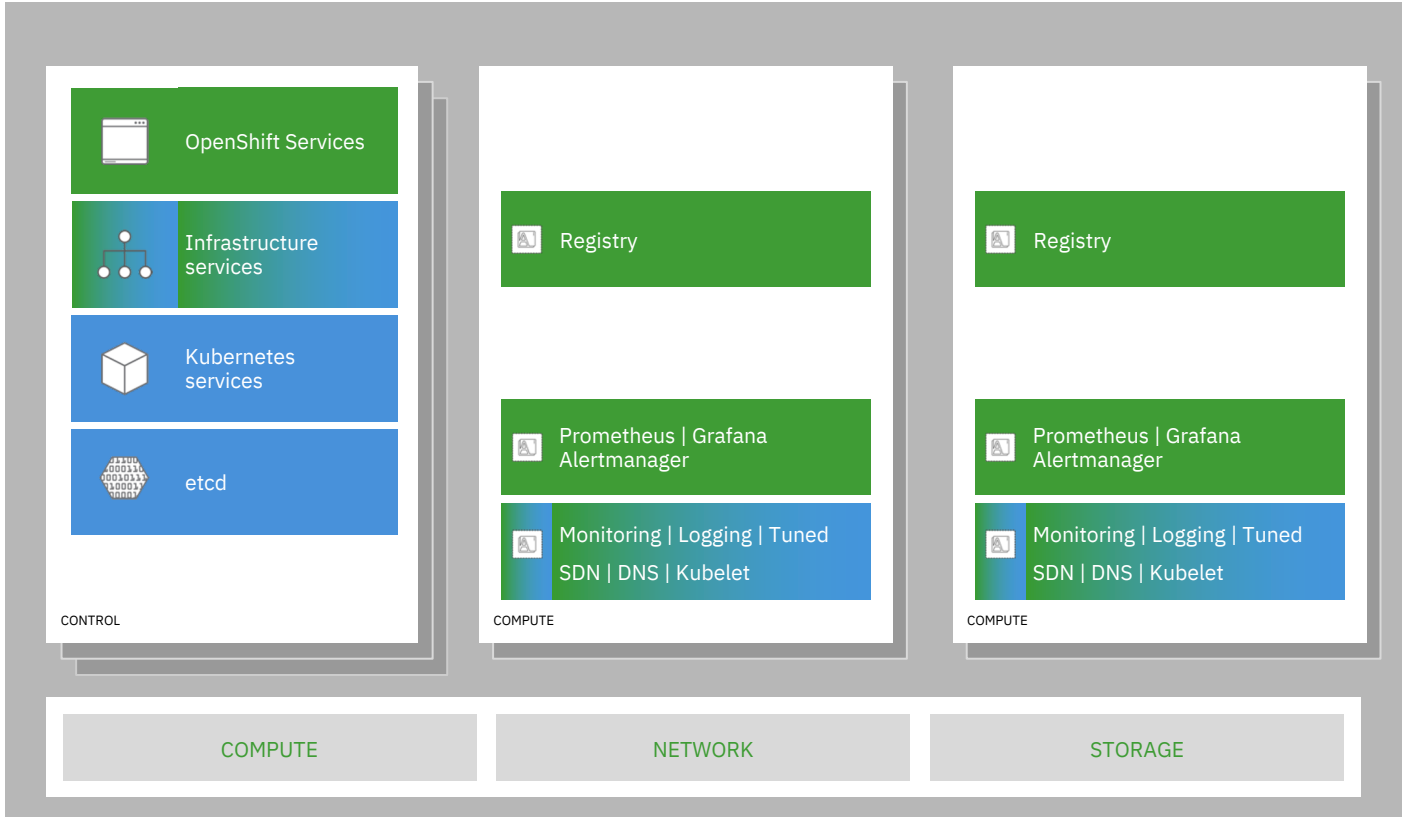
internal and support infrastructure services



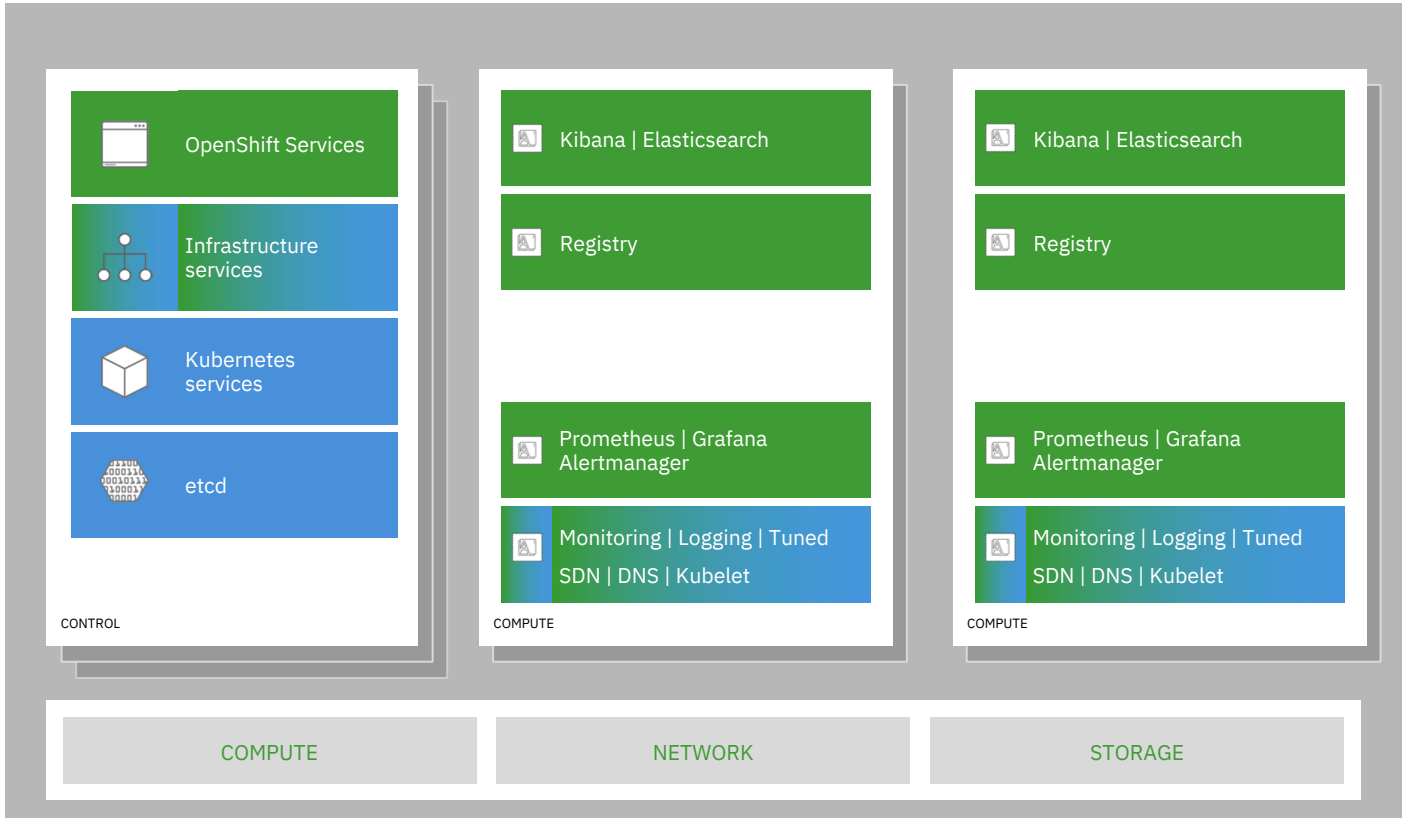
run on all hosts



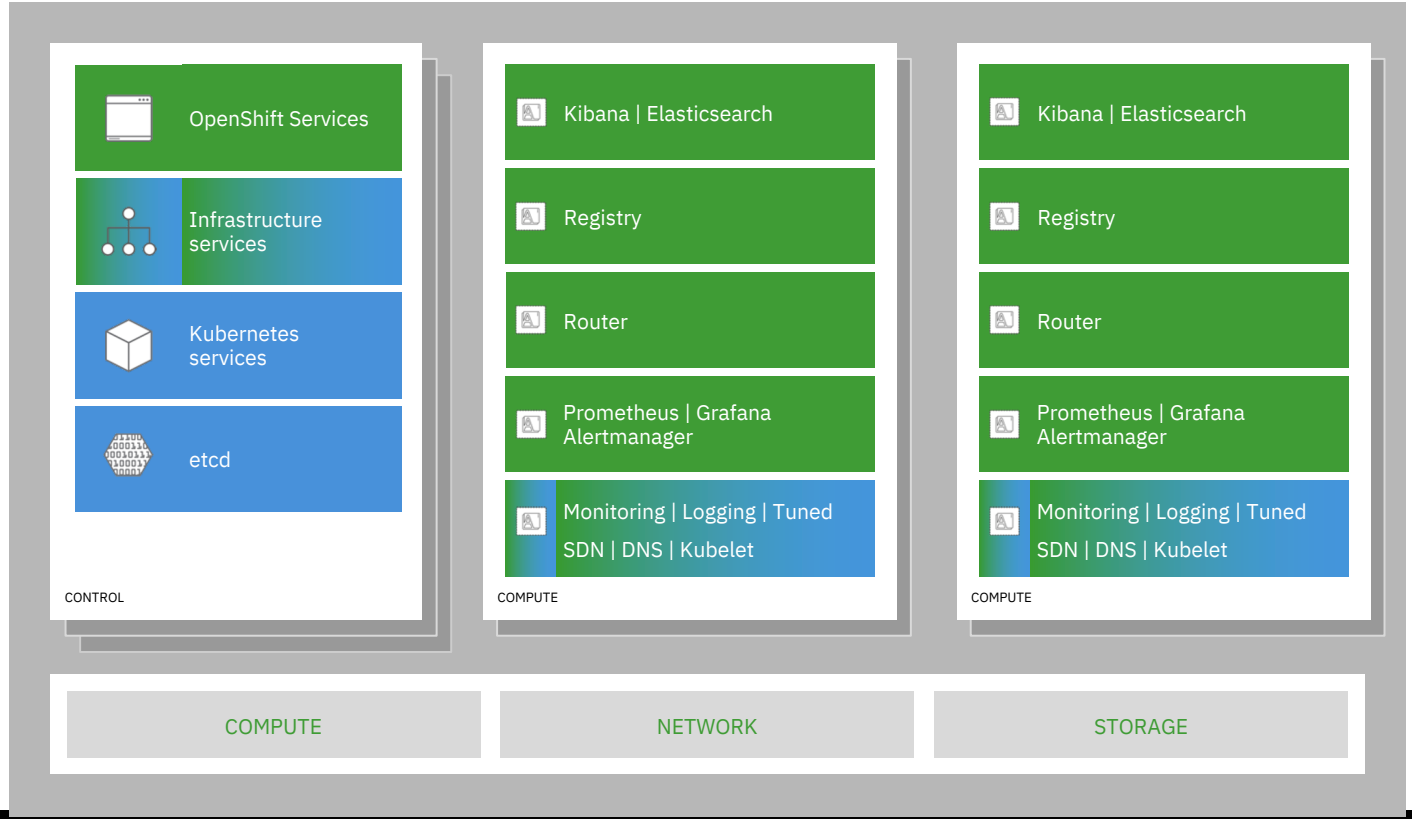
integrated image registry



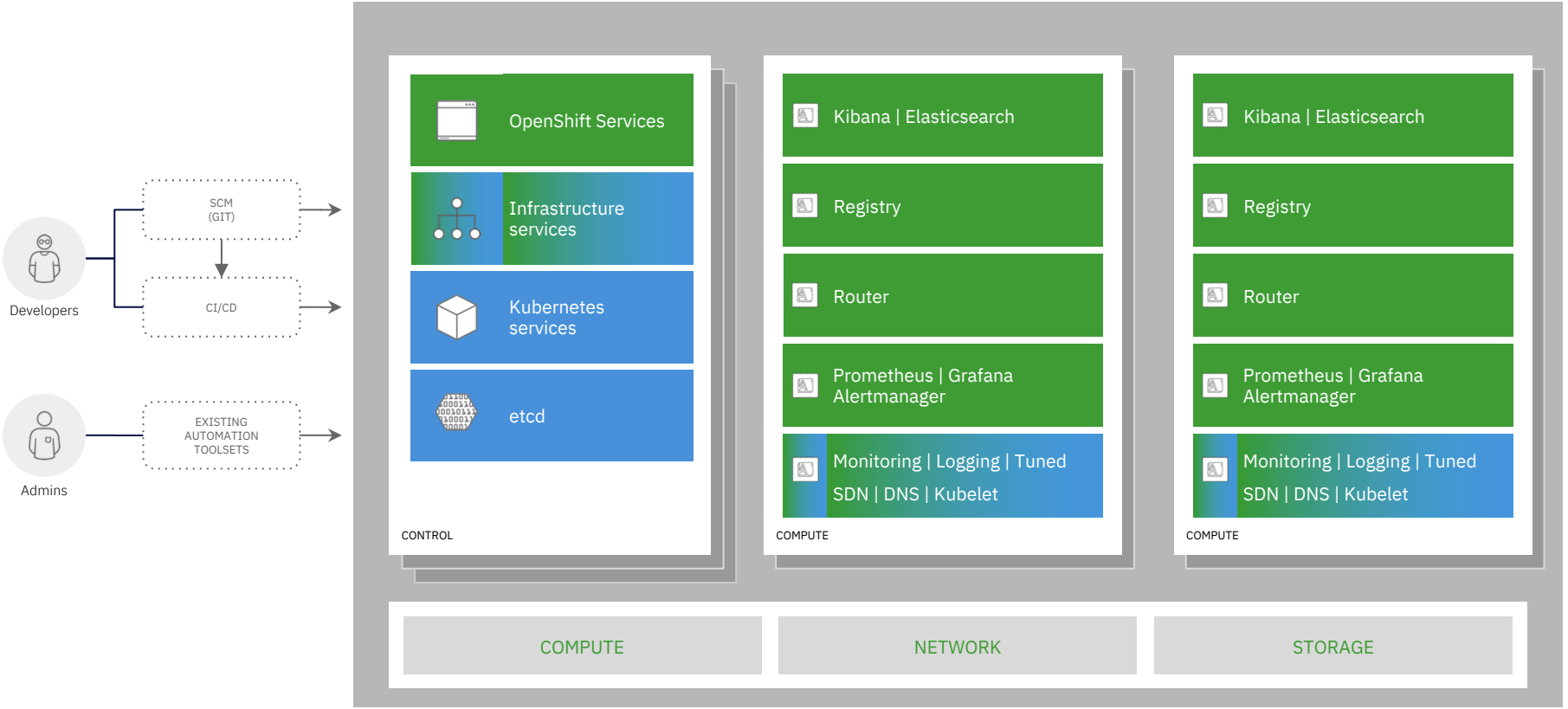
cluster monitoring



log aggregation



integrated routing



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

OpenShift lifecycle, installation & upgrades

Two new paradigms for deploying clusters

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

IBM Cloud Red Hat OpenShift

Get a powerful cluster in the IBM Cloud, fully managed by IBM engineers and support.

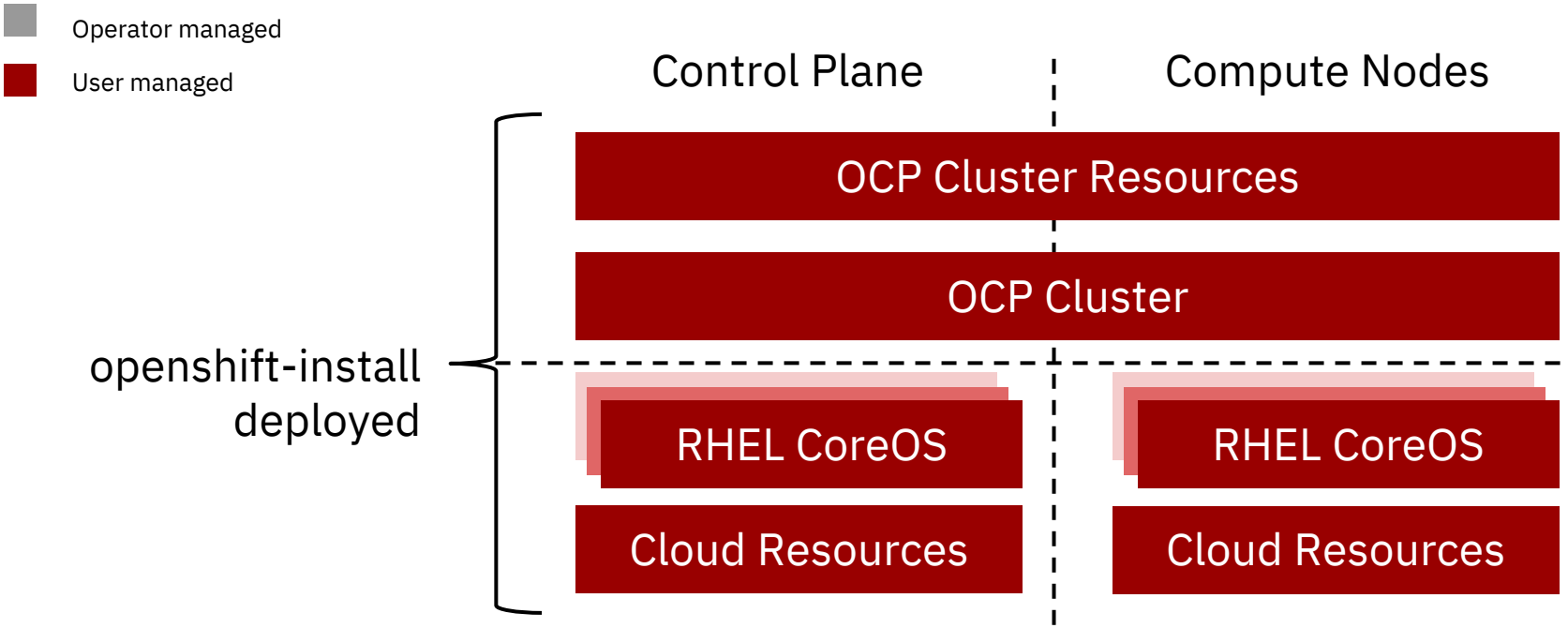
Azure Red Hat OpenShift

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

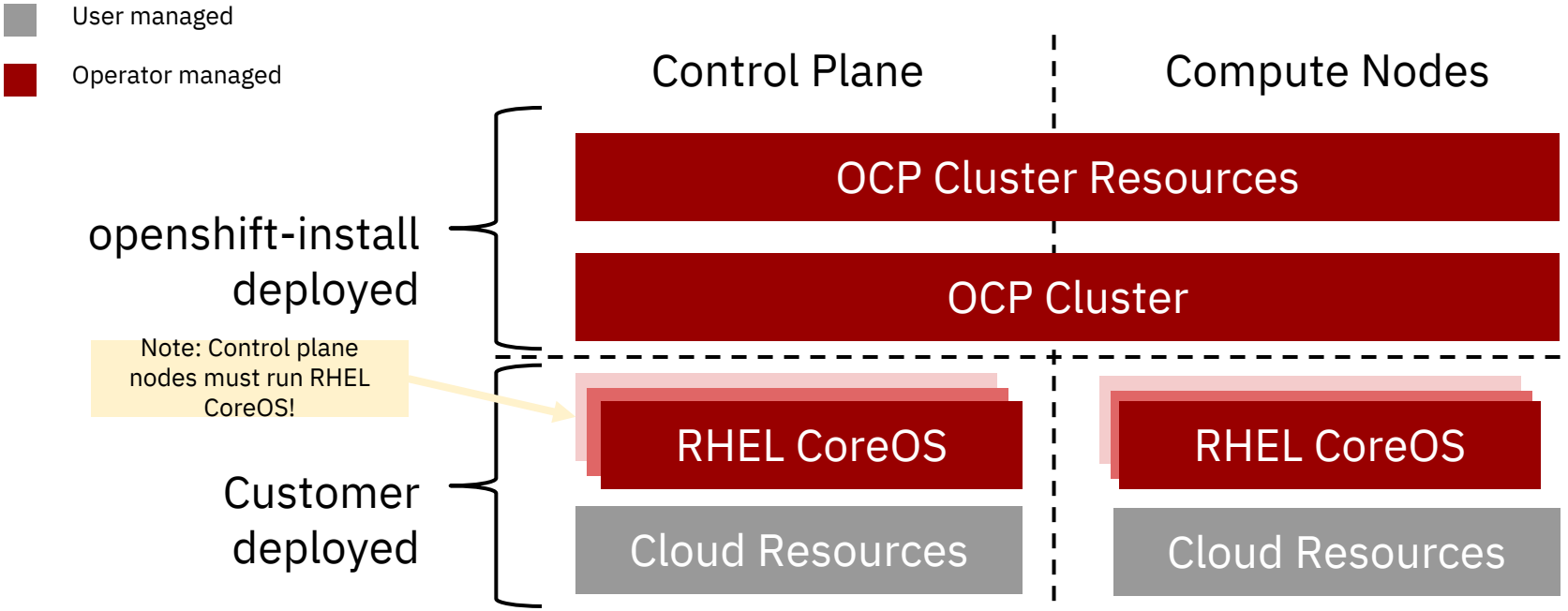
OpenShift Dedicated

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

Installation Paradigms



Full-stack Automated Installation



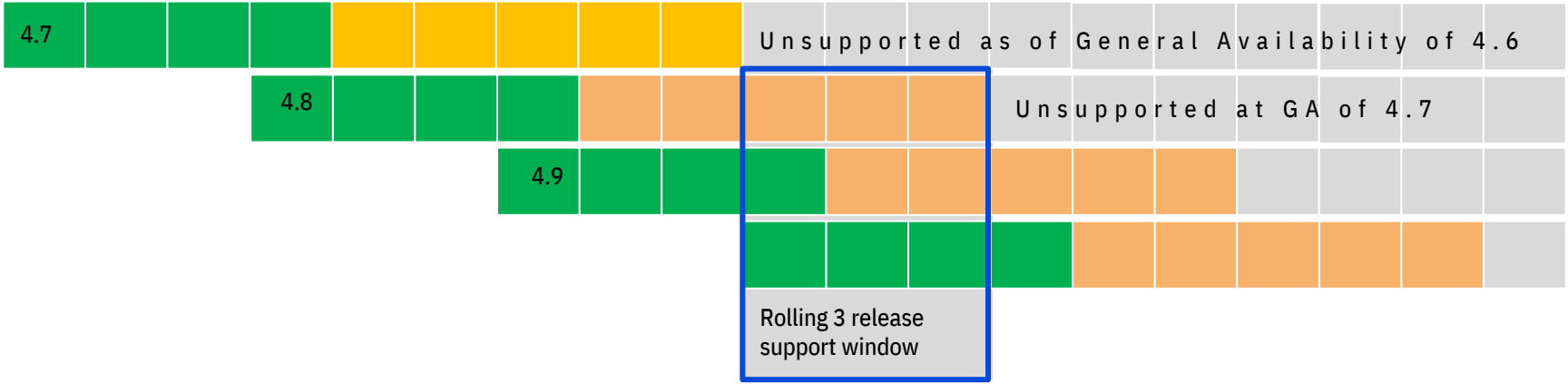
Pre-existing Infrastructure Installation

	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
Node Provisioning / Autoscaling	Yes	Only for providers with OpenShift Machine API support

Comparison of Paradigms

Supported paths for upgrades and migrations

* 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

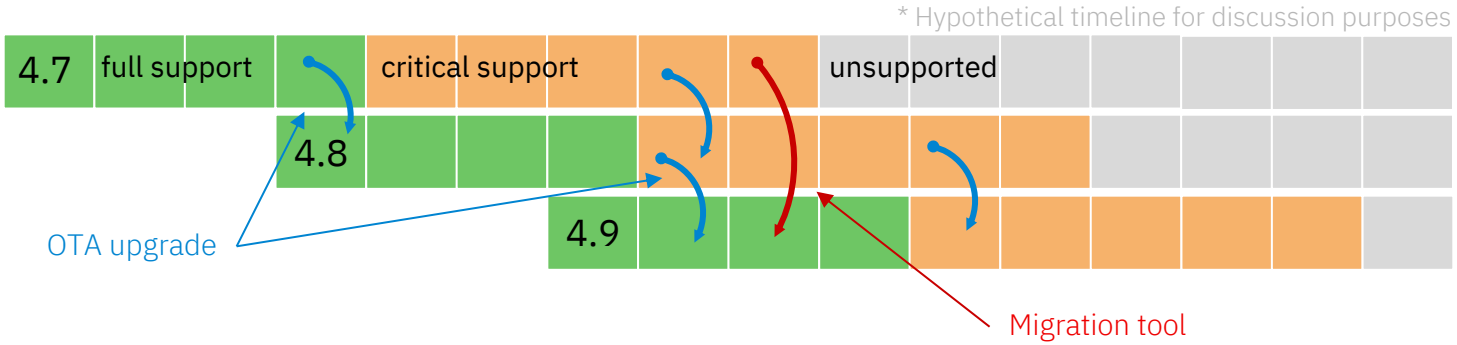
- Minimum two years full support (likely more)
- One year maintenance past the end of full support

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

https://access.redhat.com/support/policy/updates/openshift#ocp4_phases

Support Timelines



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.

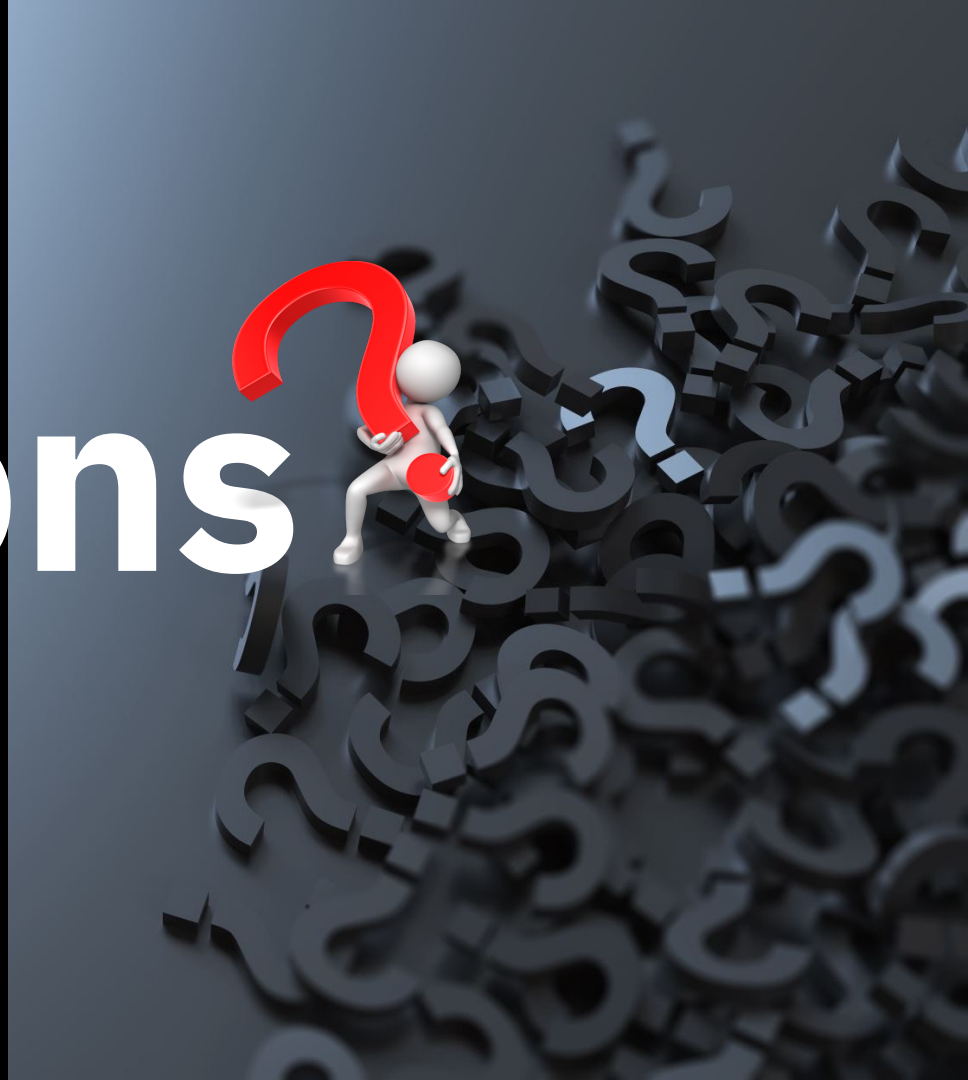
Optional path = migration tooling
If you fall more than two releases behind, you must use the application migration tooling to move to a new cluster.

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

Upgrades vs. Migrations

Questions?



Operations and infrastructure deep dive

Red Hat Enterprise Linux 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.

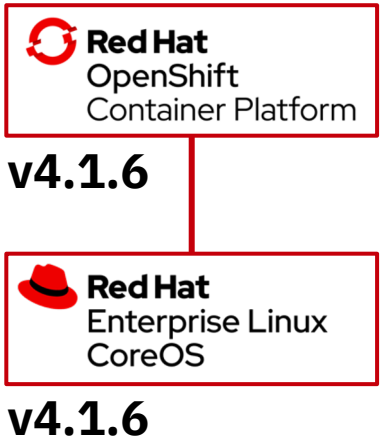
Red Hat Enterprise Linux CoreOS is managed by the cluster

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. 😀🙌



More about CoreOS

Minimal and Secure Architecture

Optimized for Kubernetes

Runs any OCI-compliant image (including docker)

A lightweight, OCI-compliant container runtime

CRI-O Support in OpenShift

CRI-O 1.16 	Kubernetes 1.16 	OpenShift 4.6 
CRI-O 1.17 	Kubernetes 1.17 	OpenShift 4.7 
CRI-O 1.18 	Kubernetes 1.18 	OpenShift 4.8 

Broad ecosystem of workloads

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

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

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. Control machines fetch the remote resources from the bootstrap machine and finish booting.
3. Control 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.

OpenShift Bootstrap Process: Self-Managed Kubernetes

Controls (Special)

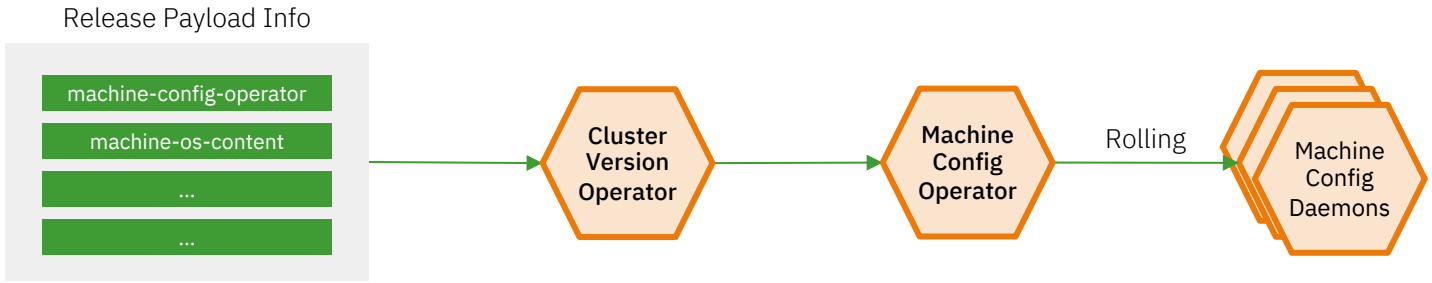
- Terraform provisions initial masters on public cloud / full-stack automated installs. BaNWIS / Bastion on Z and LinuxONE
- Machine API adopts existing masters post-provision
- Each master is a standalone Machine object
- Termination protection (avoid self-destruction)

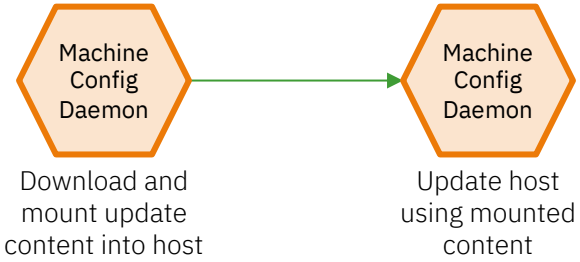
Computes

- Each Machine Pool corresponds to MachineSet
- Optionally autoscale (min,max) and health check (replace if not ready > X minutes) on public cloud. This is not much of a concern on Z and LinuxONE because hardware failure is such a remote possibility.

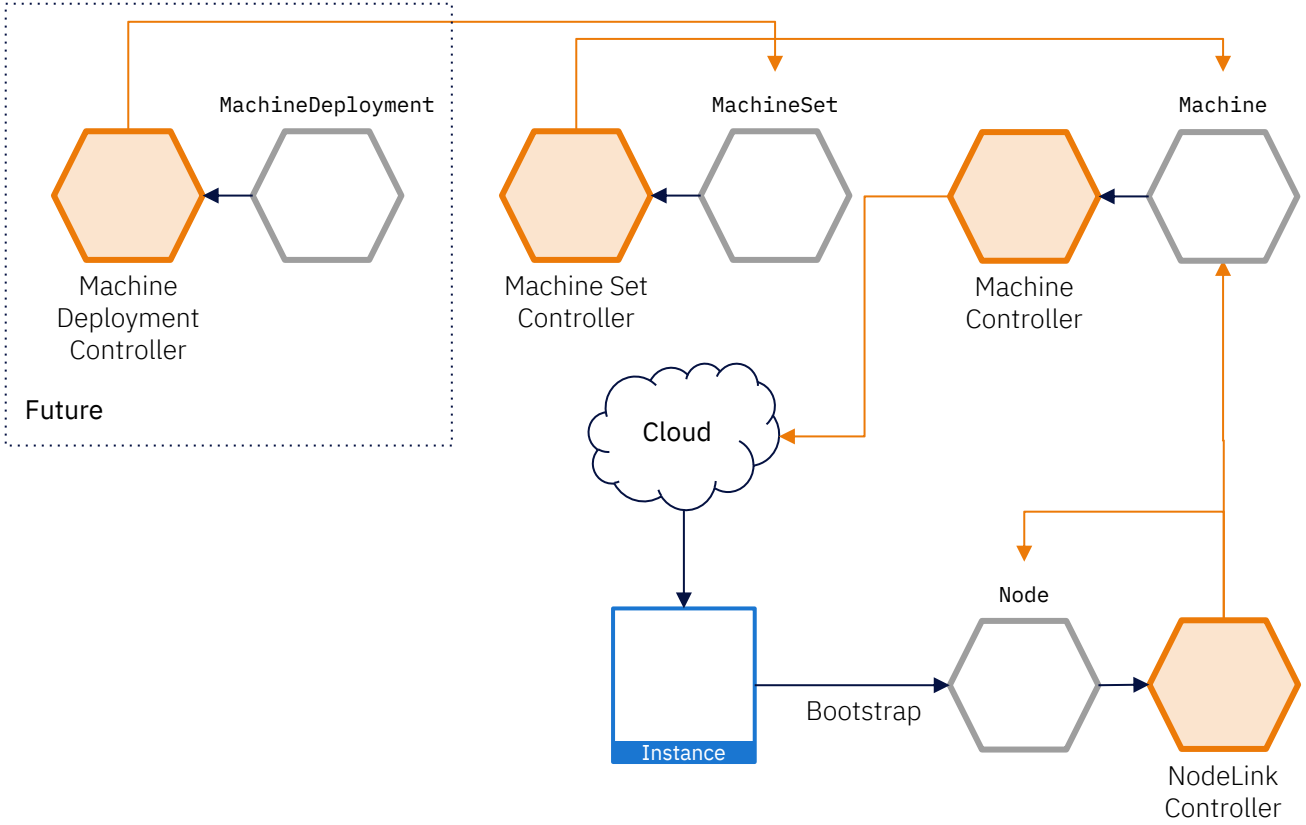
How everything deployed comes under management

Powered by Operators, OpenShift 4 automates many cluster management activities





Over-the-air updates



Cloud API

Features, mechanisms and processes for container and platform isolation

CONTROL
Application Security

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

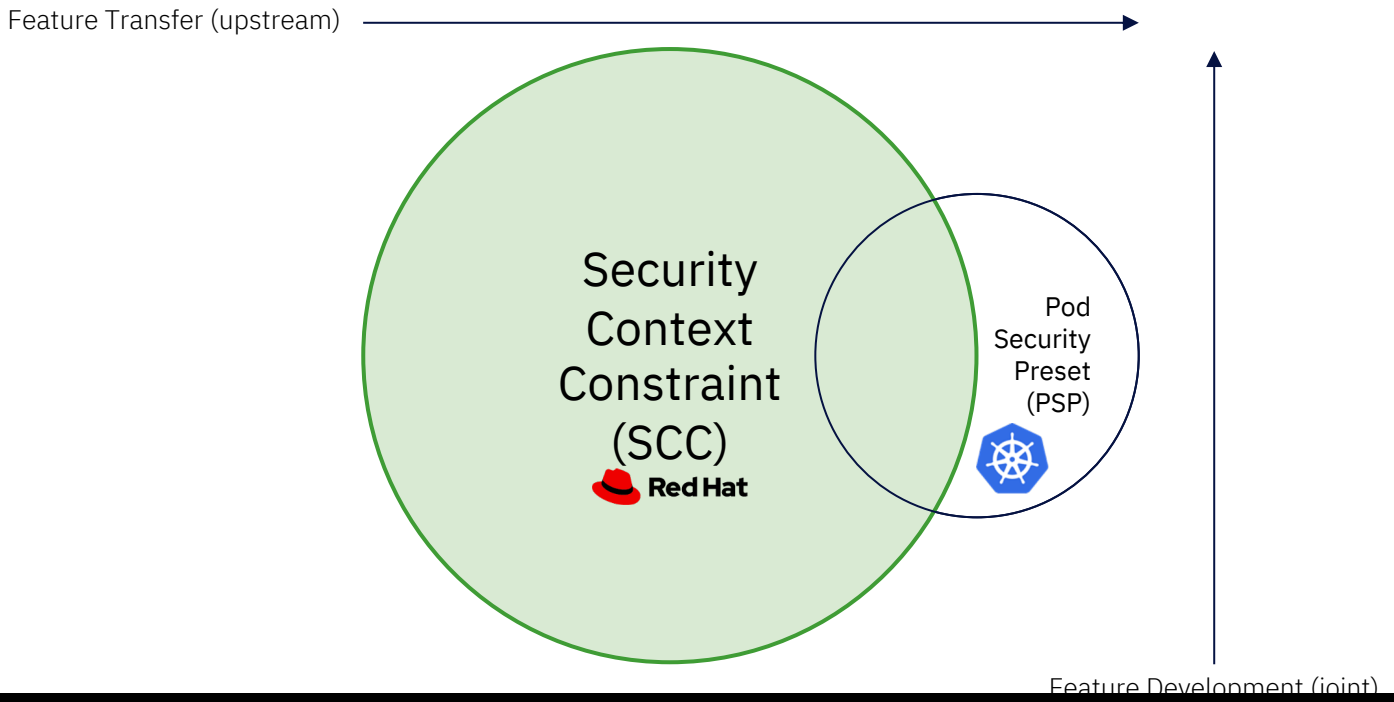
DEFEND
Infrastructure

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

EXTEND

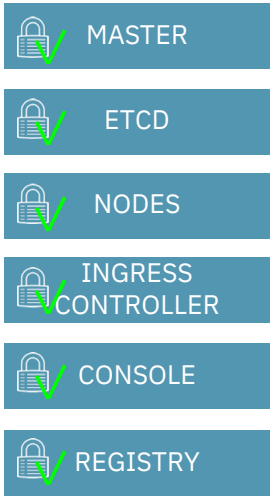
- Security Ecosystem

Security is built in

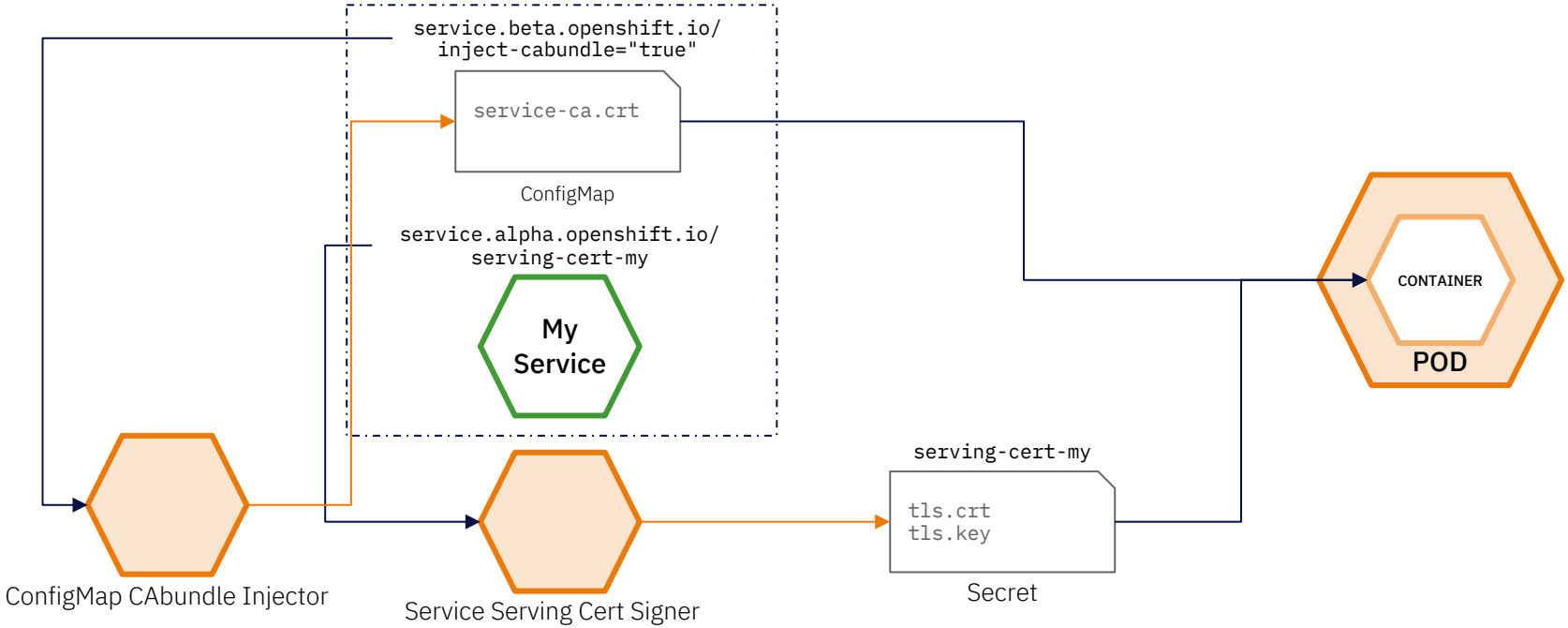


Extended Depth of Protection

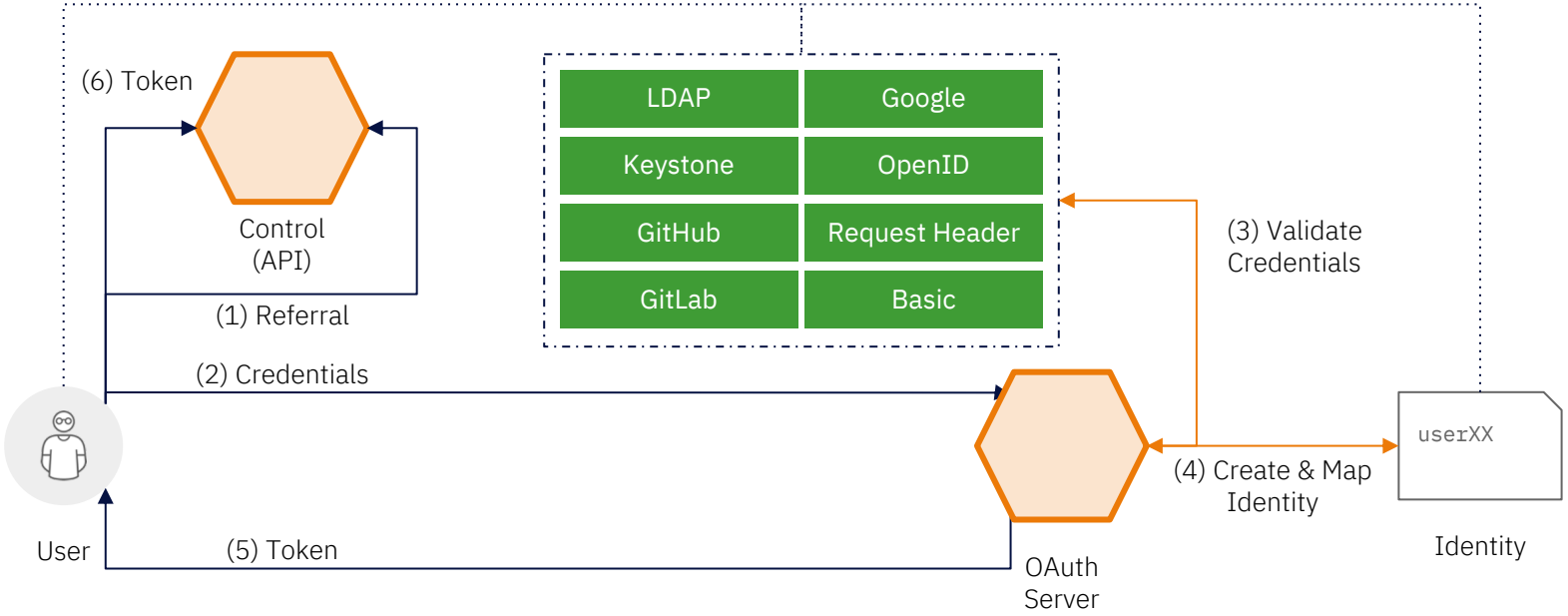
- OpenShift provides its own internal CA
- Certificates are used to provide secure connections to
  - master (APIs) and nodes
  - Ingress controller and registry
  - etcd
- Certificate rotation is automated



Certificates and Certificate Management



Service Certificates



Identity and Access Management

- 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

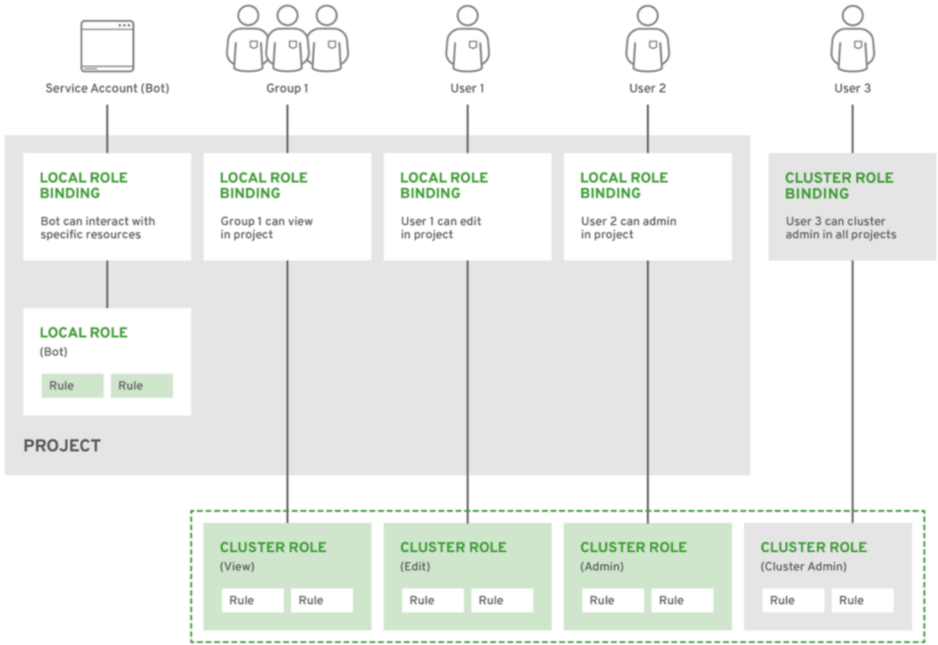


Figure 12 - Authorization Relationships

Fine-Grained RBAC

An integrated cluster monitoring and alerting stack

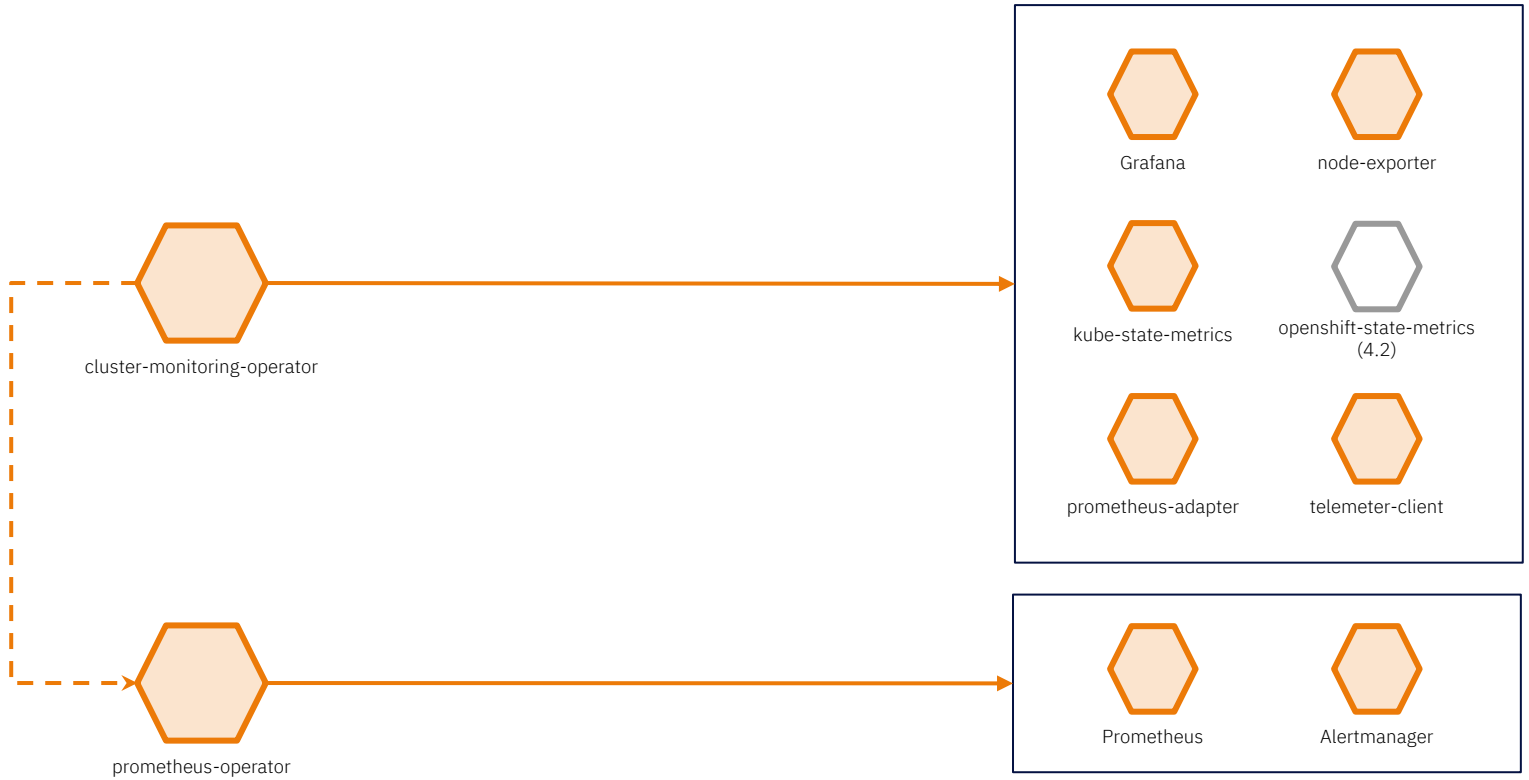
OpenShift Cluster Monitoring

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

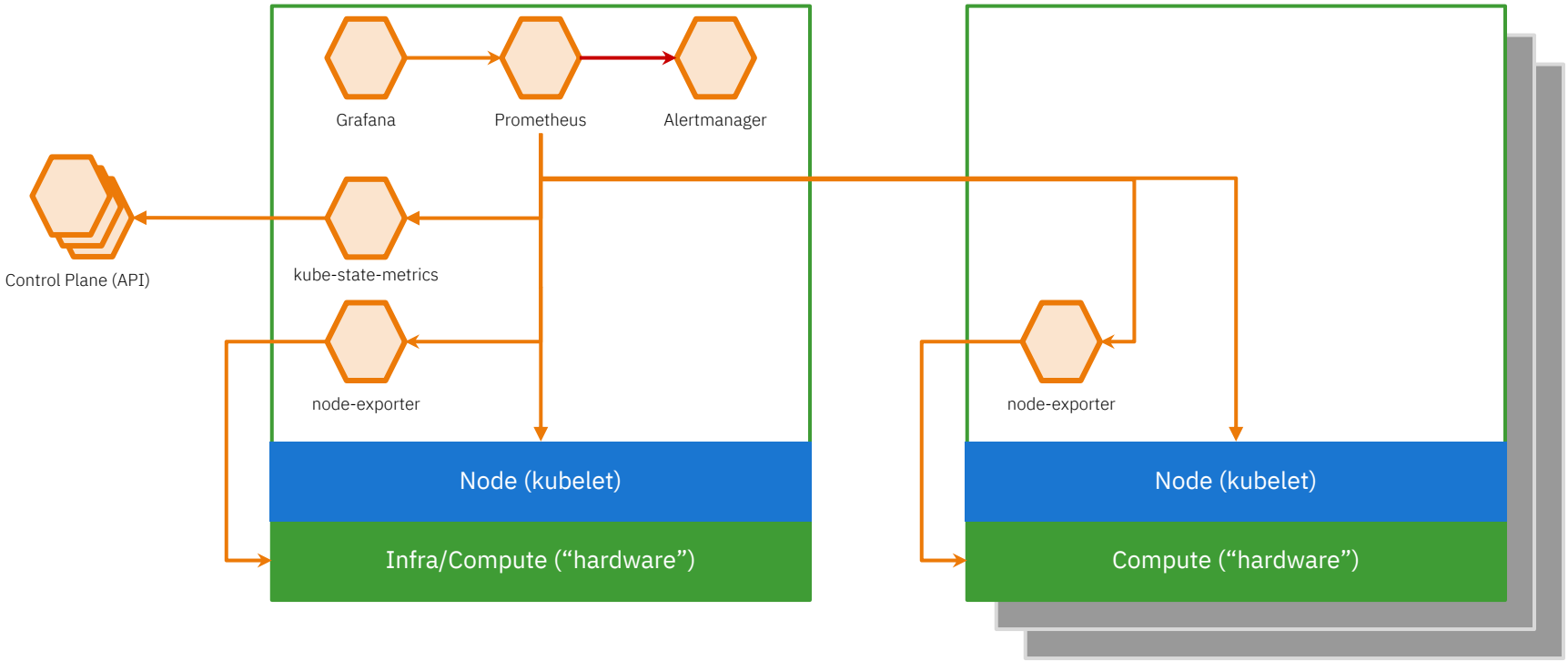
Alerting/notification via Prometheus' Alertmanager, an open-source tool that handles alerts send by

Metrics visualization via Grafana, the leading metrics visualization technology.

The monitoring bundle



Monitoring relationships



The “plumbing”

An integrated solution for exploring and corroborating application logs

Components

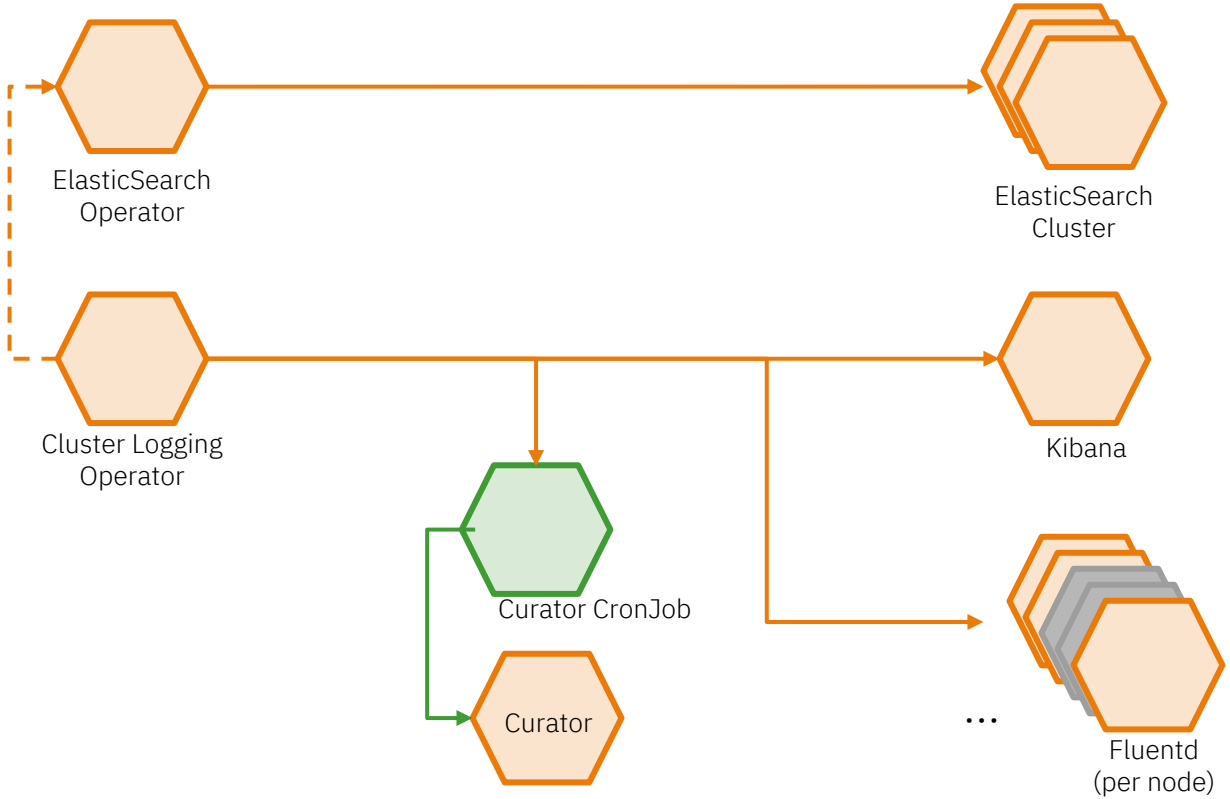
- **Elasticsearch:** a search and analytics engine to store logs
- **Fluentd:** gathers logs and sends to Elasticsearch.
- **Kibana:** A web UI for Elasticsearch.

Access control

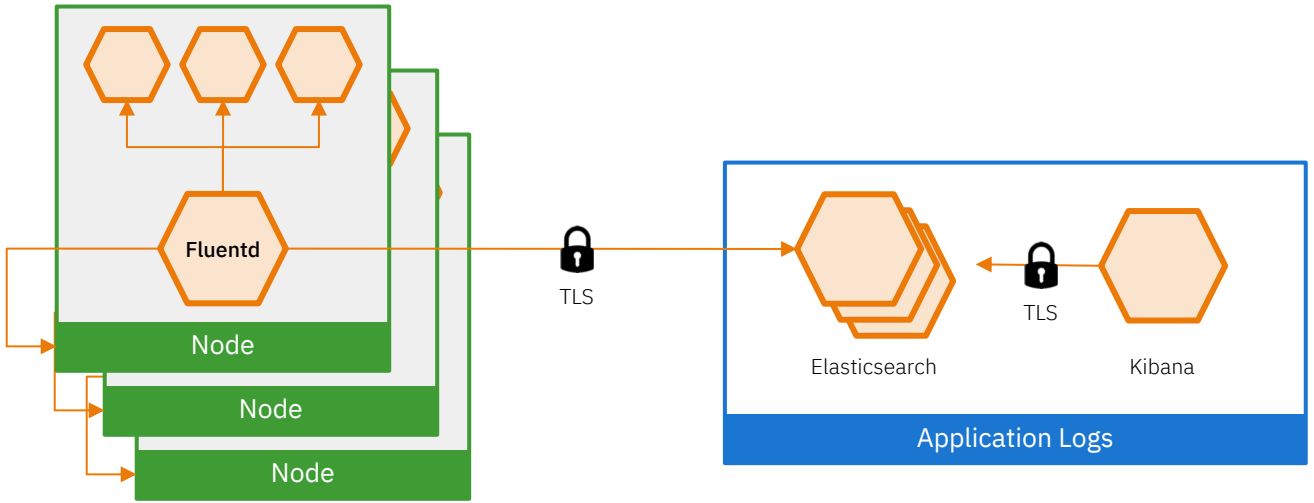
- Cluster administrators can view all logs
- Users can only view logs for their projects

Ability to forward logs elsewhere

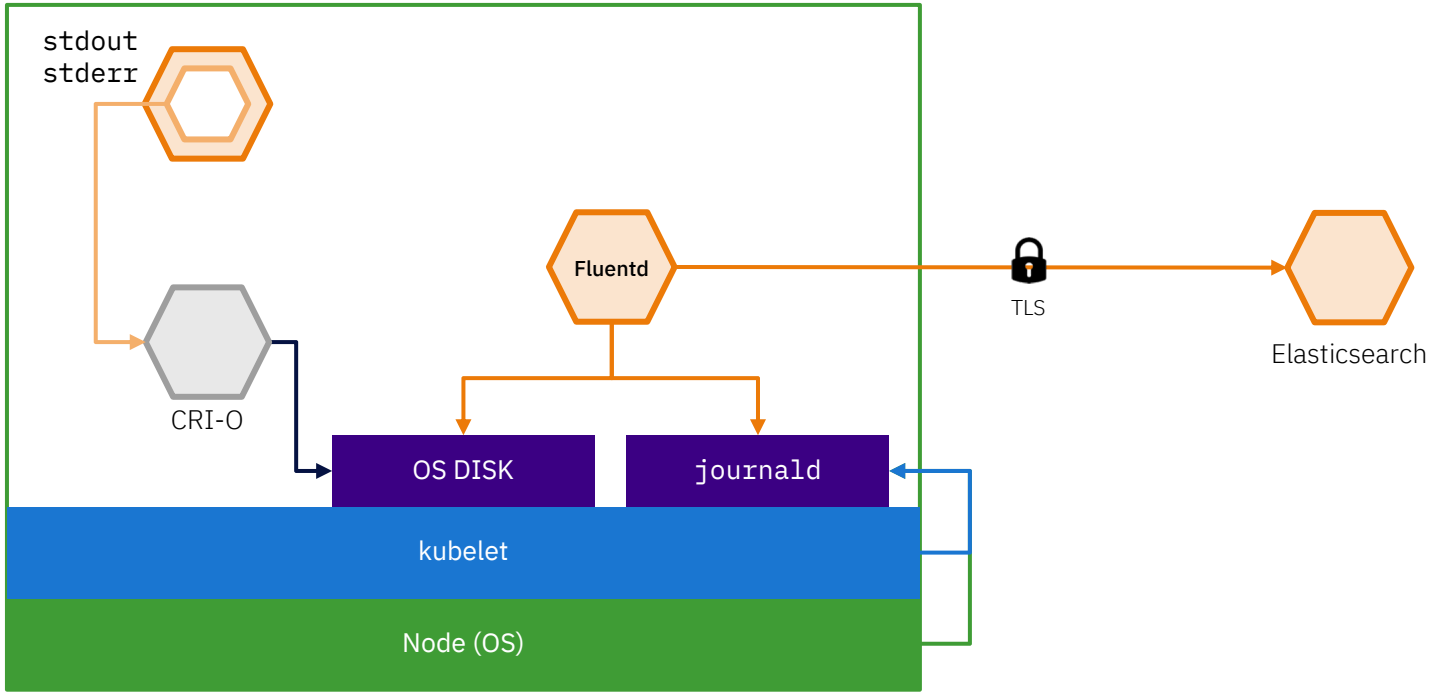
- External elasticsearch, Splunk, etc



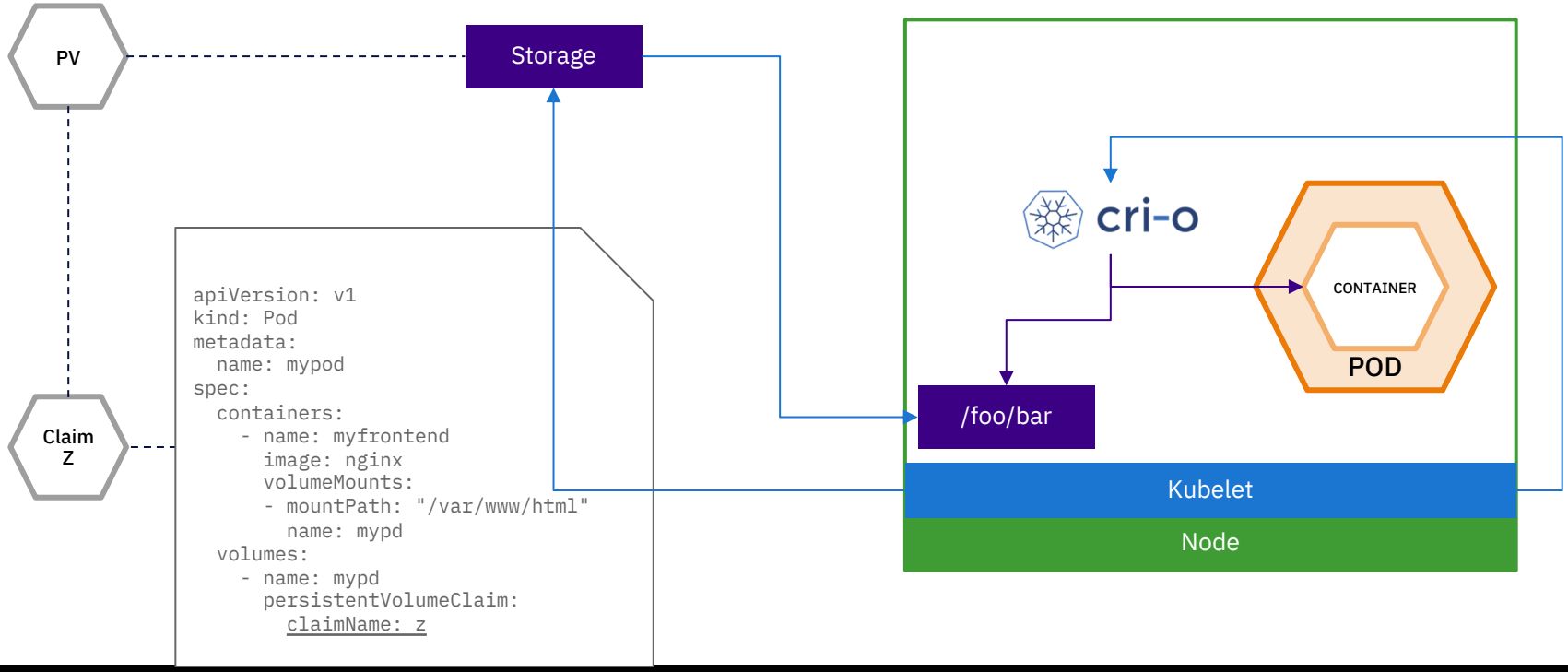
Relationships for logging



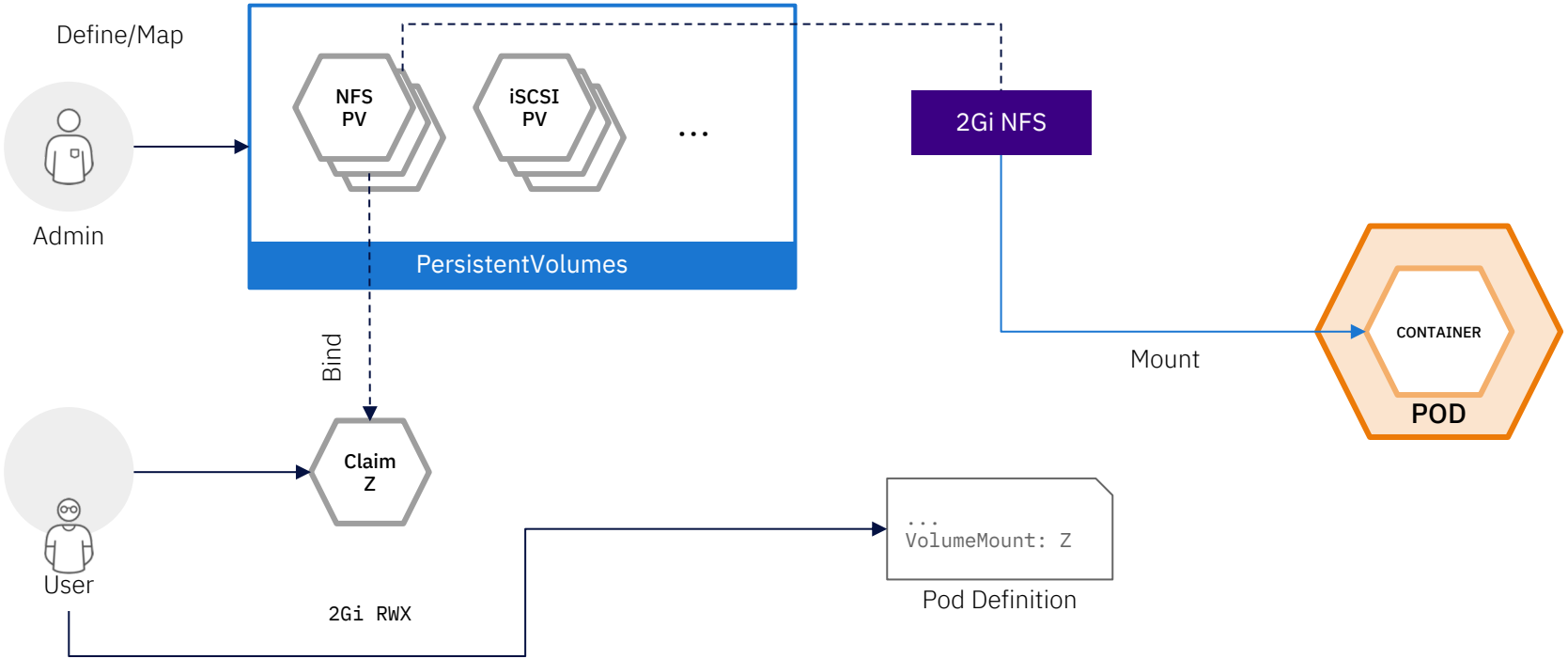
Log data flow in OpenShift



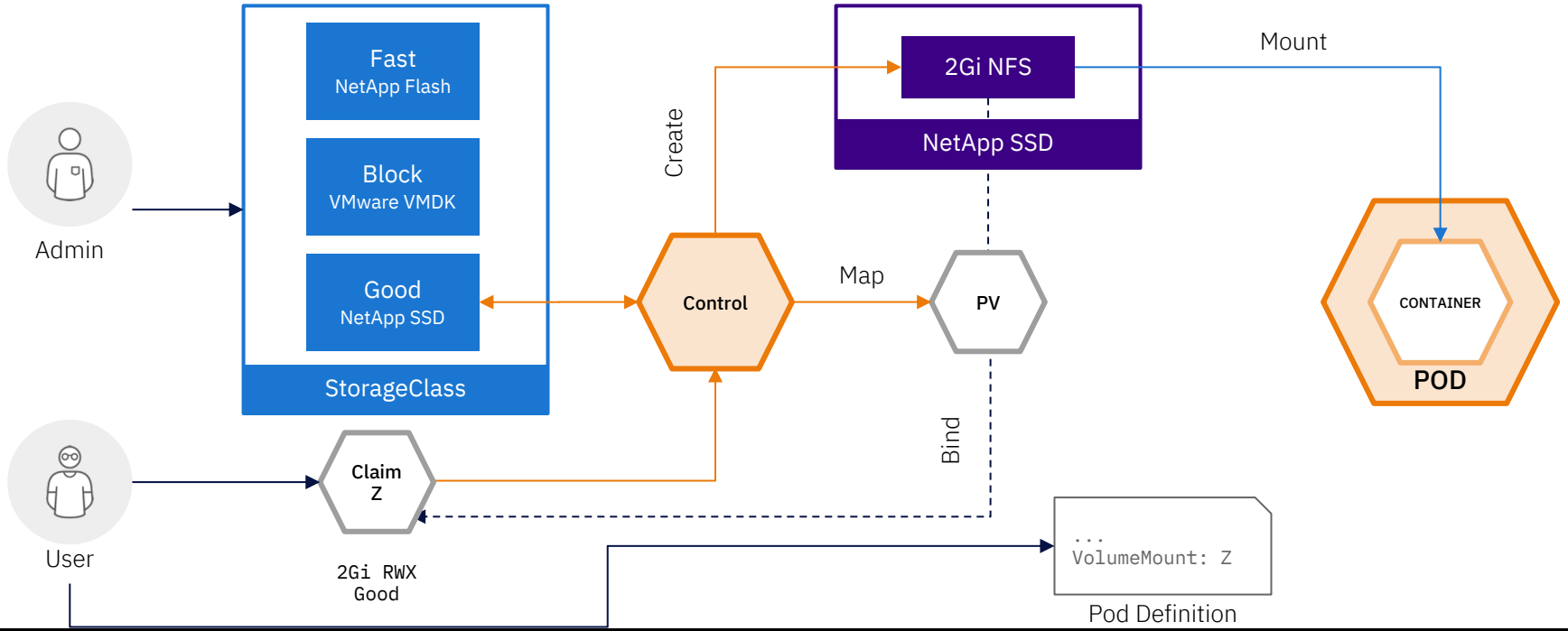
Log data flow in OpenShift



PV Consumption

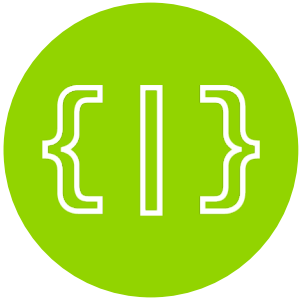


Static Storage Provisioning



Dynamic Storage Provisioning

Tools and automation that makes developers productive quickly

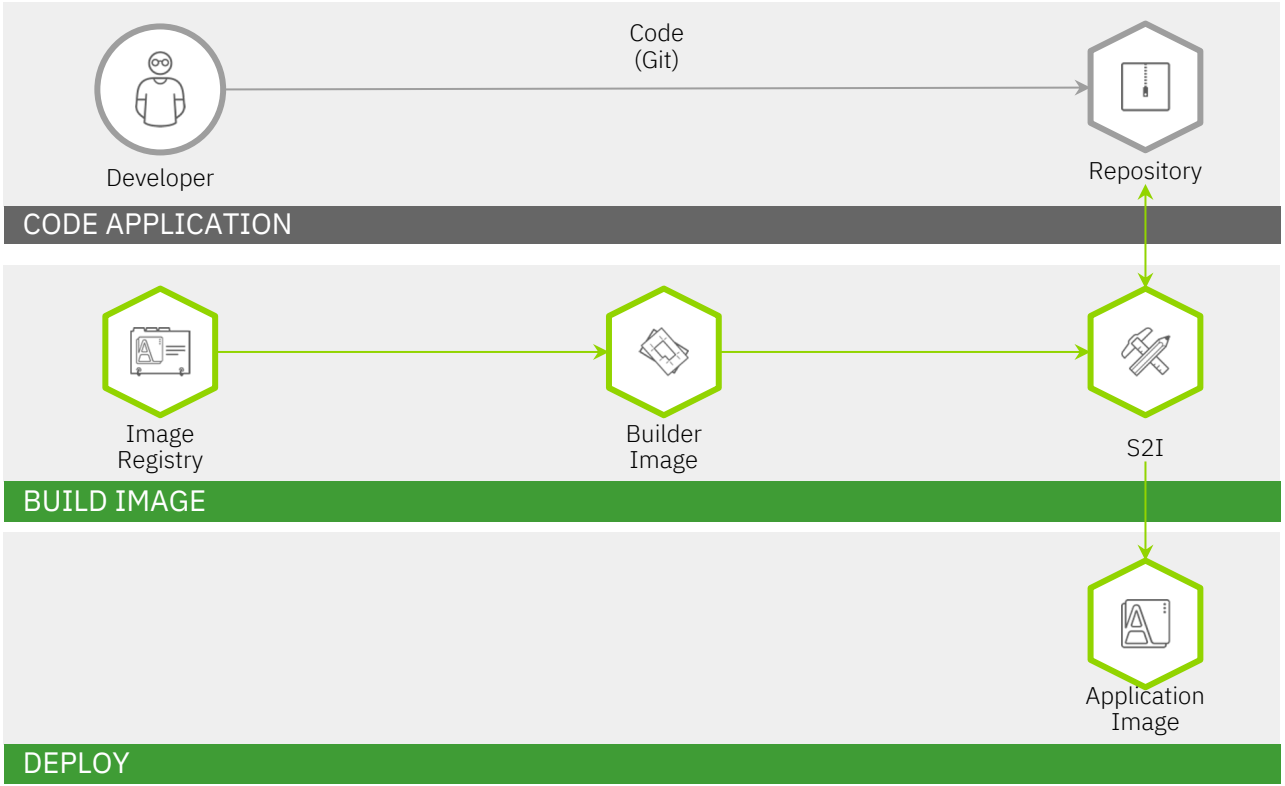


DEPLOY YOUR SOURCE CODE

DEPLOY YOUR APP BINARY

DEPLOY YOUR CONTAINER IMAGE

Methodologies to build and deploy



The Source-to-Image concept

Multitenancy

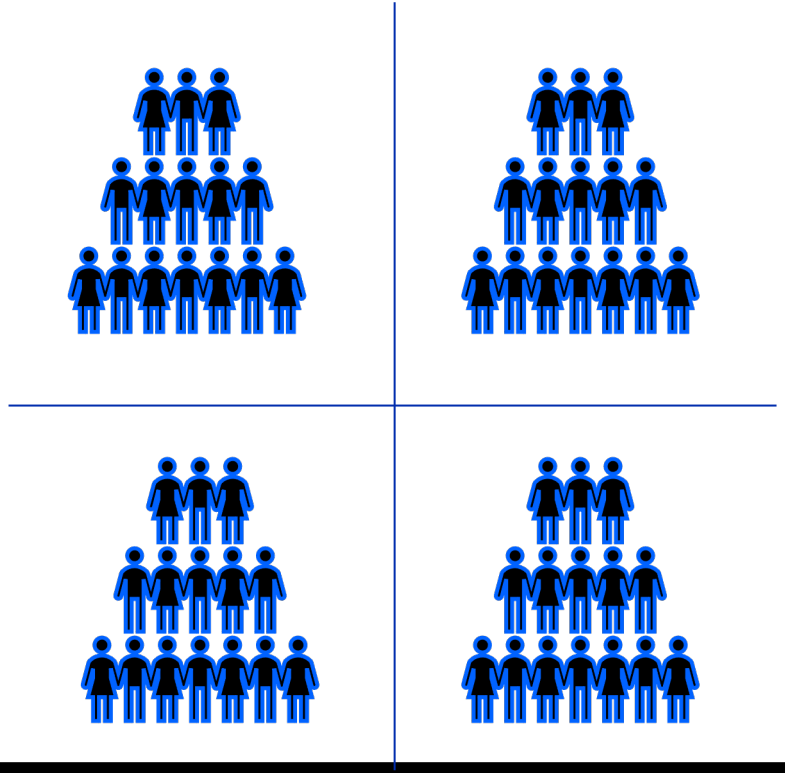
The main prerequisite: Thoughtful planning

- Software architecture where a single software instance can serve multiple, distinct user groups.
- Software-as-a-service (SaaS) offerings are an example of multitenant architecture.
- In cloud computing, multitenancy can also refer to shared hosting, in which server resources are divided among different customers.
- Multitenancy is the opposite of single tenancy, when a software instance or computer system has one end-user or group of users.

When referring to a container orchestration platform such as Kubernetes, the term multitenancy usually means *a single cluster that serves multiple projects*. The cluster is configured so each project runs with some degree of isolation from the others.

- When using Kubernetes for container orchestration, it’s possible to set up multitenant environments using a single Kubernetes cluster.
- Separate each tenant into their own namespace
- Create policies that enforce tenant isolation.
- There are benefits and risks associated with this which need to be considered as part of the decision-making process.

Multitenant security is essential for enterprise-scale use of Kubernetes. Multitenancy allows you to have different teams use the same cluster while preventing unauthorized access to each other’s environments.



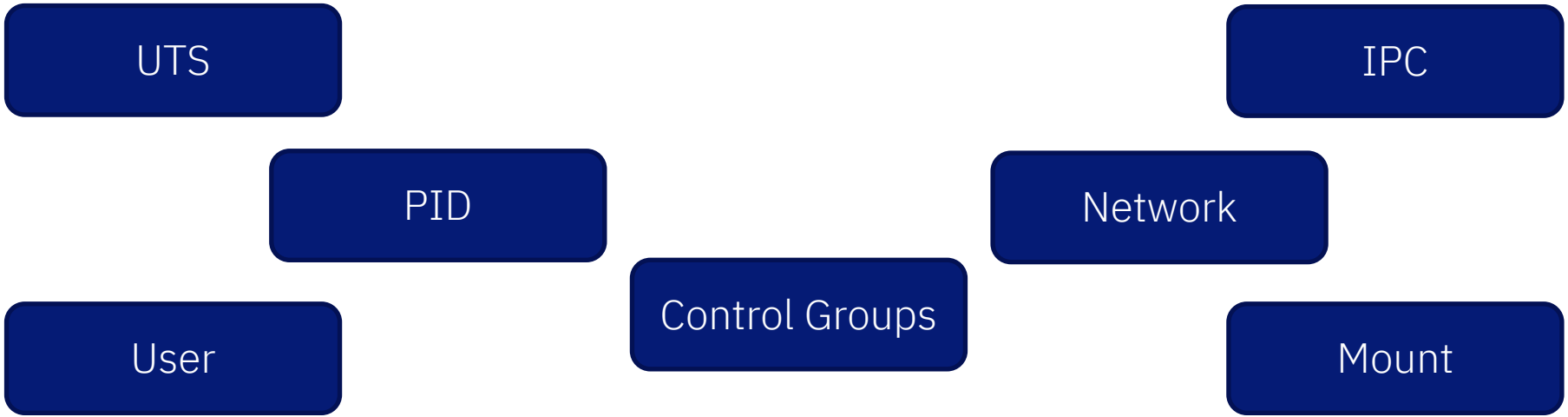
Red Hat OpenShift supports multitenancy through a combination of:

- Linux kernel namespaces
- SELinux
- Role-based Access Control (RBAC)
- Kubernetes namespaces
- Network policies.

Linux kernel Namespaces must not be conflated with Kubernetes Namespaces.

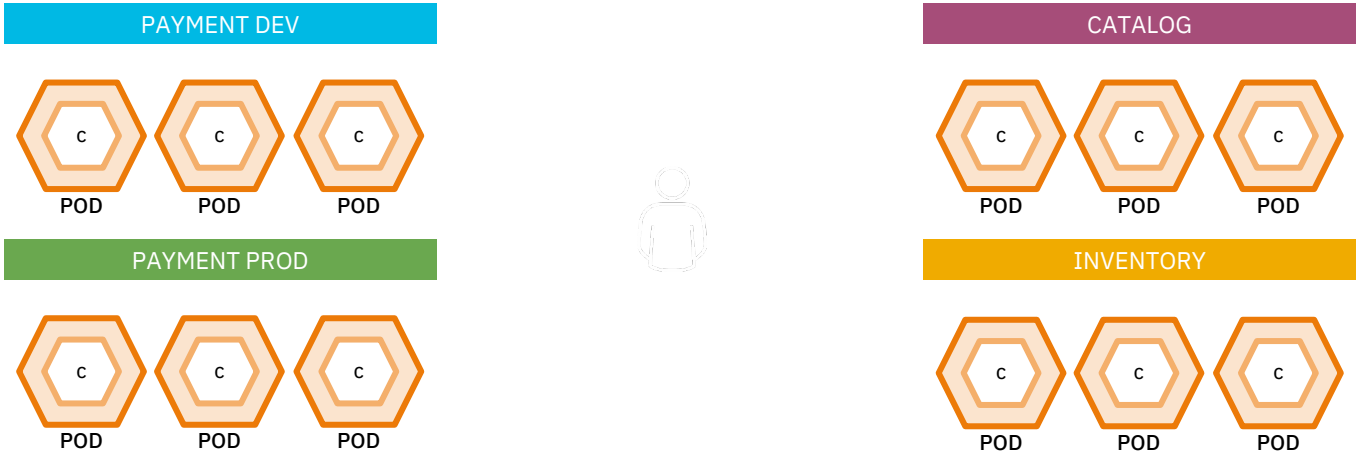
They are entirely different.

Linux kernel Namespaces are a set of seven abstraction wrappers for global system resources and in-scope processes.



https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/system_design_guide/what-namespaces-are_setting-limits-for-applications

Kubernetes Namespaces collate resources and isolate apps across environments, teams, groups and departments.



Namespaces were designed as a construct for cluster resource management, not security.

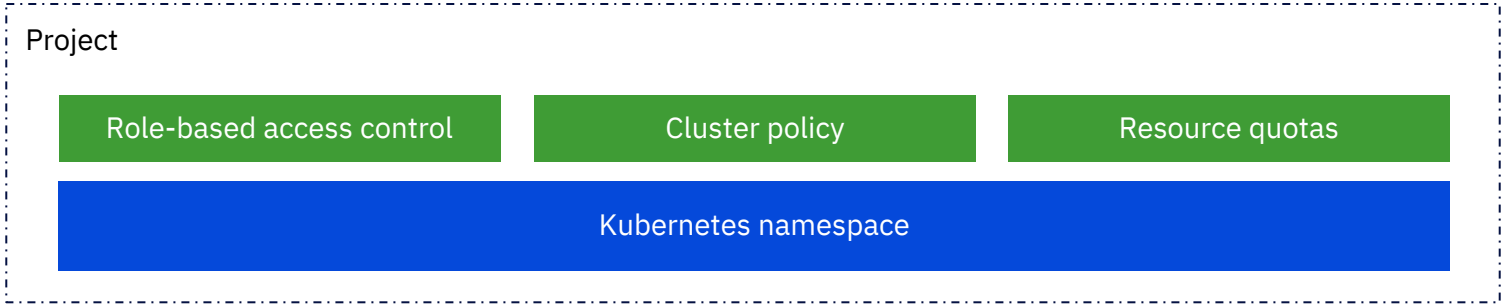
Do not rely on namespaces as a security feature outside of cluster internals within trusted domains.

Do not rely on namespaces to deny a cluster user access to resources in other namespaces.

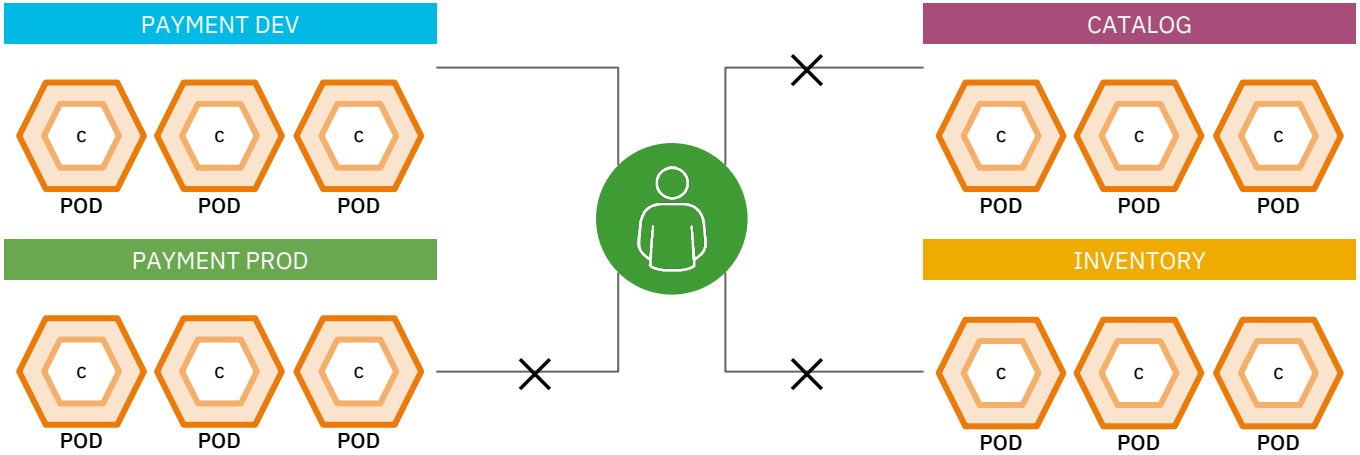
- NIST special publication 800-190: Use a container-optimized OS for additional security
- RHEL CoreOS
  - OS base for Red Hat OpenShift
  - Reduces the attack surface by minimizing the host environment and tuning it for containers
  - Only contains the packages necessary to run Red Hat OpenShift
  - Userspace is read-only
  - Tested, versioned, and shipped in conjunction with OCP 4 and managed by the cluster

Red Hat OpenShift supports multitenancy through a combination of kernel namespaces, SELinux, RBAC, Kubernetes namespaces (OCP Projects), and network policies.

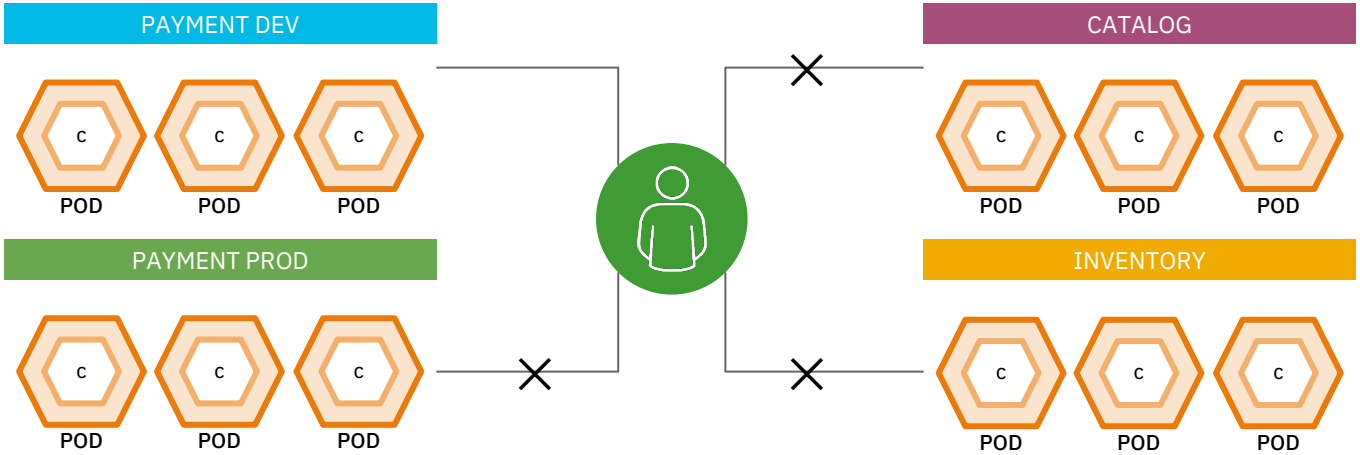
A namespace plus the RBAC layer and some other enhancements is a project



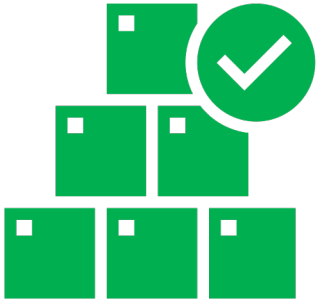
Projects isolate apps across environments, teams, groups and departments in a secure way.

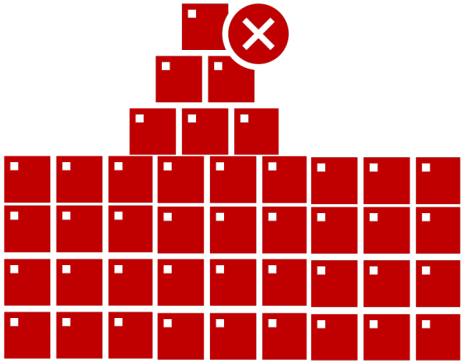


IBM Z and LinuxONE are the only platform where SECURE multi-tenant usage is possible



Embrace projects and use them on a sensible scale. Balance their performance enhancement against operational complexity.





Is this a good fit for purpose?

Web middleware / J2EE

Messaging and integration such as Kafka / EventStreams

HTTP content

🤔 Anything that needs to rapidly scale up to handle a burst in demand, and then scale back down afterward

🤔 Relational databases and other types of warehouses are exceptionally **unlikely** candidates

⛔ Put everything into containers because all the cool kids are doing it

sprawl

chaos

governance

Foundational governance is key to your success.

IBM has discovered that the surest path to container and microservices sprawl is to not have sound DevOps processes in place before adopting them.

As IT environments scale, thanks to the rise of containers and microservices, having mature processes in place to manage dynamic IT environments will be critical.

- Most IT organizations today still don't have many mature DevOps processes
  - What they do have in place was never really designed to address rapid changes to code enabled by microservices and containers.

Rise of microservices and containers is creating one of those seminal moments where organizations need to decide what role they want their internal IT operations teams to play.

The issue facing IT organizations now is how much do they want to take care of that problem today versus waiting for an outcome that, at this point, is all but inevitable.

user experience must never be an afterthought!

If you are planning to expose any of the *https://<<application name>>.apps.<<clustername>>.<<domain>>* URLs to your end-users, you are going about this incorrectly.

If you are planning to deploy applications without governance to ensure they use a unique URI path, you are going about this incorrectly.

Especially if you are thinking about using the server root! Please don't!

OCP for Web applications belongs behind a proxy

Where to find more information

Datasheet – Running OCP on IBM Z and LinuxONE:

ibm.com/downloads/cas/2JMPXMZK

Cloud Native Computing Foundation (CNCF)

- landscape.cncf.io
- glossary.cncf.io

Kubernetes

kubernetes.io/docs/tutorials/kubernetes-basics/

IBM Cloud Paks

ibm.com/demos

Red Hat OpenShift

learn.openshift.com

The mission of the Washington Systems Center is to provide world-class subject matter expertise and technical sales support to IBM marketing and sales teams for the sale, design, sizing, implementation, optimization, and support of client solutions built on hardware, software, and services offerings from IBM including IBM zSystems and LinuxONE.

Storage solutions

- IBM Storage: disk, flash, replication management
- IBM Spectrum software-defined storage
- Linux container storage expertise

Capacity, performance, and sizing

- CPS applications suite for z/Architecture systems
  - CP3000, zBNA, zPCR

Servers and software

- IBM Z and LinuxONE hardware
- Linux, z/TPF, z/OS, z/VM, z/VSE operating systems
- Select software & Z software subsystems
  - CICS, IMS, Db2, WAS, MQ, JakartaEE, OracleDB

LinuxONE z/VM z/VSE Linux z/TPF z/OS

Workshops and knowledge transfer

No-charge knowledge transfer and skills enablement workshops for IBM customers seeking to enhance their knowledge of hardware, software, and services offerings from IBM Z, LinuxONE, Storage, Spectrum, and middleware.

- www.ibm.com/support/pages/node/1125513
- www.ibm.com/support/pages/node/6354049

The IBM Washington Systems Center

Thank you

Paul Novak
Senior IT Specialist
SME, Virtualization & Cloud
on IBM Z and IBM LinuxONE

Endicott – The Birthplace of IBM
1701 North St
Endicott, NY 13760 USA

Tel +1 607 429 6186
pwnovak@us.ibm.com

Matt Mondics
Advisory IT Specialist
SME, Cloud on IBM Z and
LinuxONE

10500 Cedar Ave
Cleveland, OH 44106 USA

Tel +1 614 551 7720
matt.mondics@ibm.com

Visit our website at http://www.ibm.com/support/techdocs