Linux Container Solutions on IBM zSystems® & LinuxONE™

Jacob Emery

Matt Mondics

Paul Novak

Workshop ID LCSZ1
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.

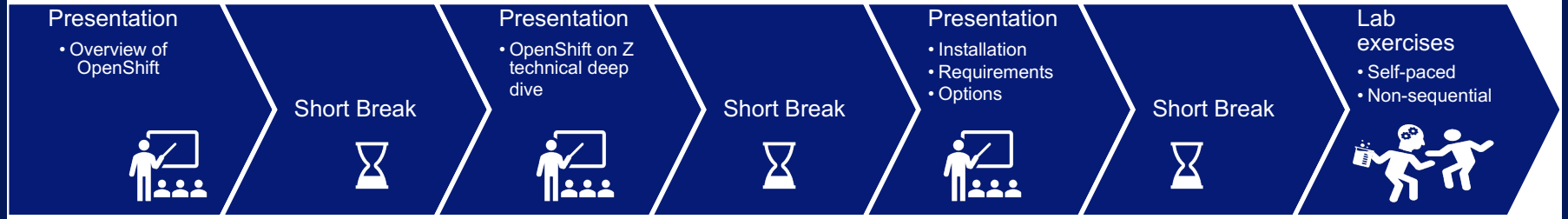
Special Notices and Trademarks

Red Hat OpenShift Container Platform on IBM Z & LinuxONE

All workshop materials can be found here:
https://mmondics.github.io/ocp-cloudpak-workshop/

Schedule for the day

Start 10:00 AM Eastern



End 4:00 PM Eastern

Workshop Content

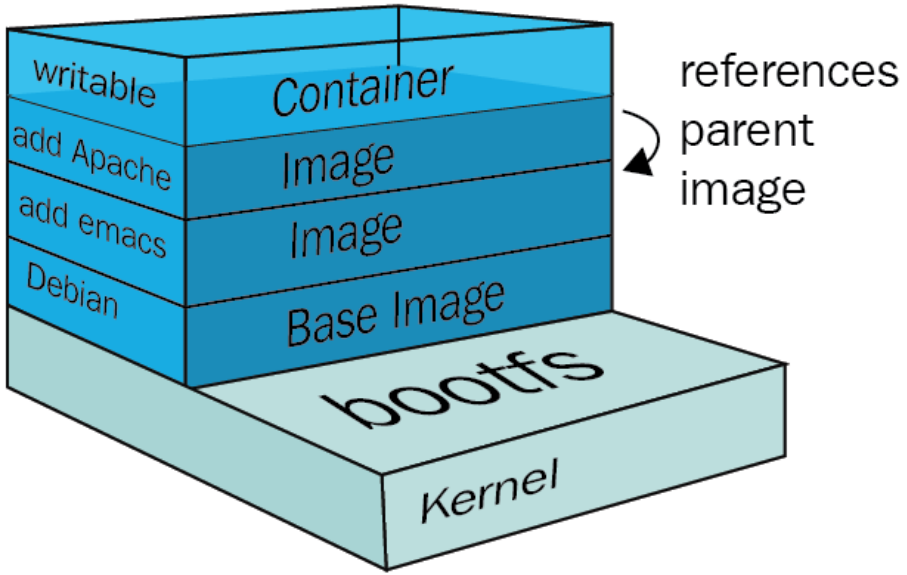
Part One: Containers, Kubernetes, and OpenShift on IBM Z

Matt Mondics
Technical Sales Enablement Specialist
IBM Washington Systems Center

Containers

Container Terminology

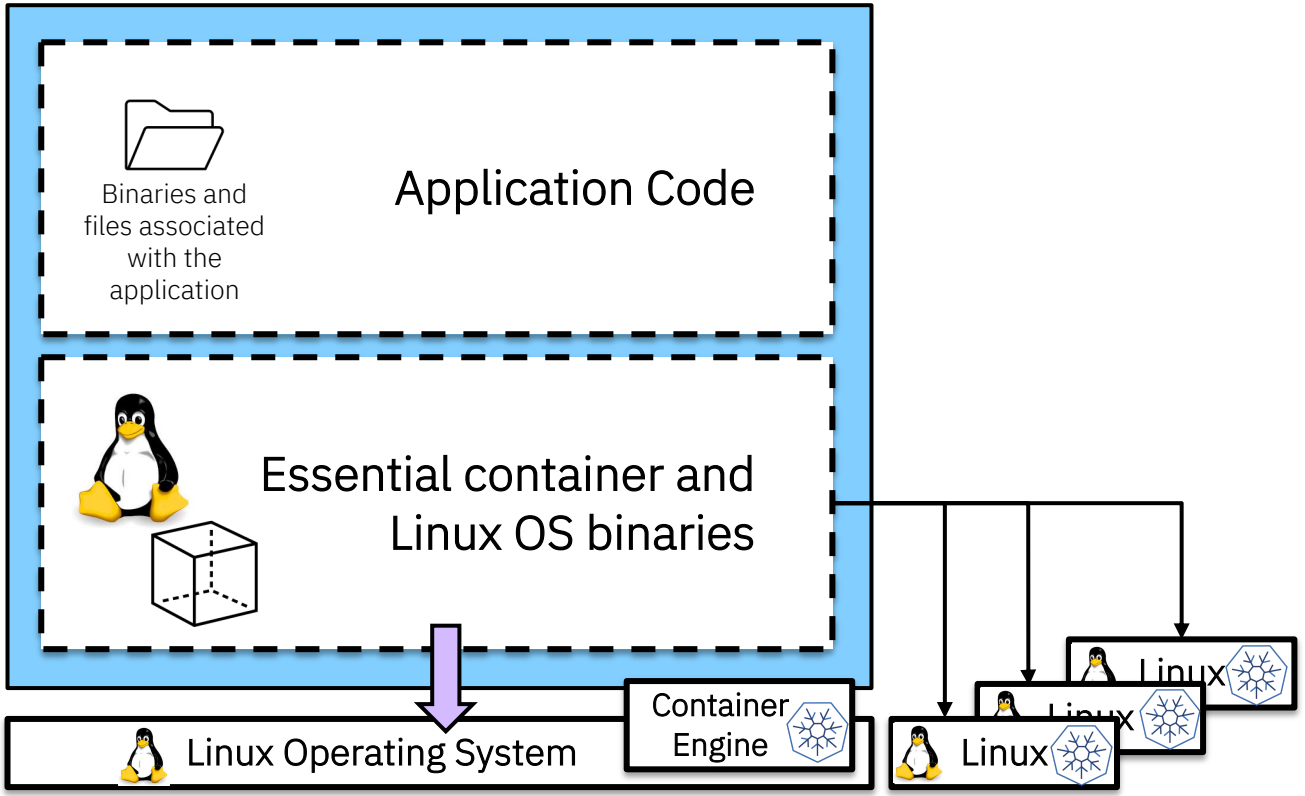
- Container: a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another
- Container image: a lightweight, standalone, executable package of software that includes everything needed to run an application
- Container engine takes a container image and turns it into a container (i.e. a running processes)



Containerization Overview

Containerization is not a mystical concept.

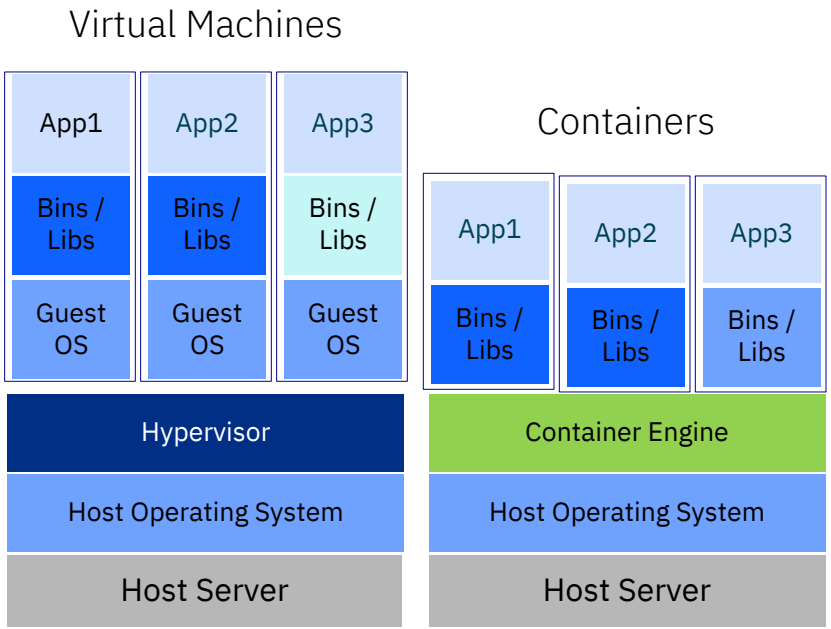
- Start with the application and associated files
- Add essential container and Linux files and binaries
- Build a container image with provided tooling
- Run image on systems where container runtime exists



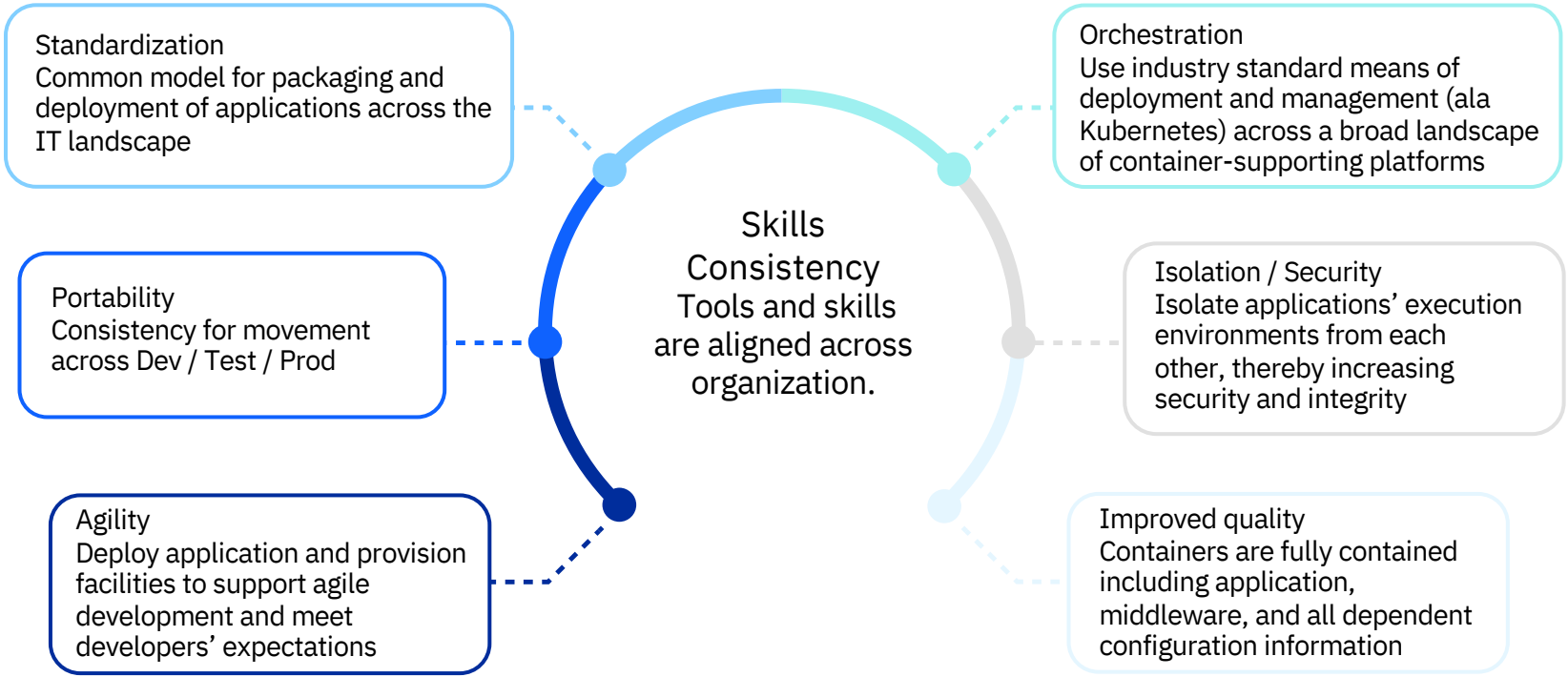
Containers vs. VMs

Unlike virtual machines, containers don’t virtualize the entire operating system or any underlying hardware.

- Portability
- Build Once, Deploy Anywhere
- Performance
- Resource Isolation
- Management Cost
- Layered Image Architecture
- Great for DevOps and Microservices



Benefits of Containers



Kubernetes

Although container images and the containers that run from them are the primary building blocks for modern application development, to run them at scale requires a reliable and flexible distribution system. Kubernetes is the defacto standard for orchestrating containers.

Kubernetes is an open source container orchestration engine for automating deployment, scaling, and management of containerized applications.

What is Kubernetes?

A Container Orchestrator designed to automate container deployment, scaling, and management

- Developed by Google in 2014
- Used by Google to manage billions of containers per week running their services

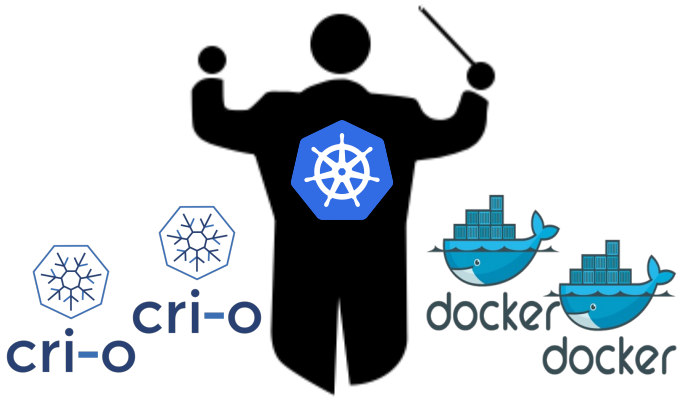
Seed technology of the Cloud Native Computing Foundation (CNCF)

CLOUD NATIVE
COMPUTING FOUNDATION

First production grade version (1.0) released July 2015

~Quarterly release since 1.2.0 in March 2016

Latest 1.2.3 released in January 2022



THE Kubernetes platform for IBM Z and LinuxONE

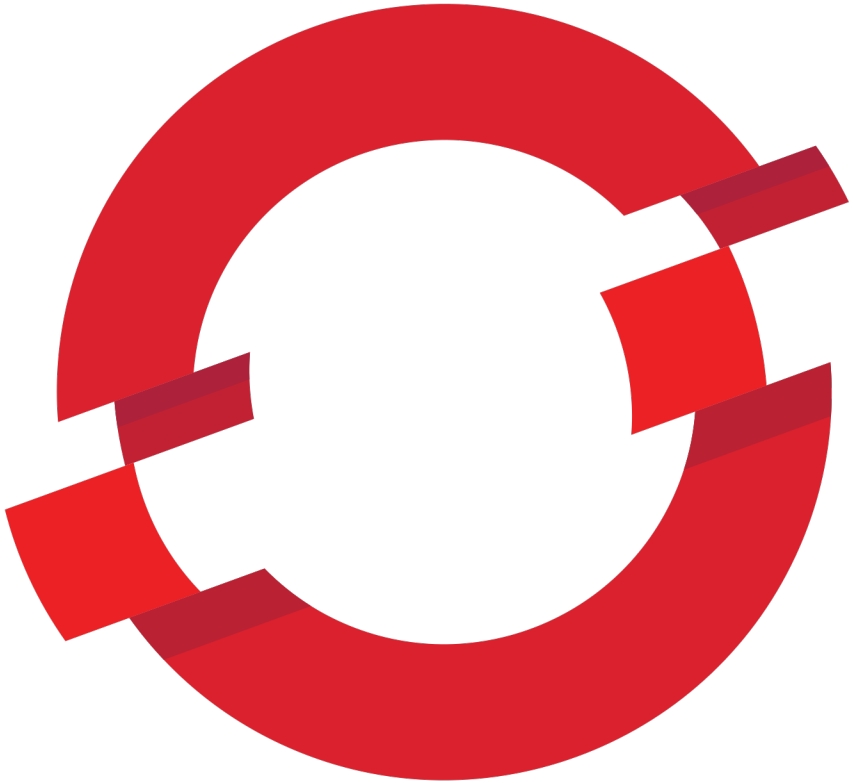
IBM is adopting the Red Hat OpenShift Container Platform 4 on Linux on IBM Z and IBM LinuxONE.

This offering will accelerate the transformation to greater portability and agility through integrated tooling and a feature-rich ecosystem for cloud native development on Linux on IBM Z and LinuxONE offerings.

IBM also intends to deliver IBM Cloud Pak offerings to Linux on IBM Z and LinuxONE offerings. These offerings will accelerate the rich IBM software ecosystem that is necessary for enterprise clients to adopt hybrid multi deployment.

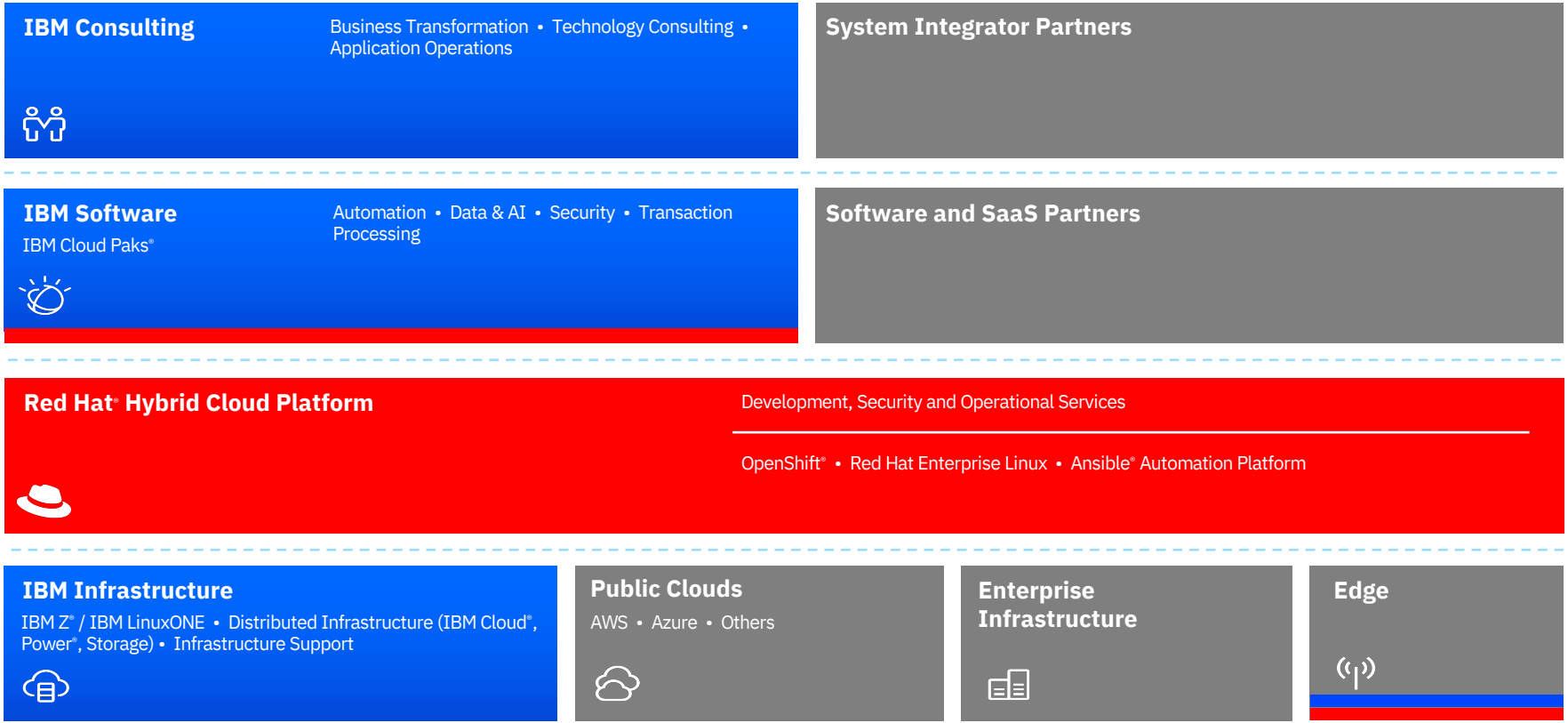
These offerings, combined with the IBM premier enterprise platforms, IBM Z and LinuxONE, will reinforce and further strengthen the IBM focus on hybrid cloud to unlock business value and drive growth for clients by providing a secure and open hybrid, multicloud platform.

OpenShift Container Platform

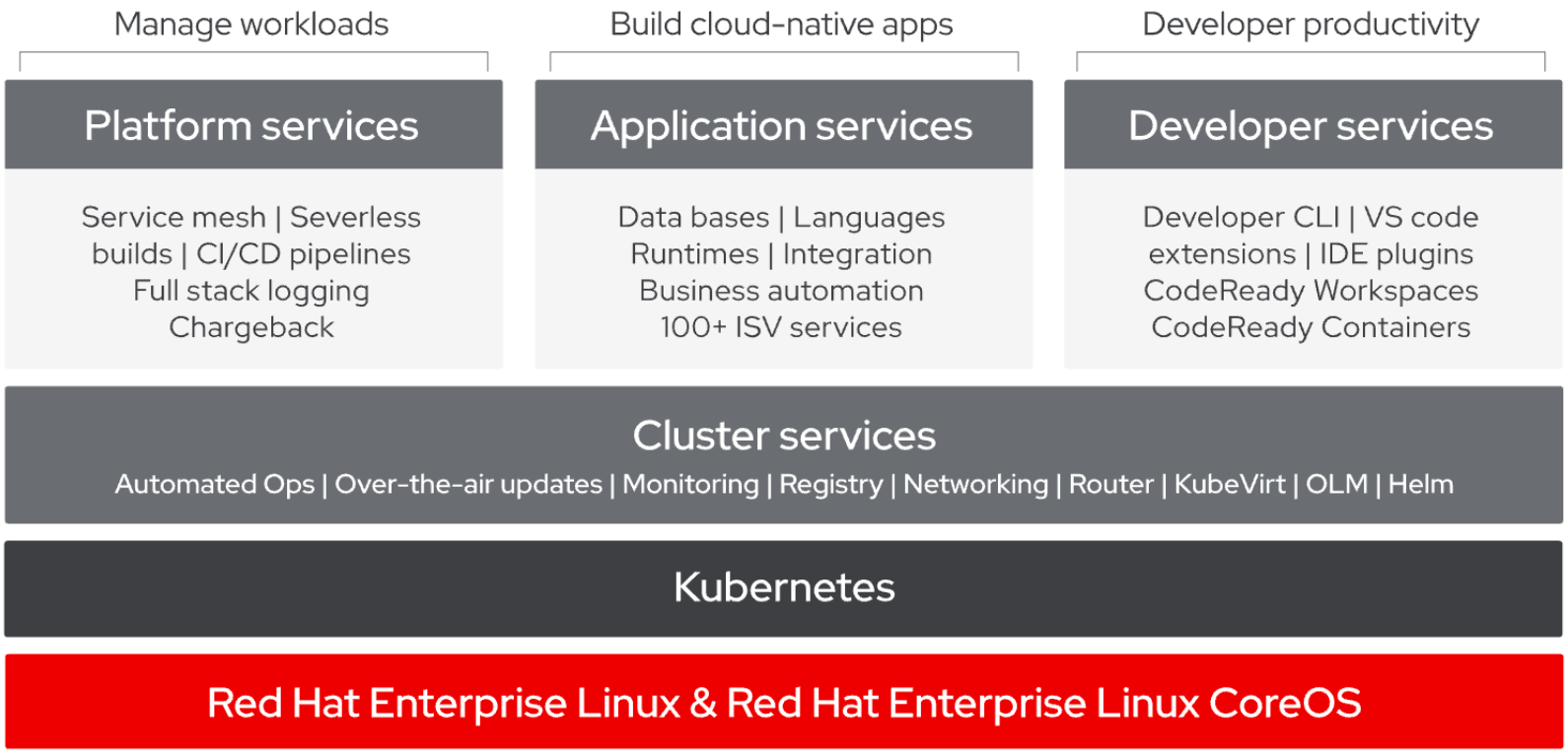


OPENSHIFT

IBM’s hybrid cloud and AI platform approach



OpenShift Container Platform (OCP) Overview



Under the Hood

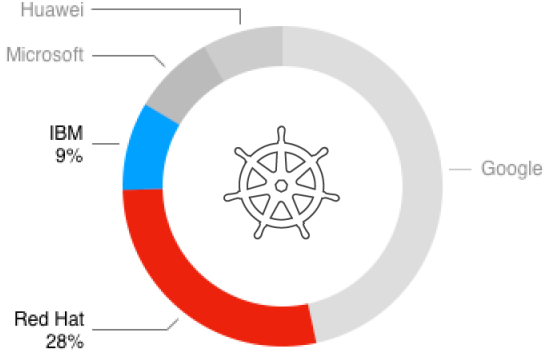
okd

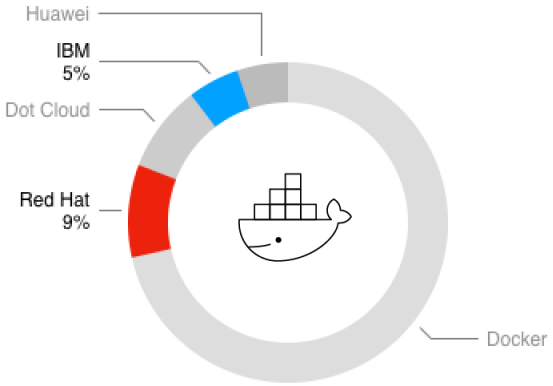
- OKD – Upstream Open Source Software
- Integrate additional OSS projects
- 100+ Integrations
- Validated OSS Innovation
- Partner Integration Platform

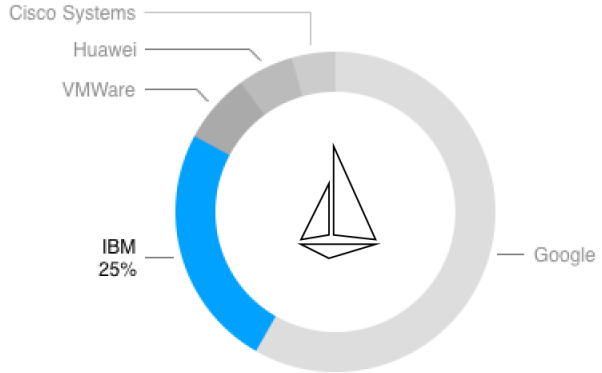
OPENSHIFT
CONTAINER PLATFORM

IBM and Red Hat Open Source Contributors

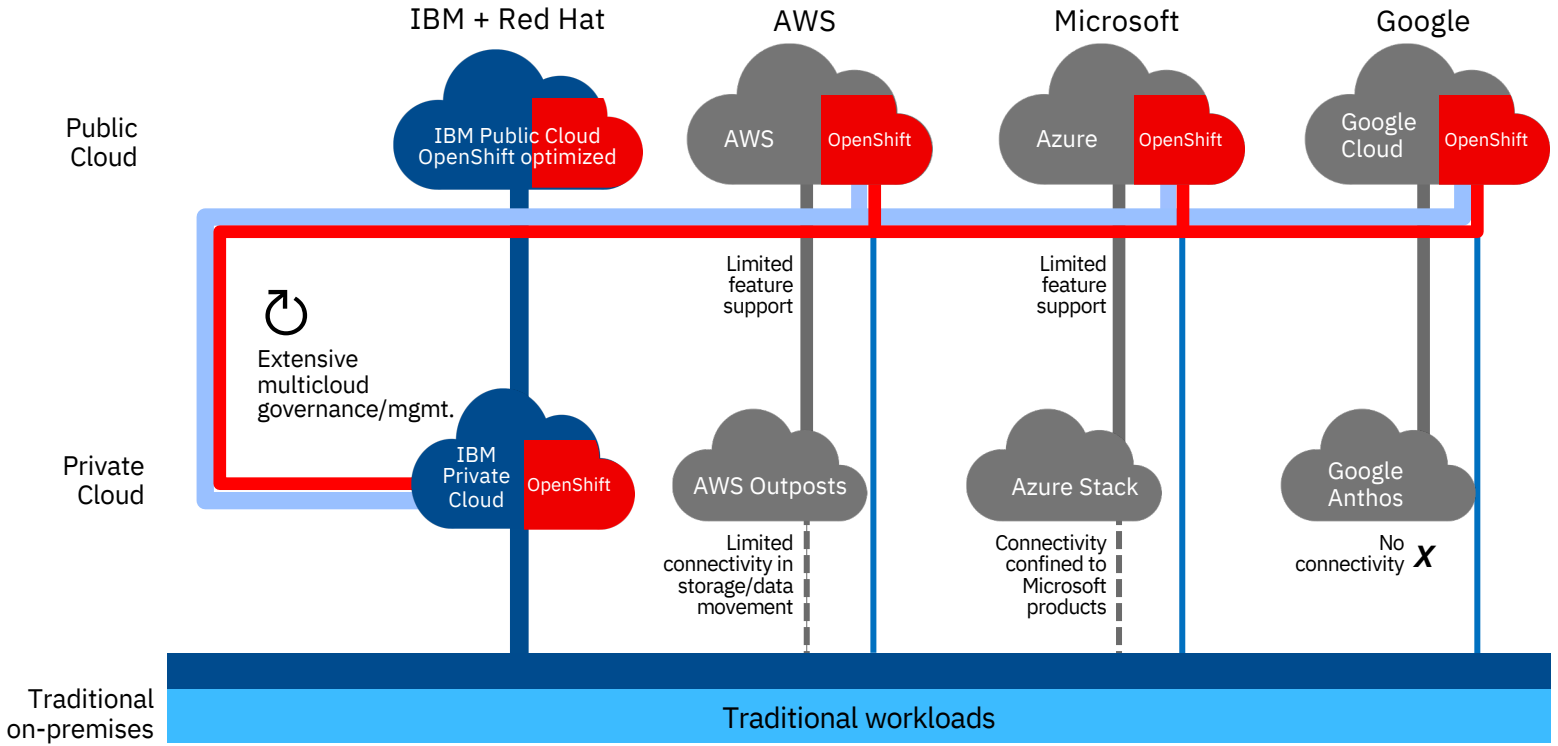
IBM and Red Hat contributions to Kubernetes, Docker, and Istio projects (top 5 orgs apart from independent developers).







True Hybrid Multicloud



Emerging adoption patterns for Red Hat OpenShift

Co-location

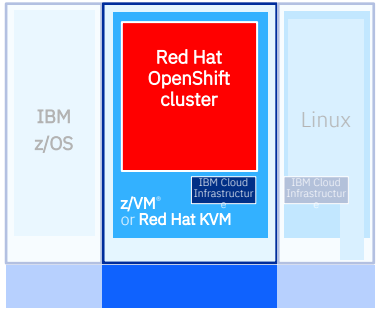
Co-locate containerized workloads with z/OS data to achieve lower response time and meet enterprise SLA

Modernization

Adopt cloud native to achieve consistency and grow containerized workloads

z/OS Integration

Modernization and automation of z/OS with hybrid cloud on IBM Z / LinuxONE



Platform capabilities

High throughput per core, low latency, high scalability, out of the box availability and resiliency

AI and Data

Leverage AI to extract critical insights for business transformation and achieve agility

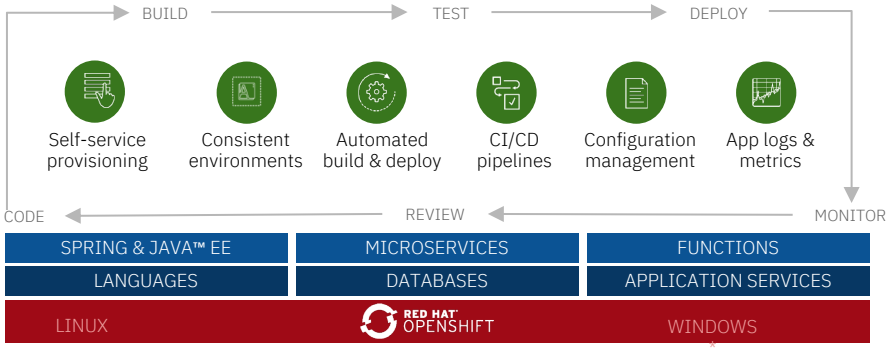
Hyperleger fabric

Hyperledger fabric deployed on-premises on IBM Z

Adoption pattern: Application modernization

Consistent cloud experience built on Red Hat OpenShift

- Solves development challenges
  - Develop once deploy on multi-arch (CI/CD & DevOps)
- Enables new ways for hybrid IT projects including cloud
  - Best fit placement for apps based on SLAs
- Solves main business problems
  - Faster time to market, perfect for dynamic workloads



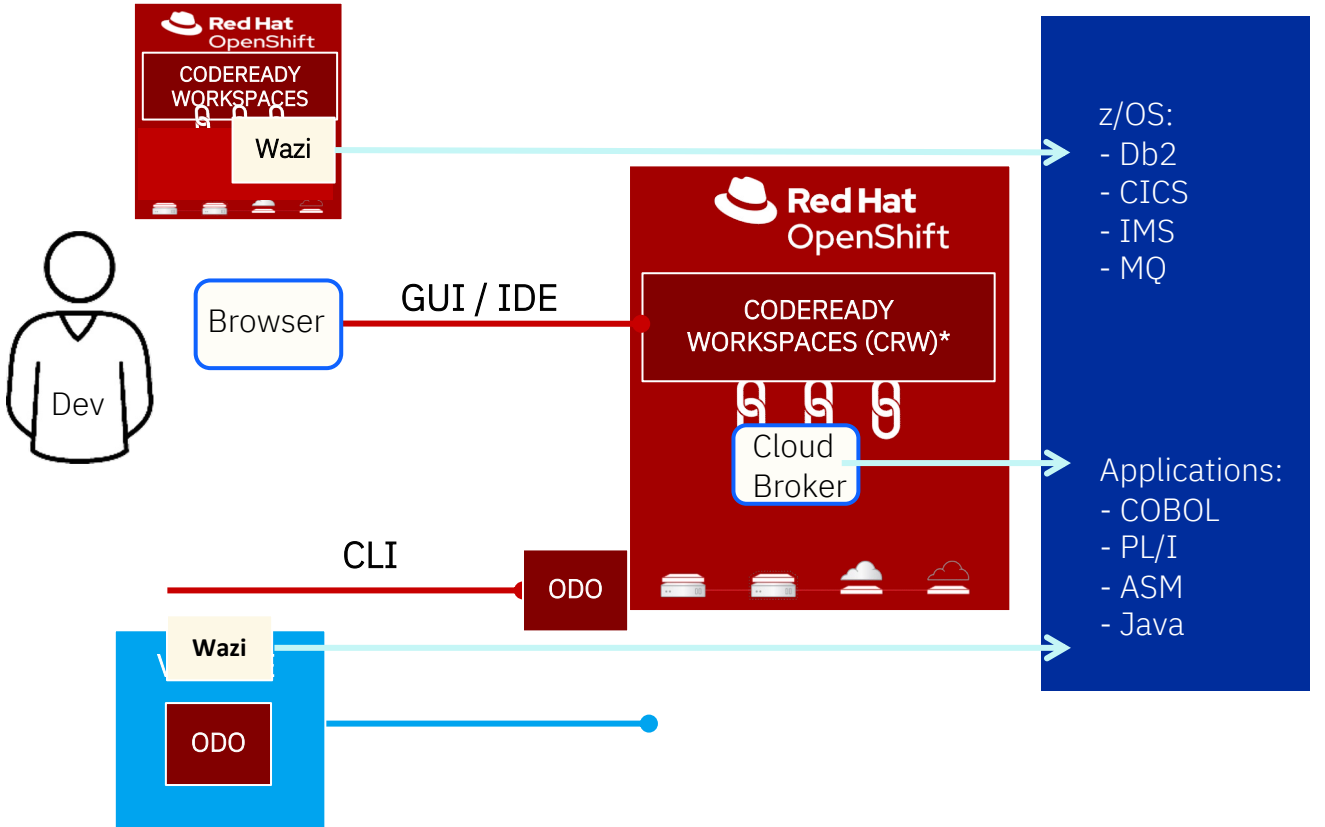
Red Hat OpenShift add-ons helps to create, deploy and manage apps with speed and consistency

- odo 2.5 on RHOCP 4.10 (with plug-ins in CodeReady Workspaces, MS Visual Studio)
- CodeReady Workspaces 2.15
- Pipelines (Tekton) 1.6 on RHOCP 4.9
- Serverless 1.20 on RHOCP 4.6/4.7/4.8/4.9/4.10
- Service Mesh 2.1.1 on RHOCP 4.6/4.7/4.8/4.9

Adoption pattern: Application modernization

Cloud-native development tools in Red Hat OpenShift

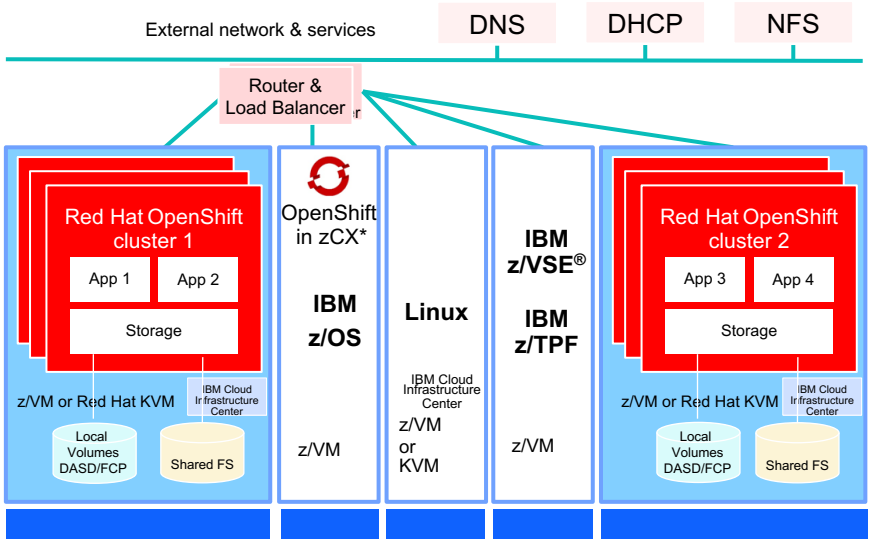
Red Hat OpenShift development tools help to achieve the CI/CD objectives, including automation of processes to promote code through build, test, and deployment.



Adoption pattern: Co-location

Co-locate containerized apps beside existing apps and data

- Centralized management of workloads
- Certified workload isolation and protection
- High levels of scalability with granular sharing
- Streamlined IT infrastructure with fewer points of attack



** zCX = IBM z/OS Container Extensions*

Adoption pattern: Co-location

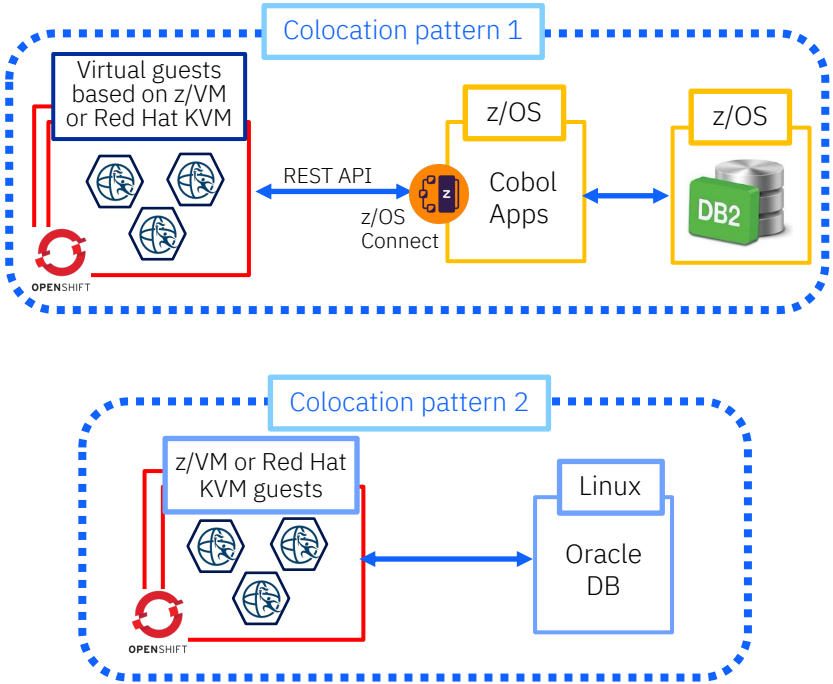
Benefits of co-locating workloads

Co-location is when the presentation, business logic, and data serving layers of a **multi-tier workload onto a single physical server.**

Co-location can provide savings in

- **throughput improvements** for both interactive and streaming workloads,
- **reductions in latency** of network communication.

When accessing your database while running an OLTP workload on OpenShift Container Platform, achieve **4.2x more throughput by co-locating** the workload on IBM z16 versus running the workload on compared x86 platform connecting remotely to the IBM z16.*



* This is an IBM internal study designed to replicate banking OLTP workload usage in the marketplace deployed on OpenShift Container Platform (OCP) 4.9 on IBM z16 using z/VM versus on compared x86 platform using KVM accessing the same PostgreSQL 12 database running in an IBM z16 LPAR. IBM z16 configuration: The PostgreSQL database ran in a LPAR with 12 dedicated IFLs, 128 GB memory, 1 TB FlashSystem 900 storage, RHEL 7.7 (SMT mode). The Compute nodes ran on z/VM 7.2 in a LPAR with 30 dedicated IFLs, 188 GB memory, DASD storage, and OSA connection to the PostgreSQL LPAR. LPAR with 2 IFL, 4 GB memory and RHEL 8.5 with OCP Proxy server. x86 configuration: The Compute nodes ran on KVM on RHEL 8.5 on 32 Cascade Lake Intel® Xeon® Gold CPU @ 2.30 GHz with Hyperthreading turned on, 192 GB memory, RAID5 local SSD storage, and 10Gbit Ethernet connection to the PostgreSQL LPAR. Results may vary.

Adoption pattern: Co-location

Large FSS company

Accelerate enterprise digital transformation

- Containerized services running in OpenShift on IBM Z are co-located on the same server with z/OS Db2 data and CICS for low latency, high volume transaction processing
- Achieve up to 7.3x lower latency co-locating applications on IBM Z compared to connecting to an x86 server

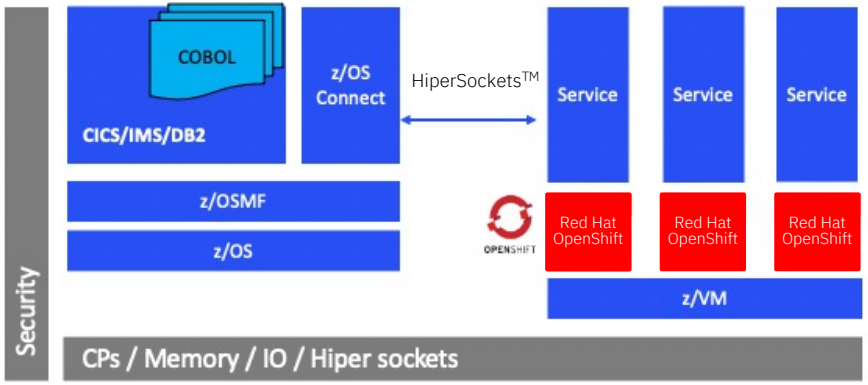
Modernize and digitally transform

- Modernize and extend mission-critical legacy assets incrementally while maintaining enterprise SLAs and keeping risk and cost low

Before co-location

- Microservices had been running with VMware on x86
- Multiple layers of network had been required

Co-location on IBM Z:

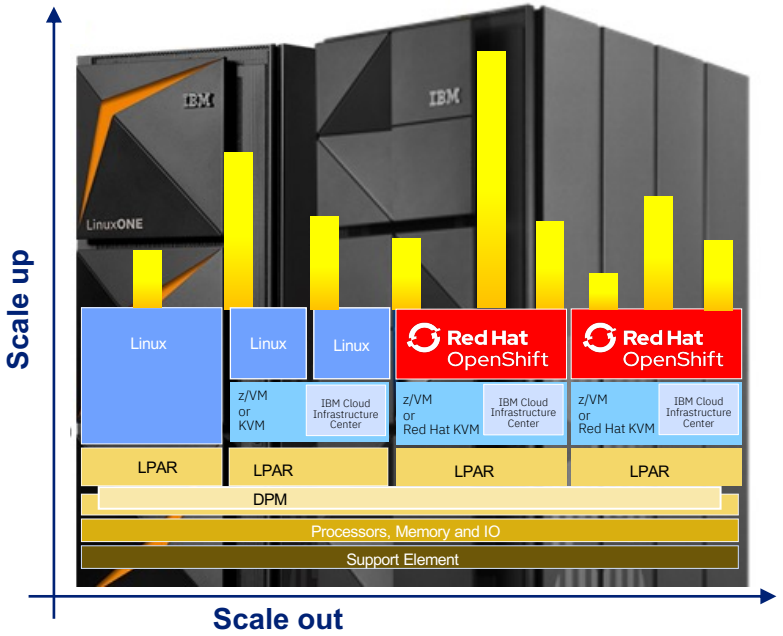


Adoption pattern: Cost optimization

Cost optimization of containerized workload

- High workload density / server capacity
- Fewer IT components require less maintenance
- High resource utilization helps on less software license costs
- Unique arrangements for administration, security, backup and DR
- Transparent exploitation of on-chip accelerated AI, compression, and encryption

Accessing your database while running an OLTP workload on OpenShift Container Platform, requires up to **3.6x fewer cores** running your workload when co-located on IBM z16 versus running the workload on compared x86 platform connecting remotely to the IBM z16.*

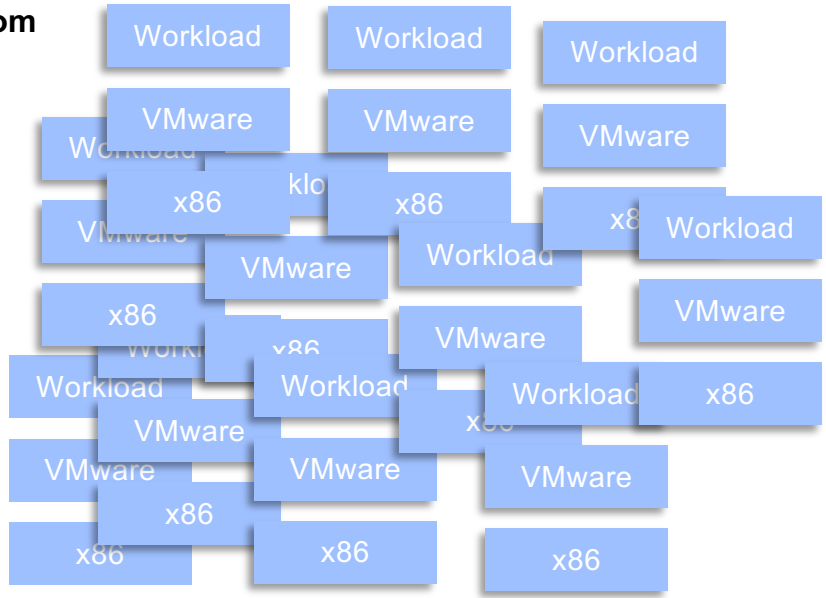


* This is an IBM internal study designed to replicate banking OLTP workload usage in the marketplace deployed on OpenShift Container Platform (OCP) 4.9 on IBM z16 using z/VM versus on compared x86 platform using KVM accessing the same PostgreSQL 12 database running in a z16 LPAR. IBM z16 configuration: The PostgreSQL database ran in a LPAR with 12 dedicated IFLs, 128 GB memory, 1TB FlashSystem 900 storage, RHEL 7.7 (SMT mode). The Compute nodes ran on z/VM 7.2 in a LPAR with 8 dedicated IFLs, 188 GB memory, DASD storage, and OSA connection to the PostgreSQL LPAR. The OCP Proxy server ran in an LPAR with 1 IFL, 4 GB memory and RHEL 8.5. x86 configuration: The Compute nodes ran on KVM on RHEL 8.5 on 32 Cascade Lake Intel® Xeon® Gold CPU @ 2.30GHz with Hyperthreading turned on, 192 GB memory, RAID5 local SSD storage, and 10Gbit Ethernet connection to the PostgreSQL LPAR. Both systems are delivering equal throughput. Results may vary.

Adoption pattern: Cost optimization

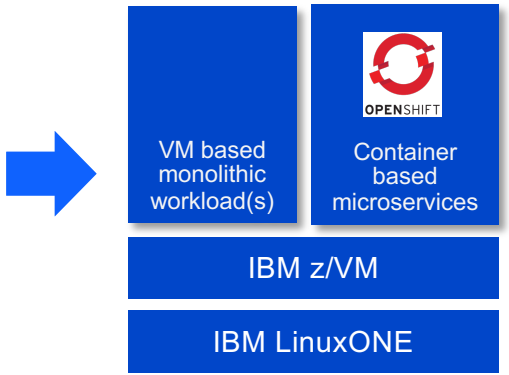
Large FSS client

From



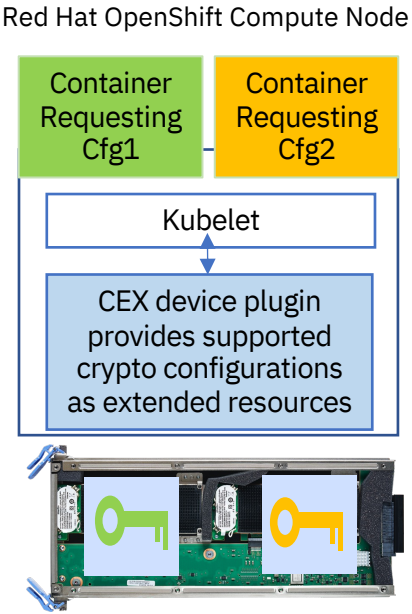
Client wanted to **consolidate** 1000's of x86 cores worth of workload to **reduce OpEx and datacenter sprawl.**

With IBM LinuxONE, the client consolidated the workloads and saw a large TCO reduction.



Other clients have seen a consolidation ratio up to 22:1 with Linux workloads.

IBM Crypto Express support for containers on Red Hat OpenShift



Kubernetes device plug-in for IBM Crypto Express (CEX) cards

Enables OpenShift containers to take advantage of crypto resources on IBM Crypto Express adapters

- Implemented as container running on each compute node
- Detects CEX resources available on each compute node
- Assigns the CEX resource to the container
- Frees resources after container termination

Supported as Red Hat certified container.

The IBM hybrid cloud software approach : Cloud Paks

AI-Powered solutions for specific use cases and problem domains

Software and AI applications

Containerized Software

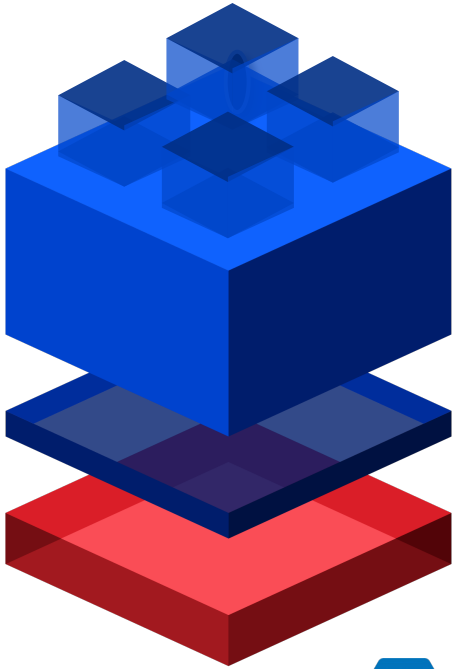
Cloud Paks

Foundational services Consistency and shared capabilities

Foundational Services

Enterprise grade container platform

Red Hat OpenShift Hybrid cloud platform



aws Azure Google Cloud Platform

IBM Cloud Paks: AI-powered hybrid cloud software

Containerized software, foundational services and Red Hat OpenShift

IBM Cloud Pak for **Data**

Unify and simplify the collection, organization and analysis of data

- Transactional and analytical databases
- Data virtualizion, governance, privacy
- Data integration/ETL
- AutoAI and Visual Data Science
- Model Risk Management, fairness and explainability

IBM Cloud Pak for **Business Automation**

Automate business operations to achieve better performance

- Workflow and decisions
- Content services
- Operational intelligence

IBM Cloud Pak for **Watson AIOps**

Automate IT operations to deliver actionable insights

- Application impact avoidance
- Hybrid application management
- Observability

IBM Cloud Pak for **Integration**

Automate application and data flows to improve client experiences

- Application integration
- API management
- Messaging and events

IBM Cloud Pak for **Network Automation**

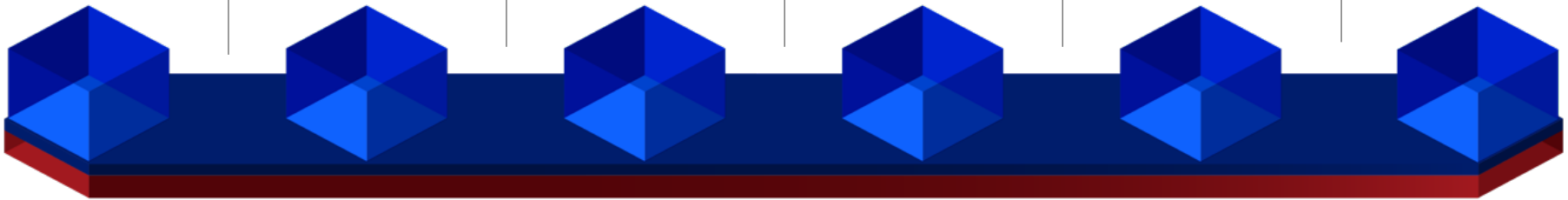
Automate networks to deliver zero-touch operations

- Intent-driven orchestration
- Closed-loop operations
- Network optimization

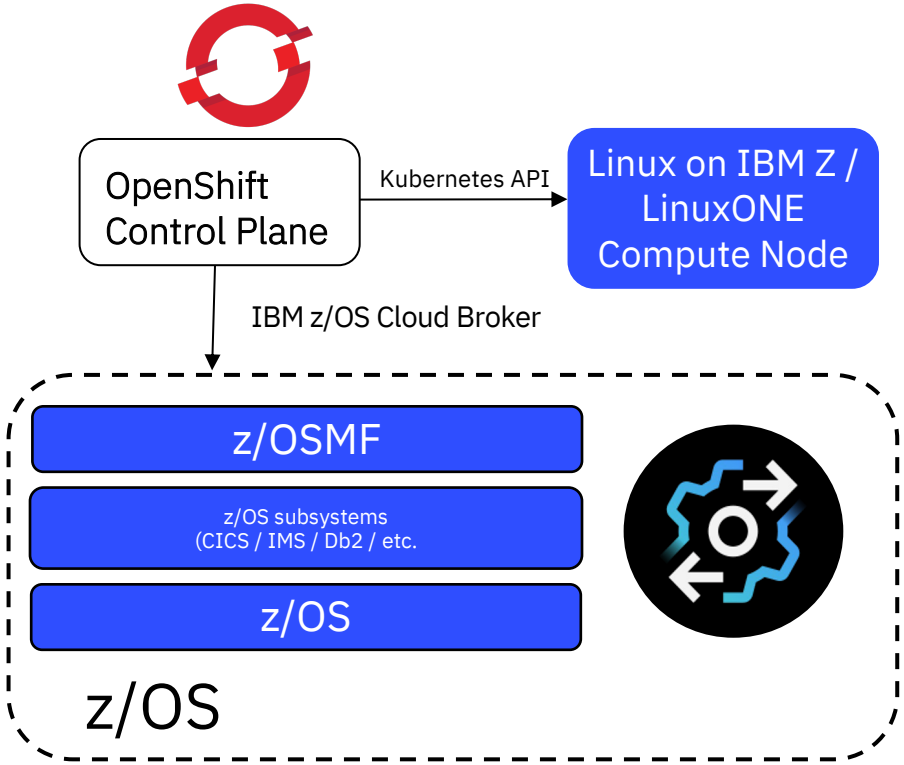
IBM Cloud Pak for **Security**

Generate deeper insights into threats, orchestrate actions and automate responses

- Threat intelligence
- Federated search
- Security information and event management
- Data activity monitoring



z/OS Cloud Broker



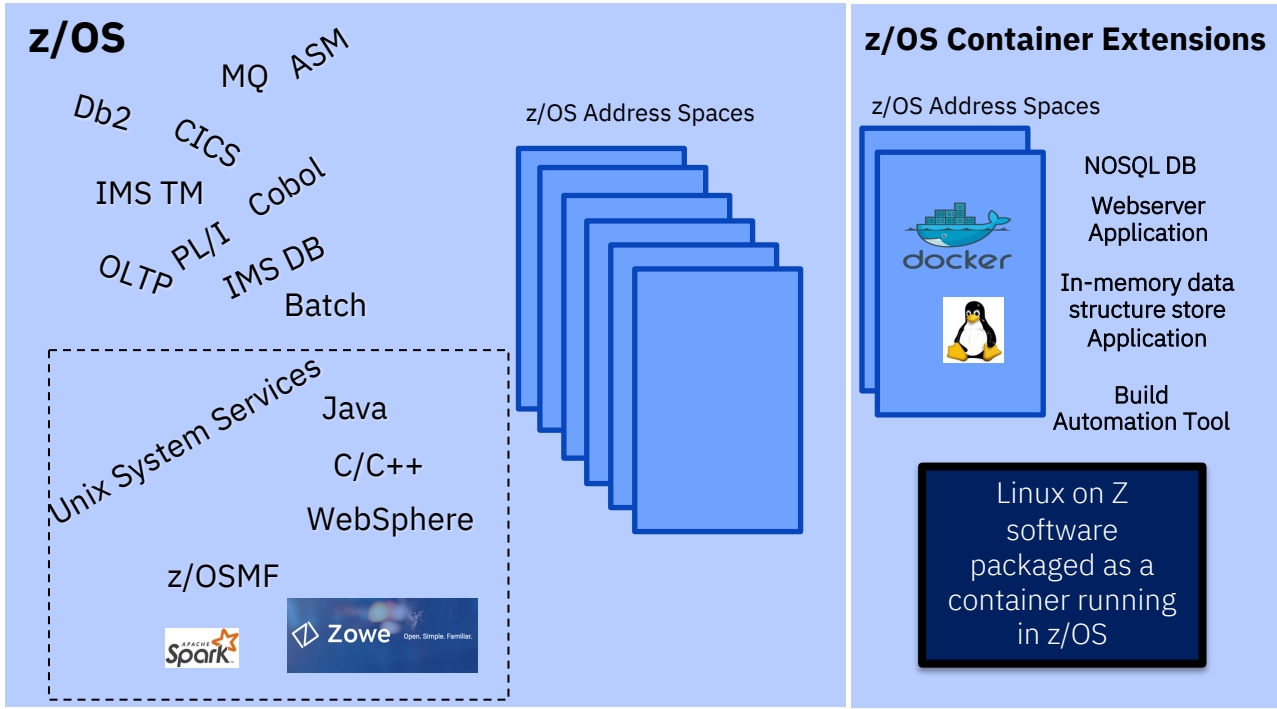
z/OS Cloud Broker is technology that gives users the ability to access and deploy z/OS resources and services on OpenShift for a seamless and universal cloud development experience.

DevOps Providing access to IBM Z resources to all flavors of application developers leveraging open standards and tools. i.e. common cloud marketplace.

Service Provider Centralization and automation of IBM Z operations to provide Z resources to agencies or clients in a self-service model.

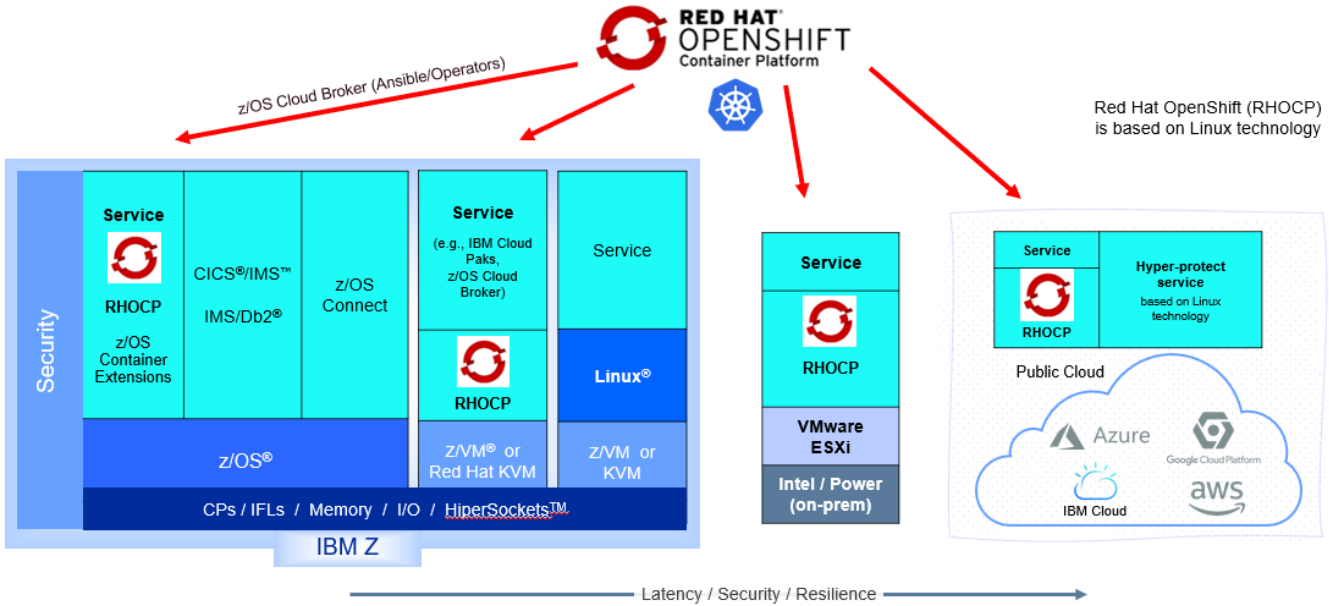
Multi-Cloud Service Integration Providing a unified experience for IBM Z cloud services across private cloud and public cloud.

z/OS Container Extensions



The vision of hybrid cloud and multicloud with Red Hat OpenShift

Hybrid workloads and multiple Red Hat OpenShift clusters can run in parallel on a physical IBM Z server.



Thank you

Matt Mondics
Technical Sales Enablement Specialist - OpenShift on IBM Z
—
matt.mondics@ibm.com
IBM Washington Systems Center (WSC)

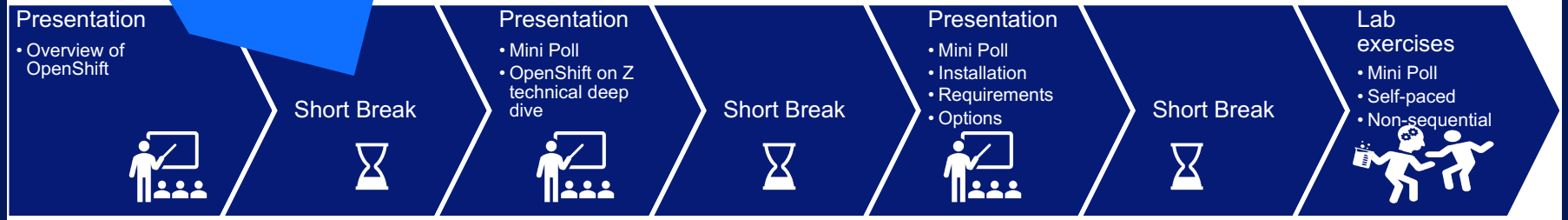
Red Hat OpenShift Container Platform on IBM Z & LinuxONE

On break until 11:15

All workshop materials can be found here:
https://mmondics.github.io/ocp-cloudpak-workshop/

Sche… day

Start 10:00 AM Eastern

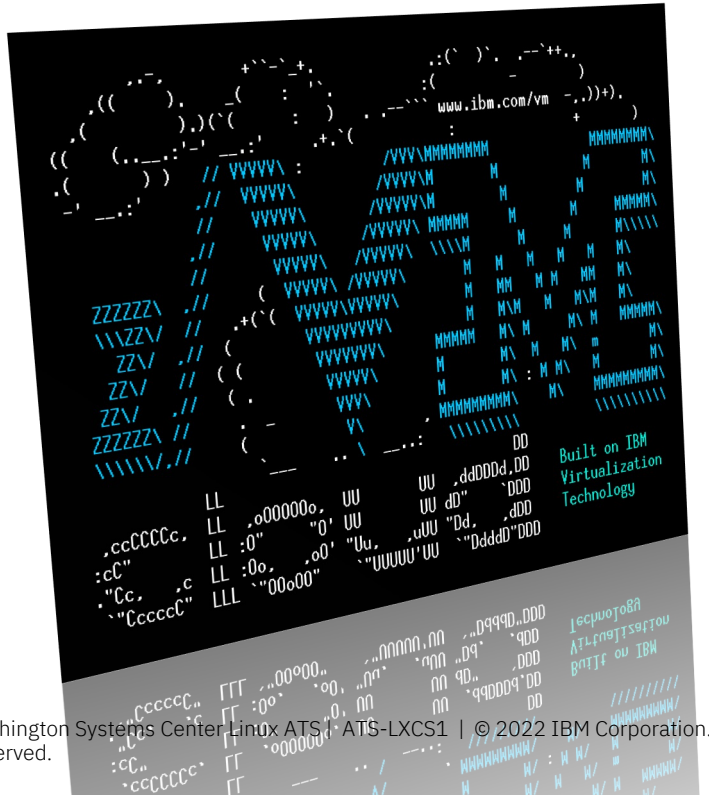


End 4:00 PM Eastern

Workshop Content

Part Two:
Red Hat OpenShift Deep Dive, Good Practices, and Lessons Learned

Virtualization



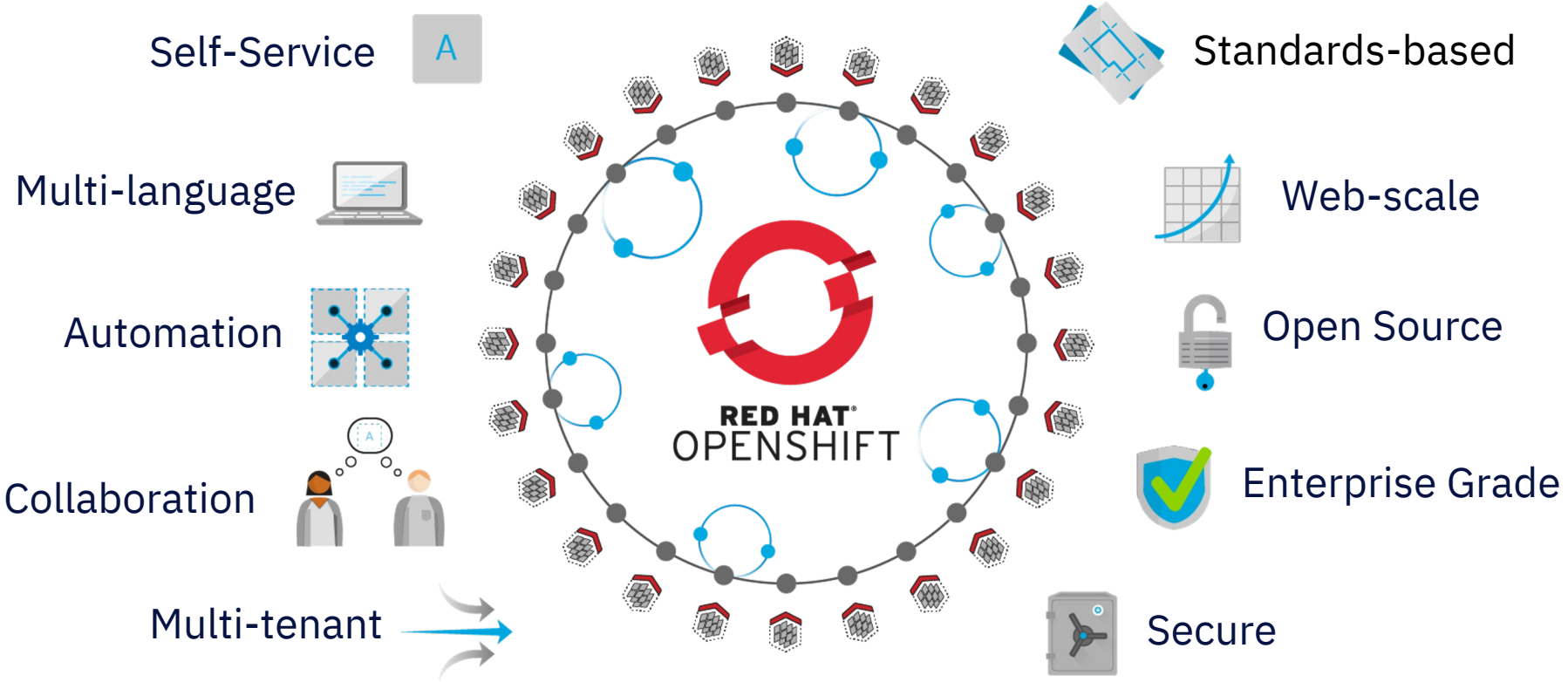
z/VM

KVM

LPAR

zCX

Functional overview

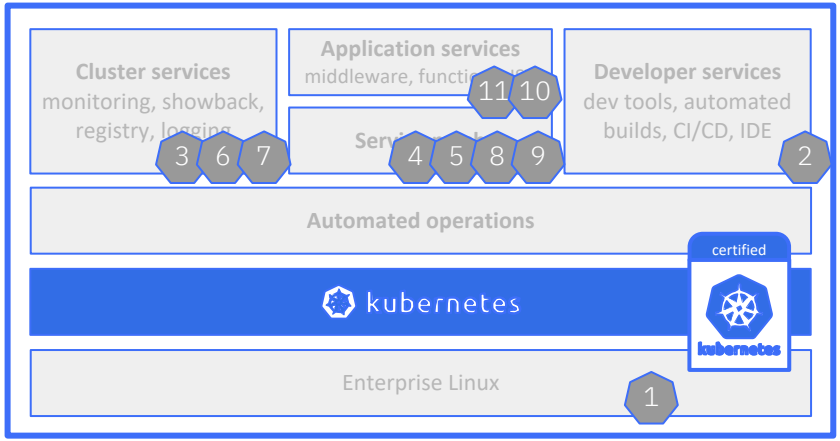


kubernetes

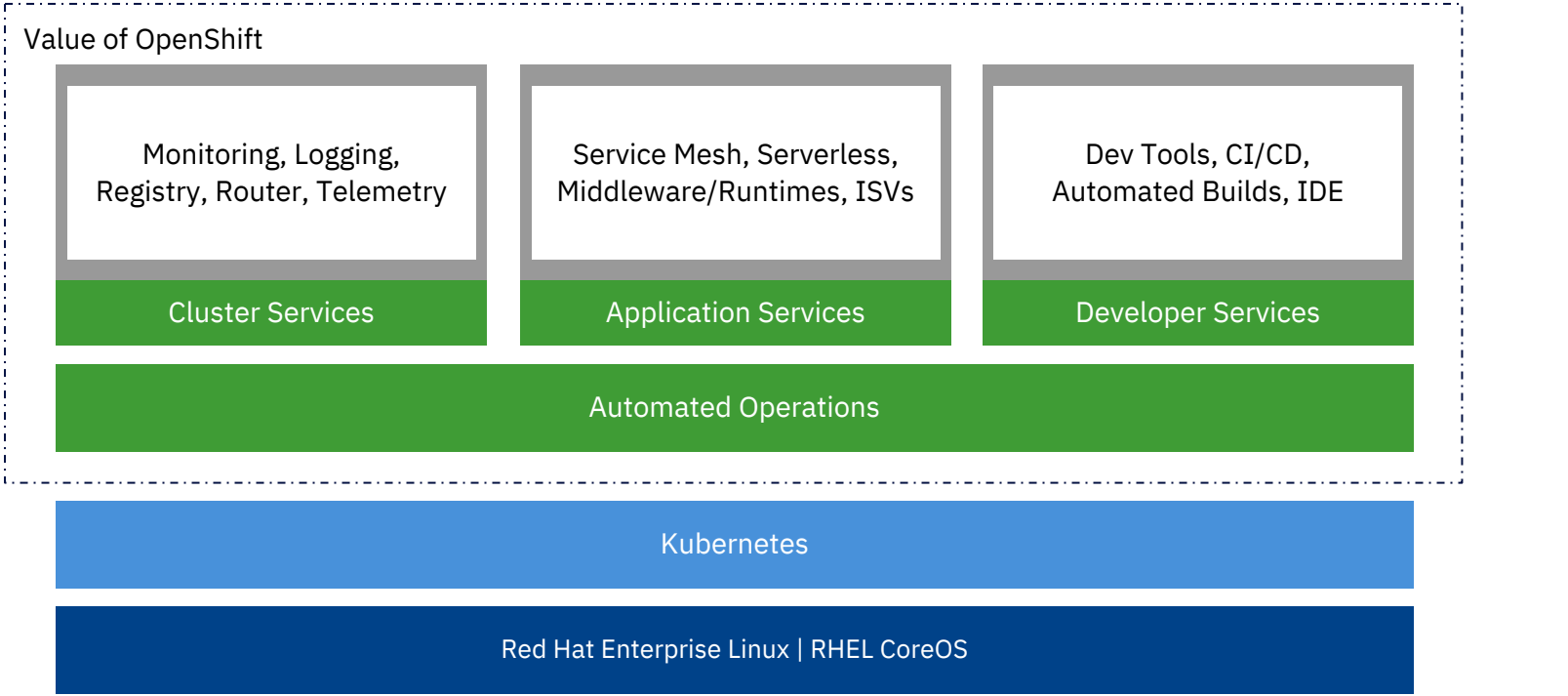
Lacks many essential components

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?

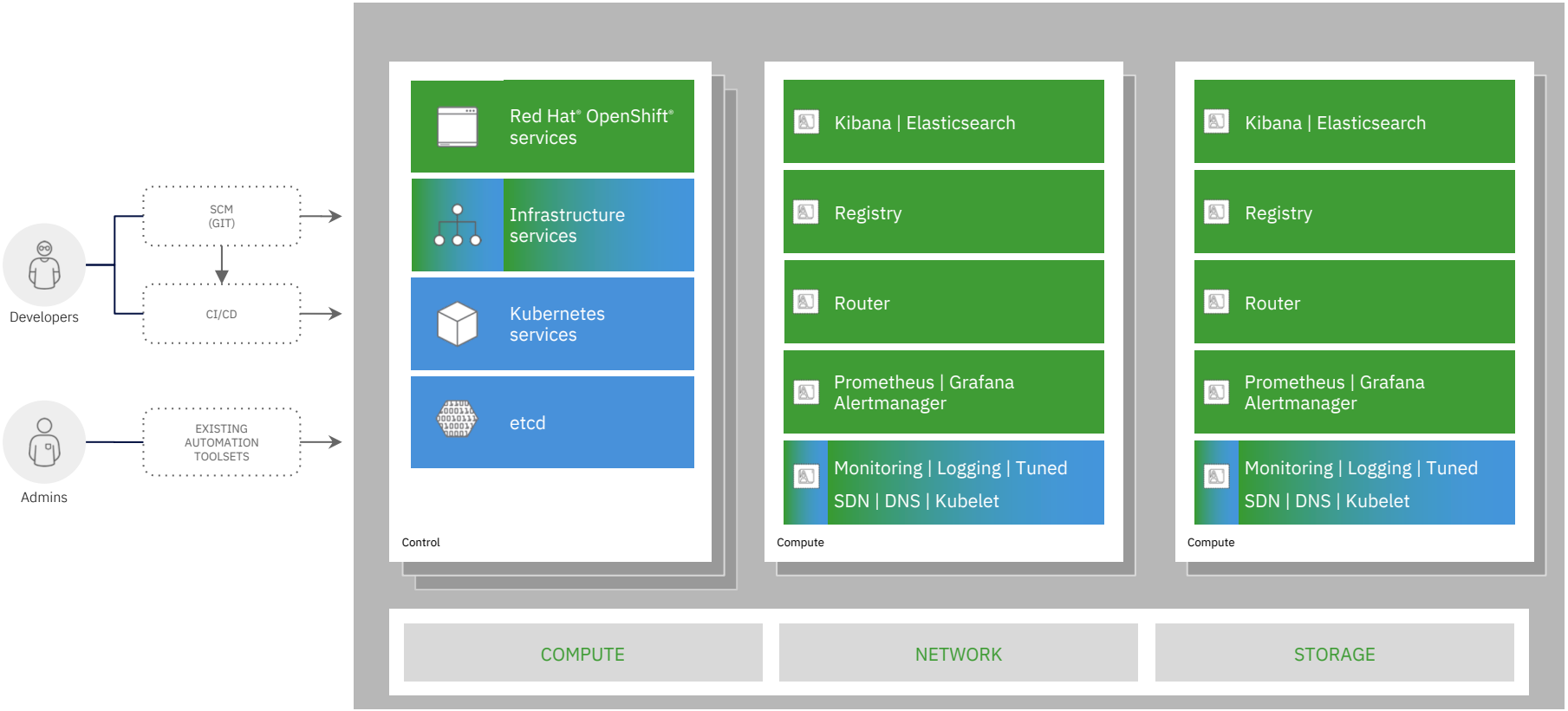


Best IT Ops Experience

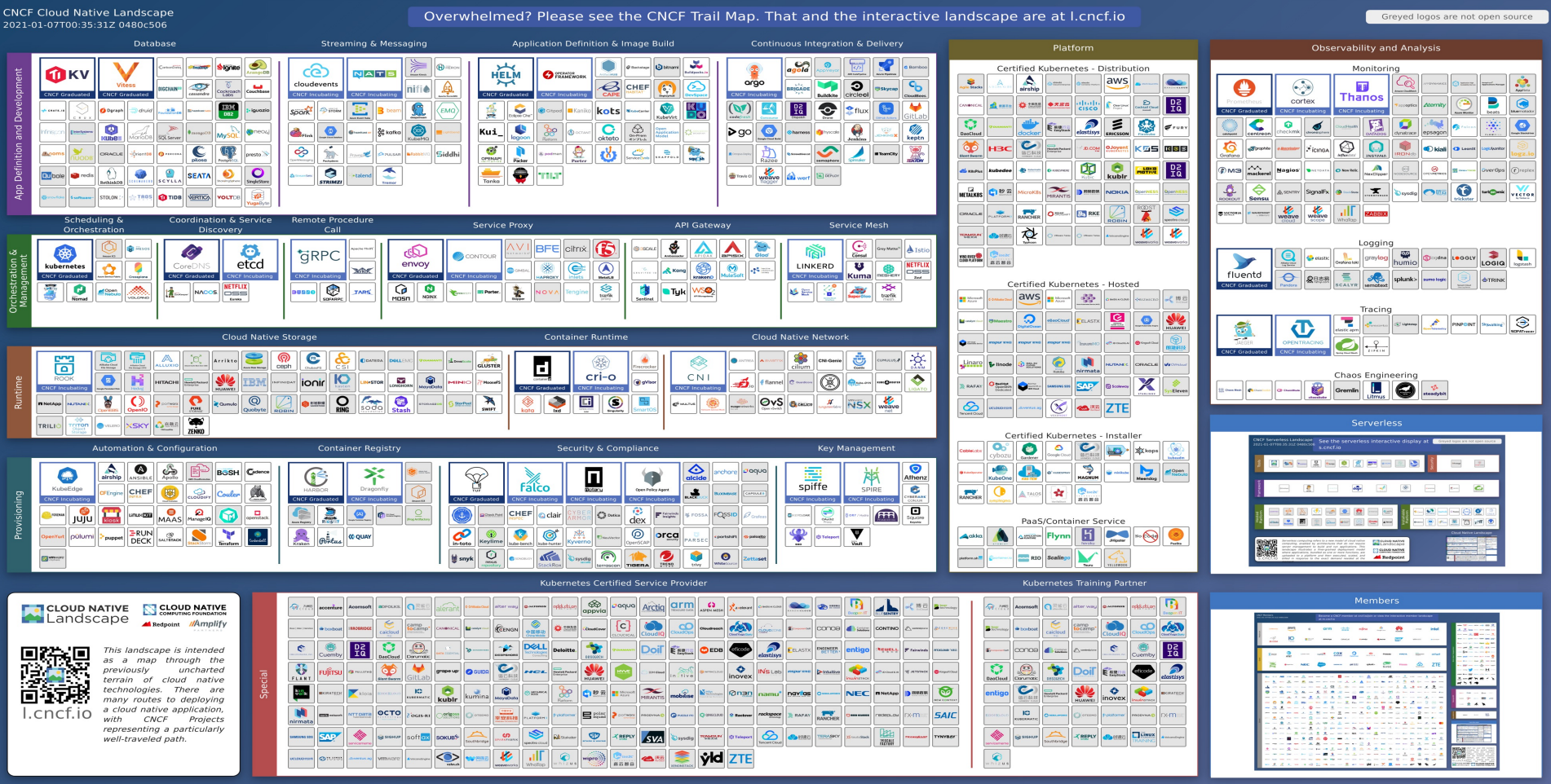
CaaS ⟷ PaaS ⟷ FaaS

Best Developer Experience

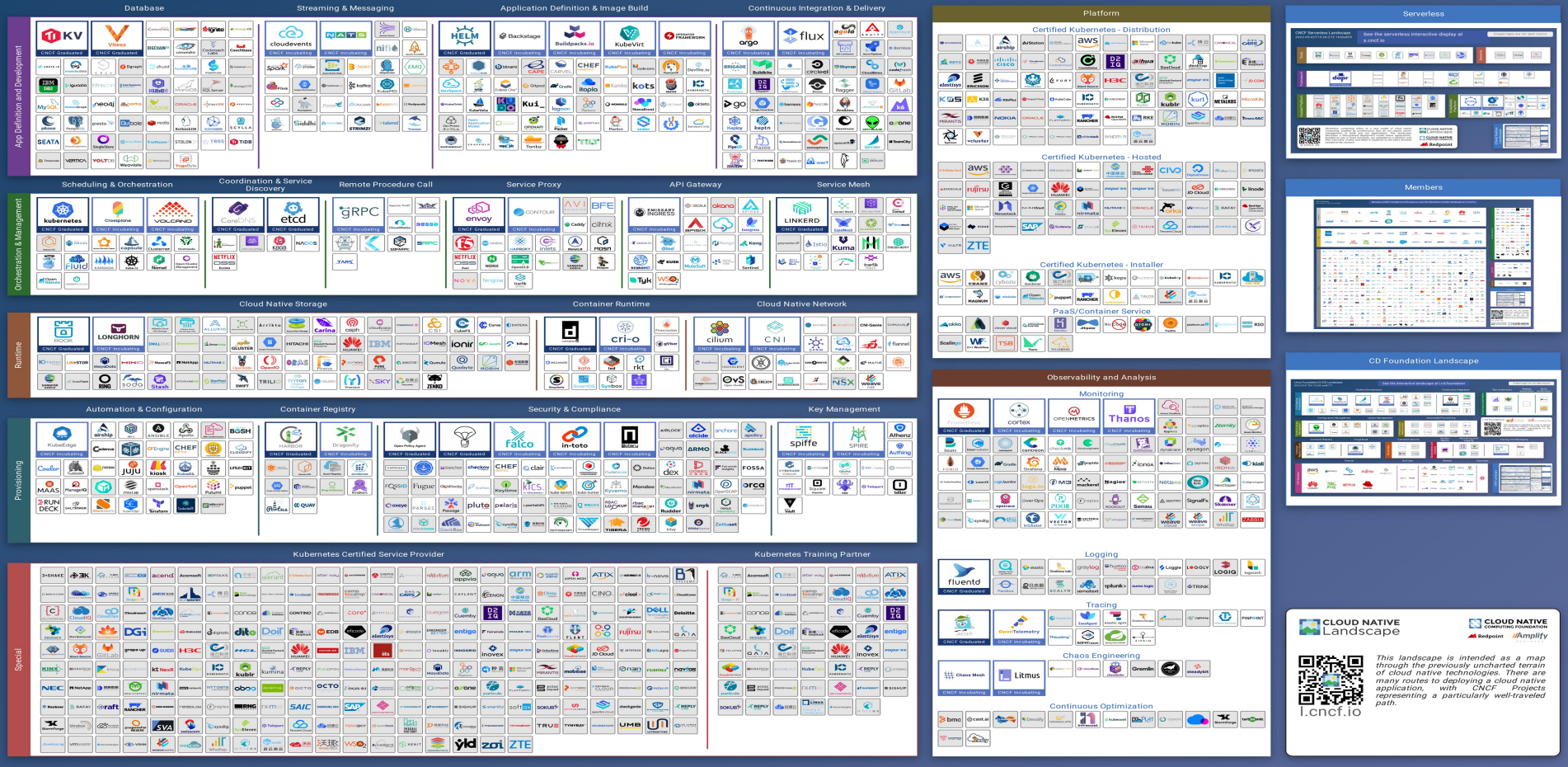
A best-of-breed concept



Architectural overview



CNCF Cloud Native Landscape – January 2021



CNCF Cloud Native Landscape – May 2022

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.

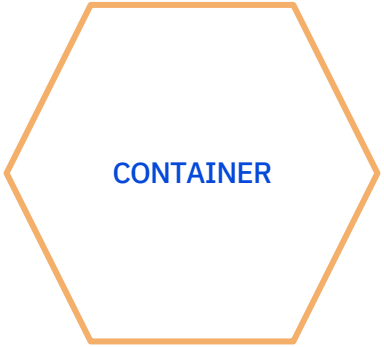
containerd — CNCF Graduated
HARBOR — CNCF Incubating
cri-o — CNCF Incubating

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

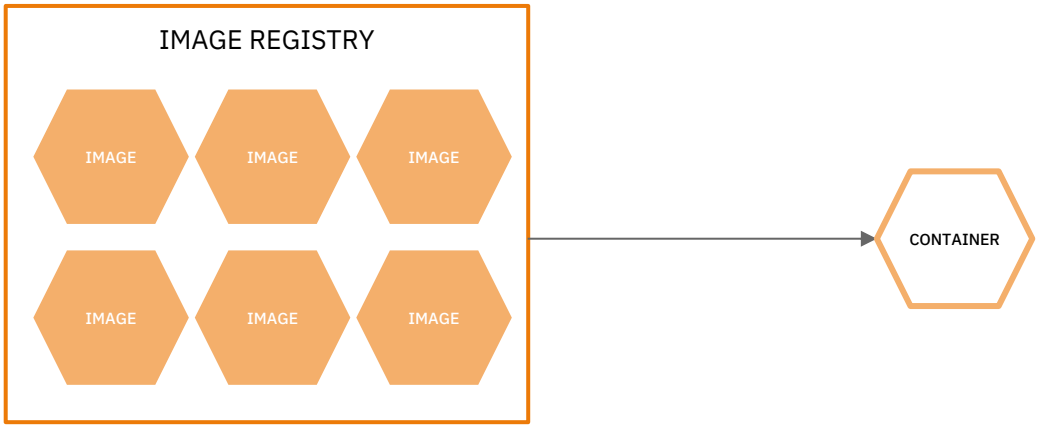
OpenShift and Kubernetes core concepts



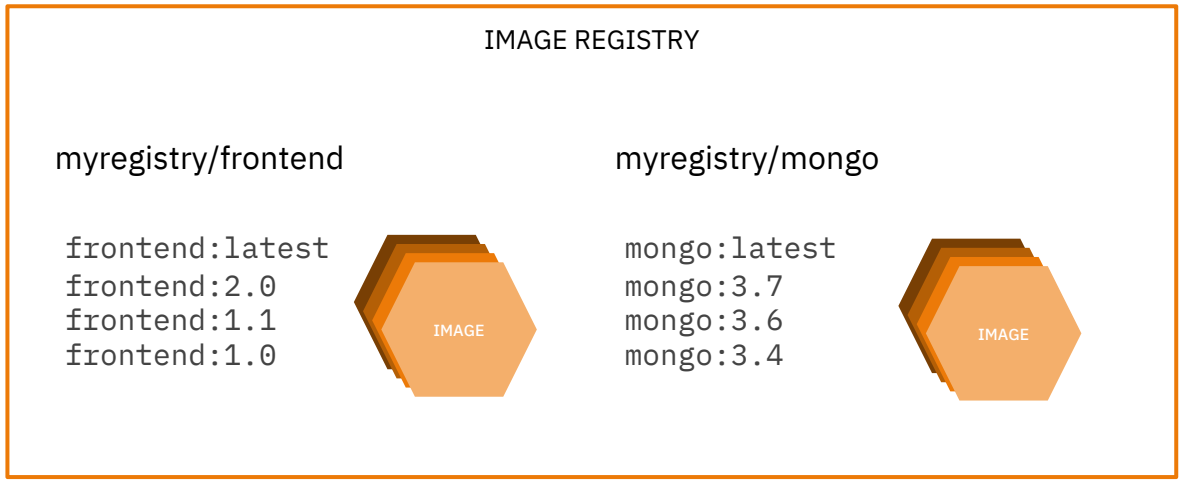
A container is the smallest compute unit

containers are created from container images

container images are stored in an image registry

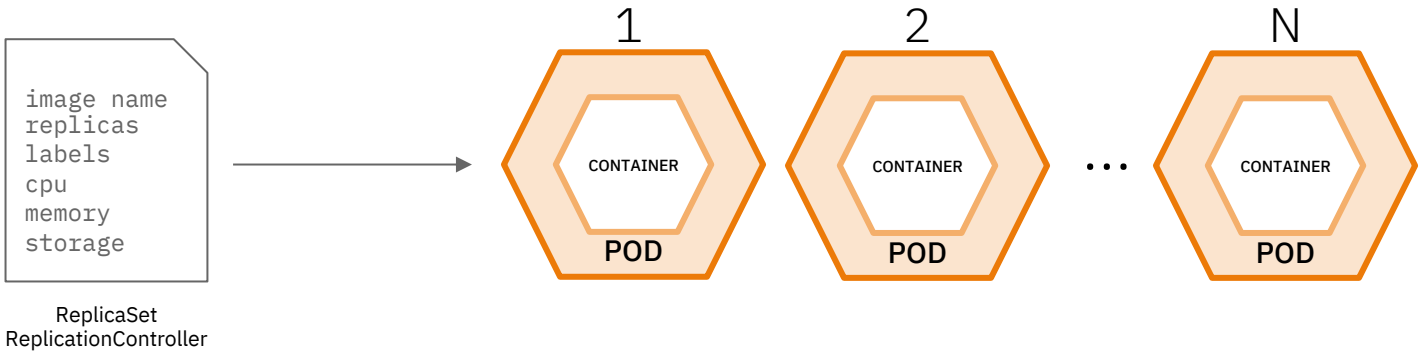


an image repository contains all versions of an image in the image registry

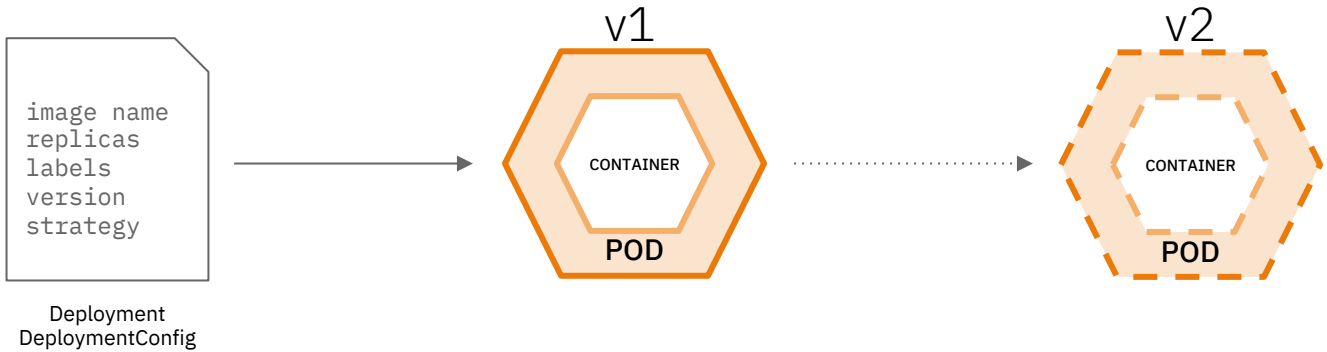


containers are wrapped in pods which are units of deployment and management

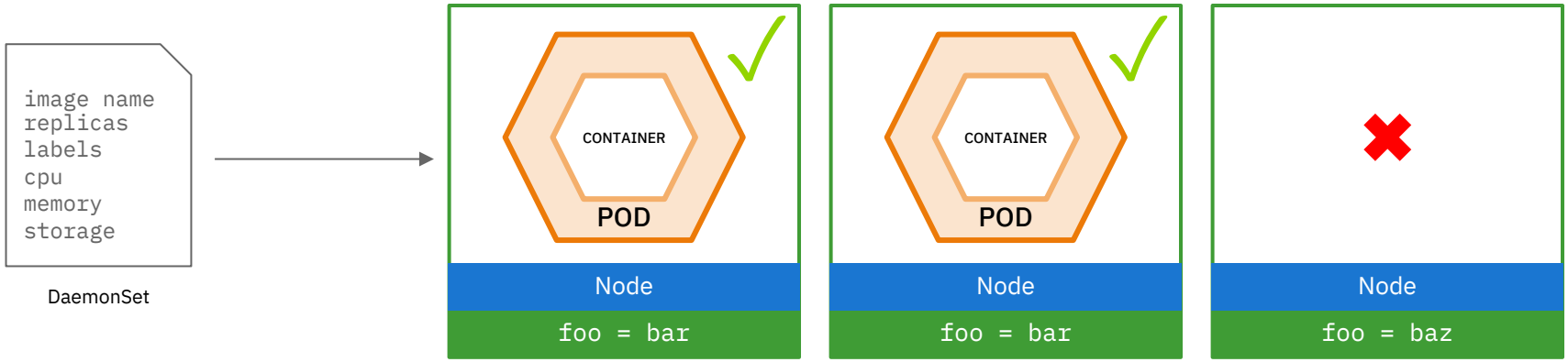
ReplicationControllers & ReplicaSets ensure a specified number of pods are running at any given time



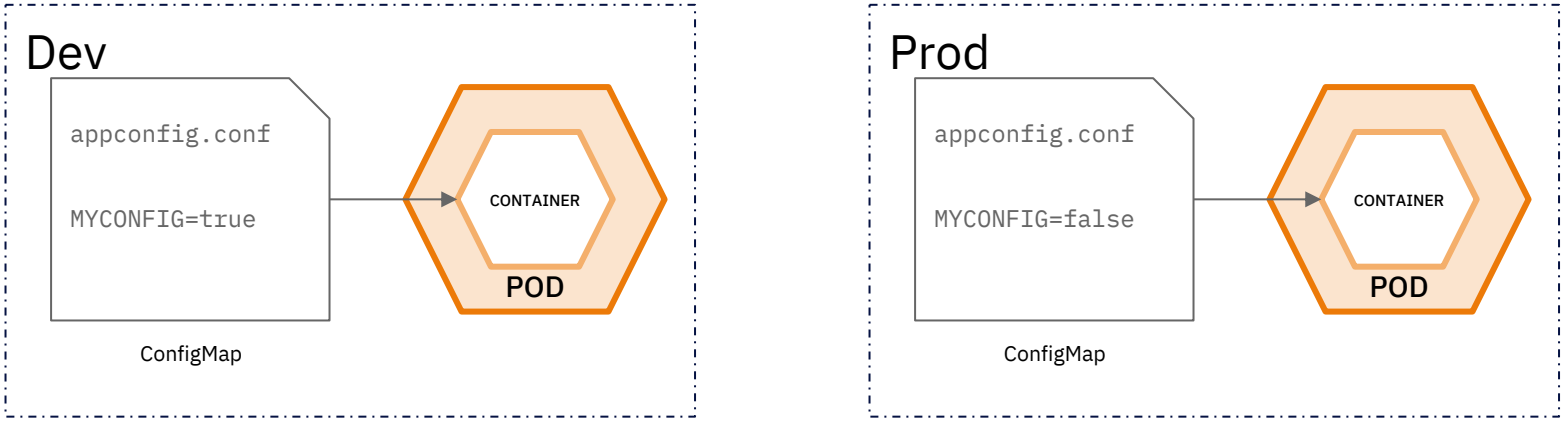
Deployments and DeploymentConfigurations define how to roll out new versions of Pods



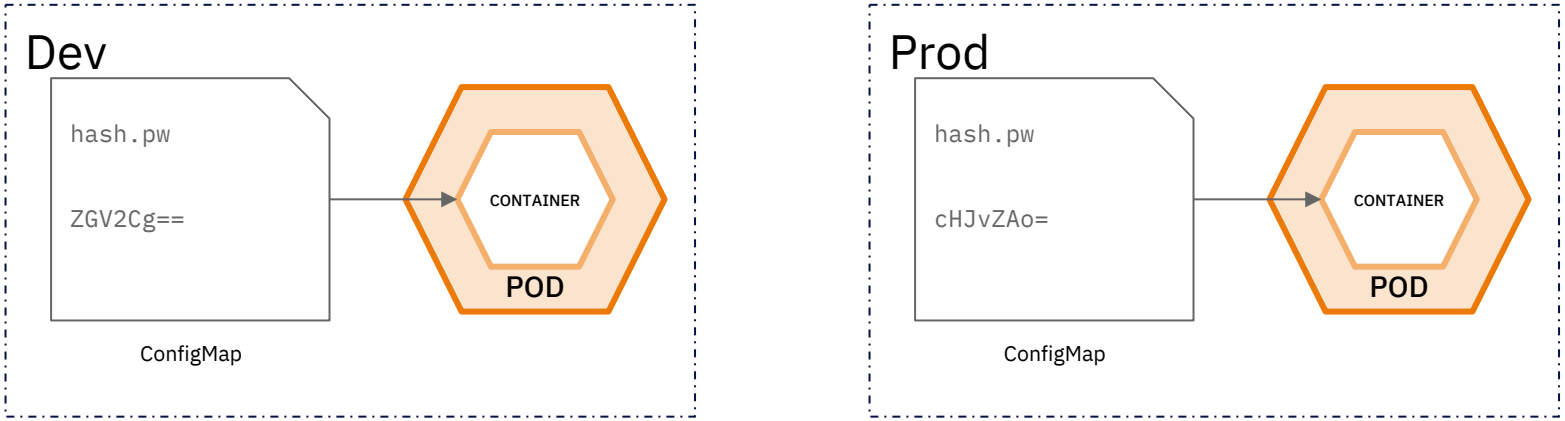
a daemonset ensures that all (or some) nodes run a copy of a pod

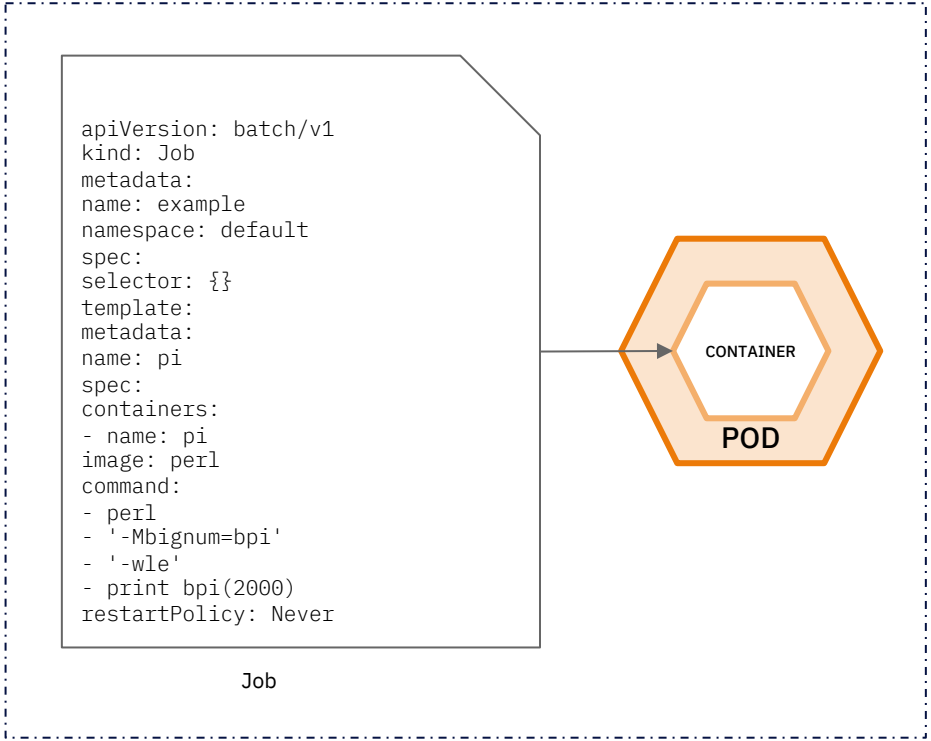


`configmaps` allow you to decouple configuration artifacts from image content

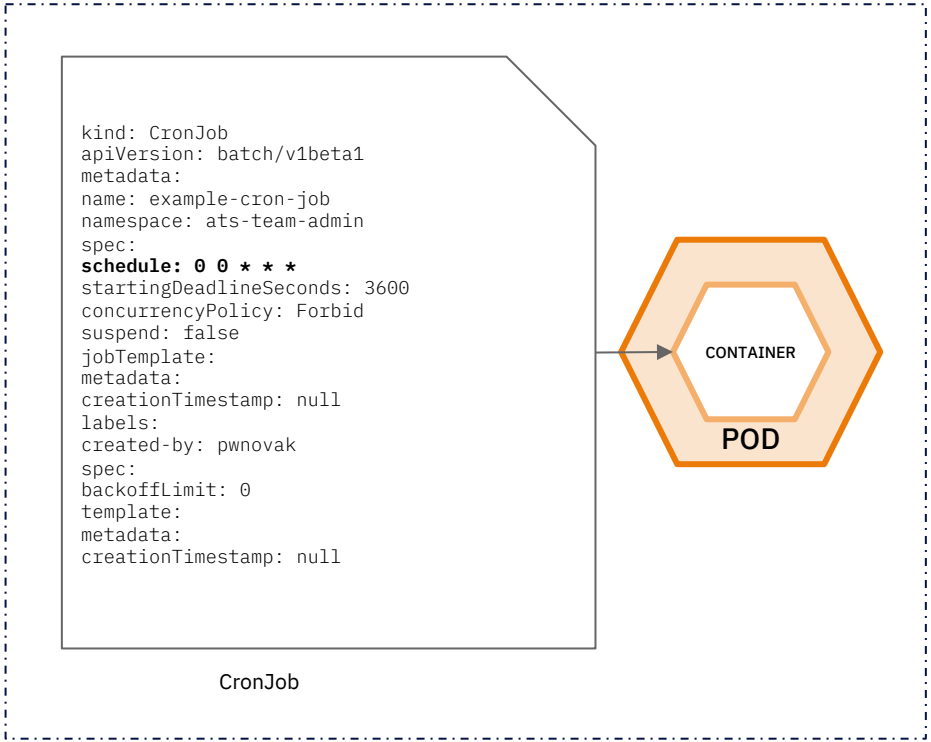


`secrets` provide a mechanism to hold sensitive information such as passwords





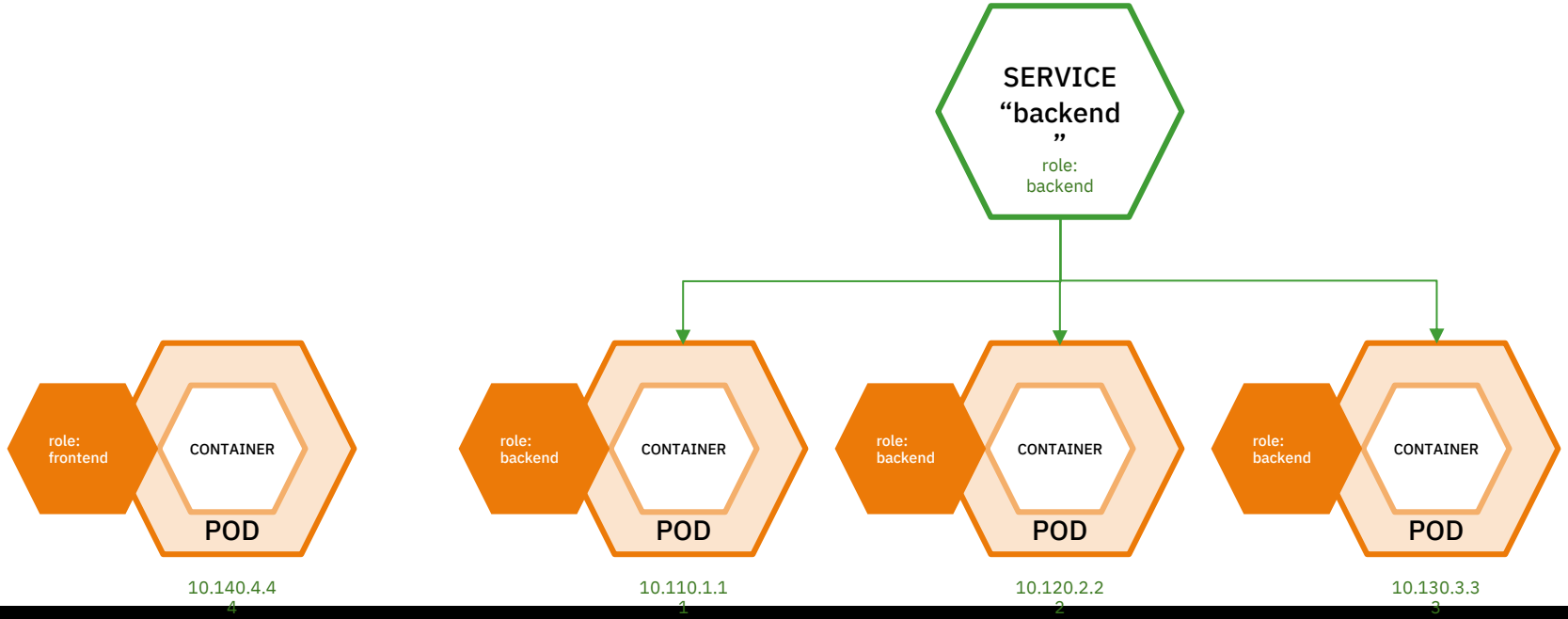
jobs are batch tasks that can be run either manually or via the cluster crontab.



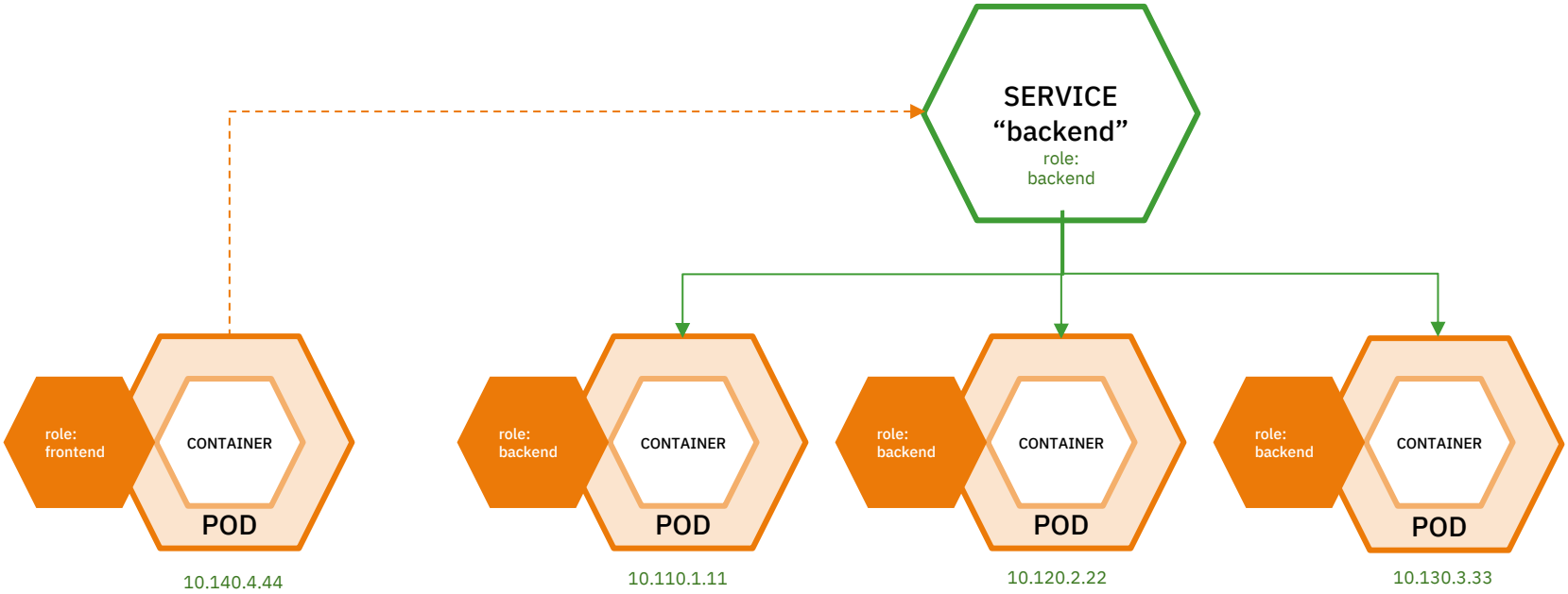
cronjobs are batch tasks run on a defined schedule via the cluster crontab.

Tip: You MUST stagger your scheduling!

services provide internal load-balancing and service discovery across pods

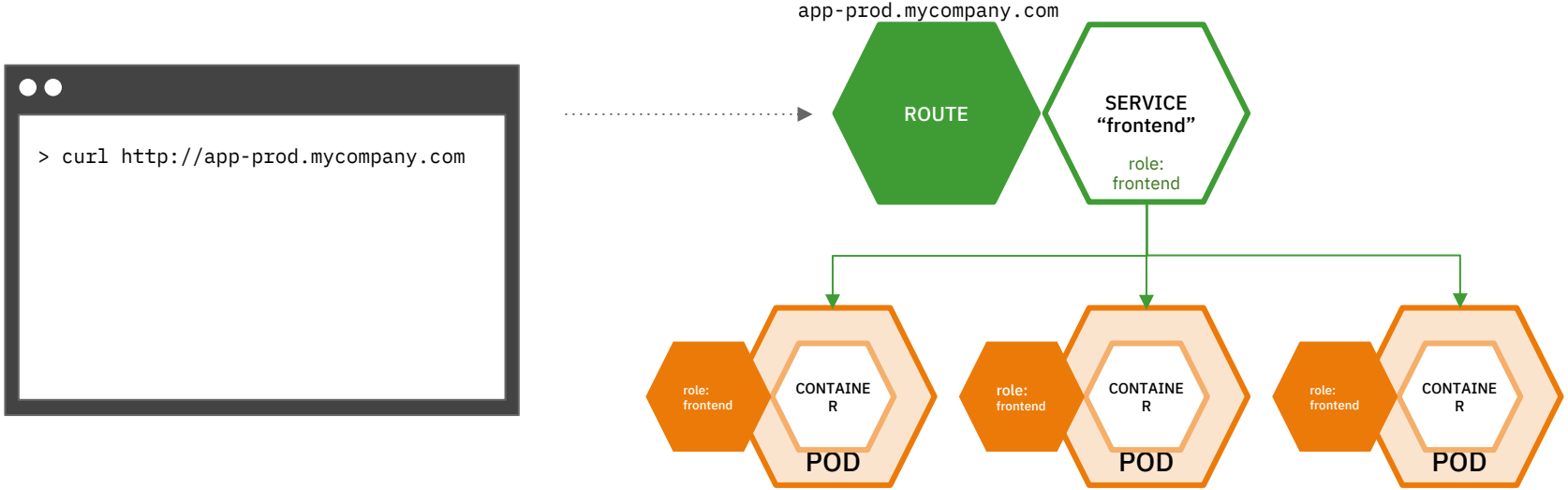


apps can talk to each other via services

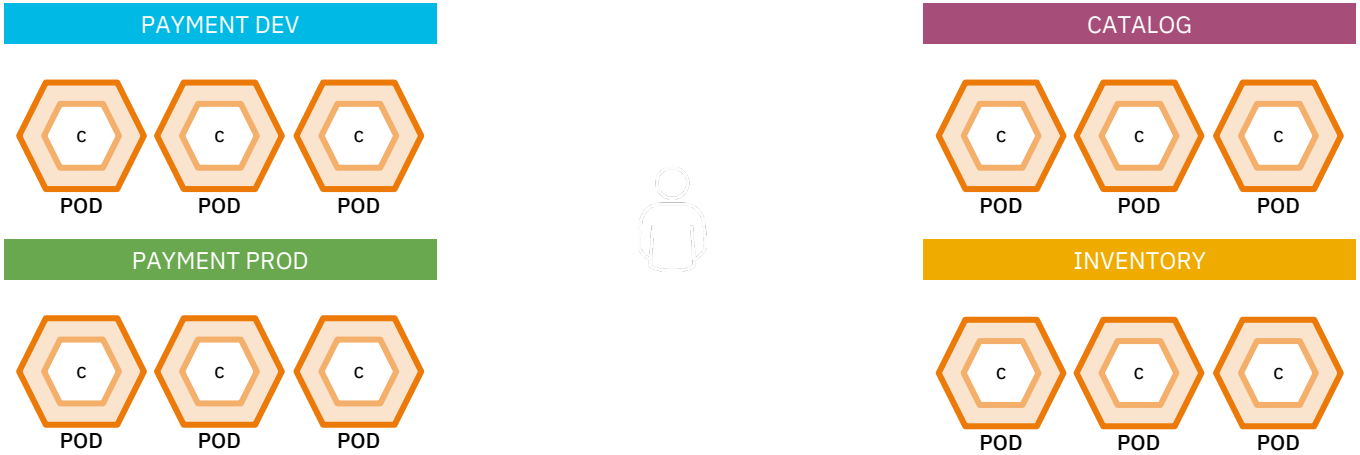


Service chatter

`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.

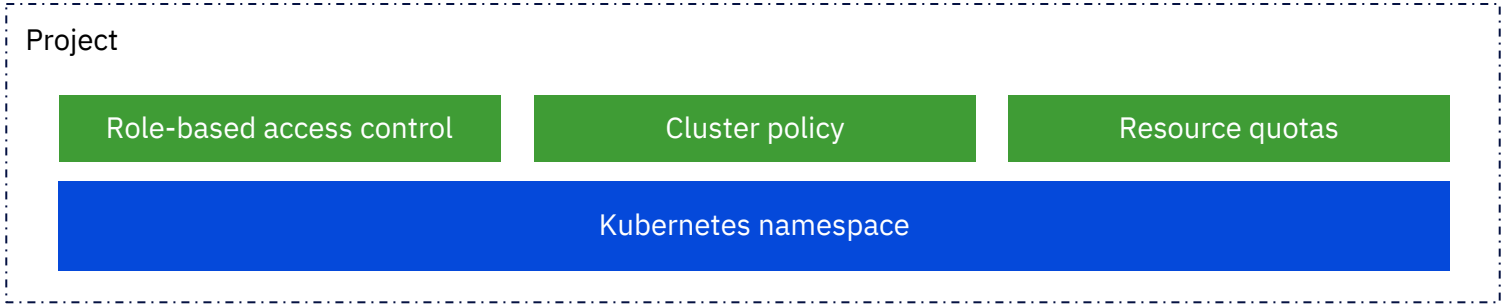


Namespaces were designed as a construct for cluster resource management, not security.

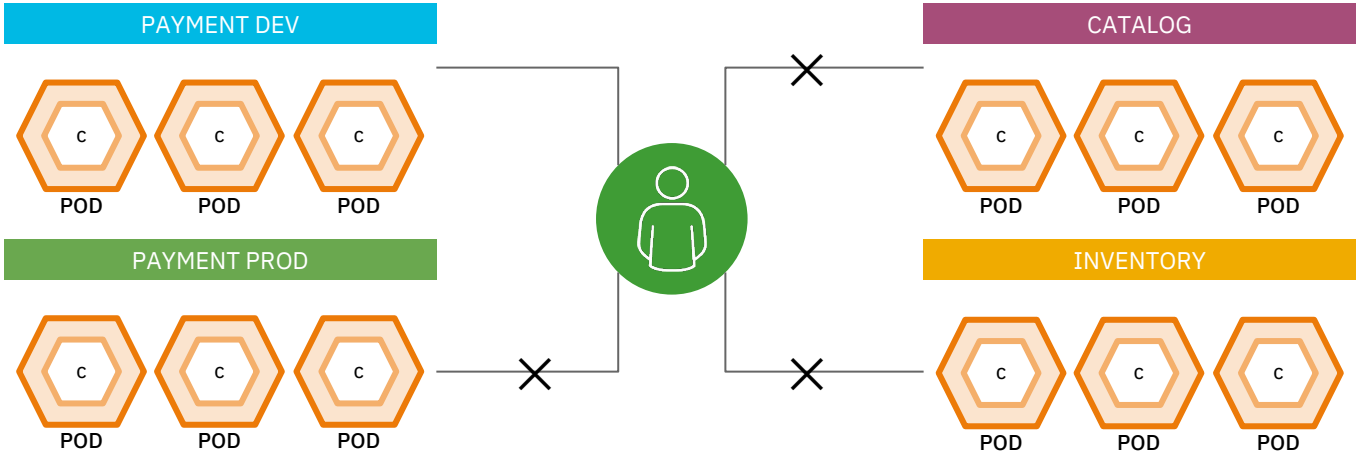
Do not rely on namespaces as a security feature outside of cluster internals within trusted domains.

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

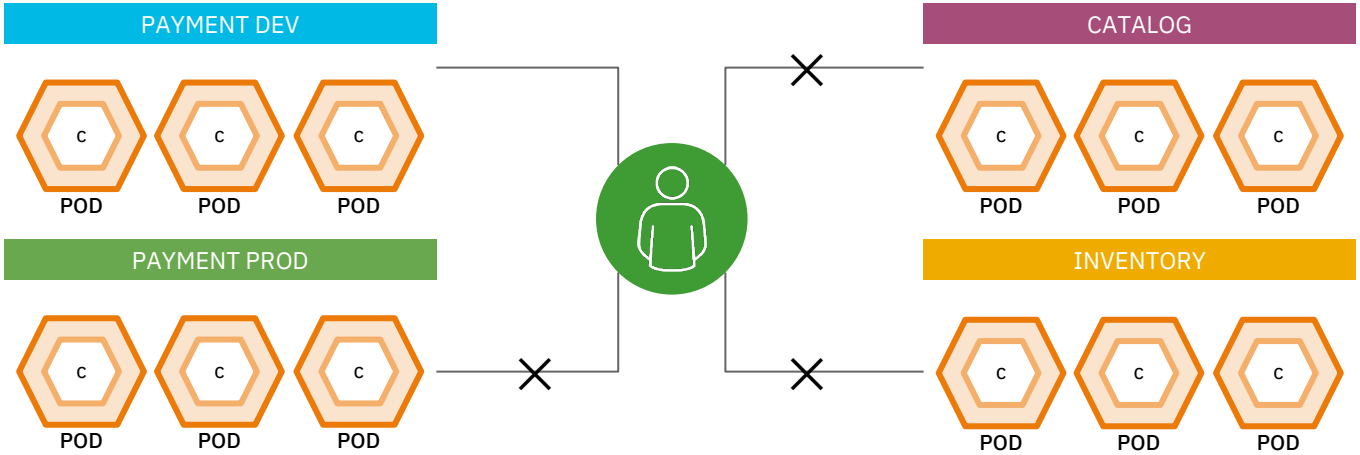
A namespace plus the RBAC layer and some other enhancements is a project



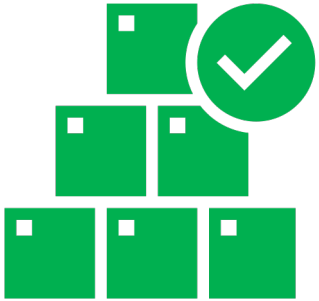
Projects isolate apps across environments, teams, groups and departments

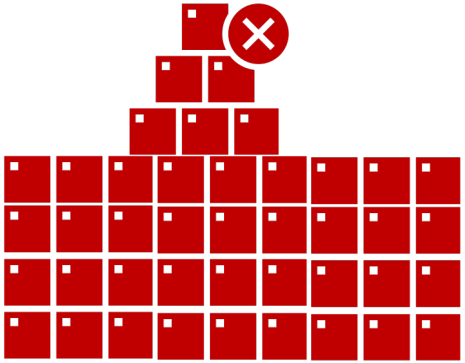


IBM Z and LinuxONE are the only platform where SECURE multi-tenant usage is possible



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





OpenShift 4 Architecture

COMPUTE

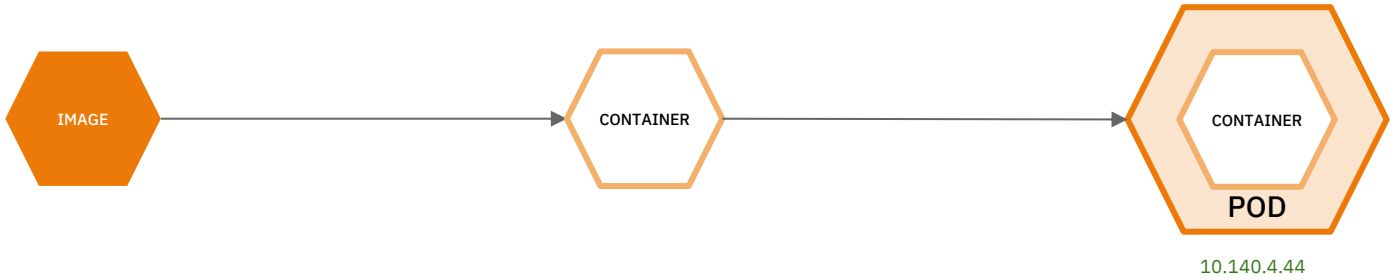
NETWORK

STORAGE

your choice of infrastructure

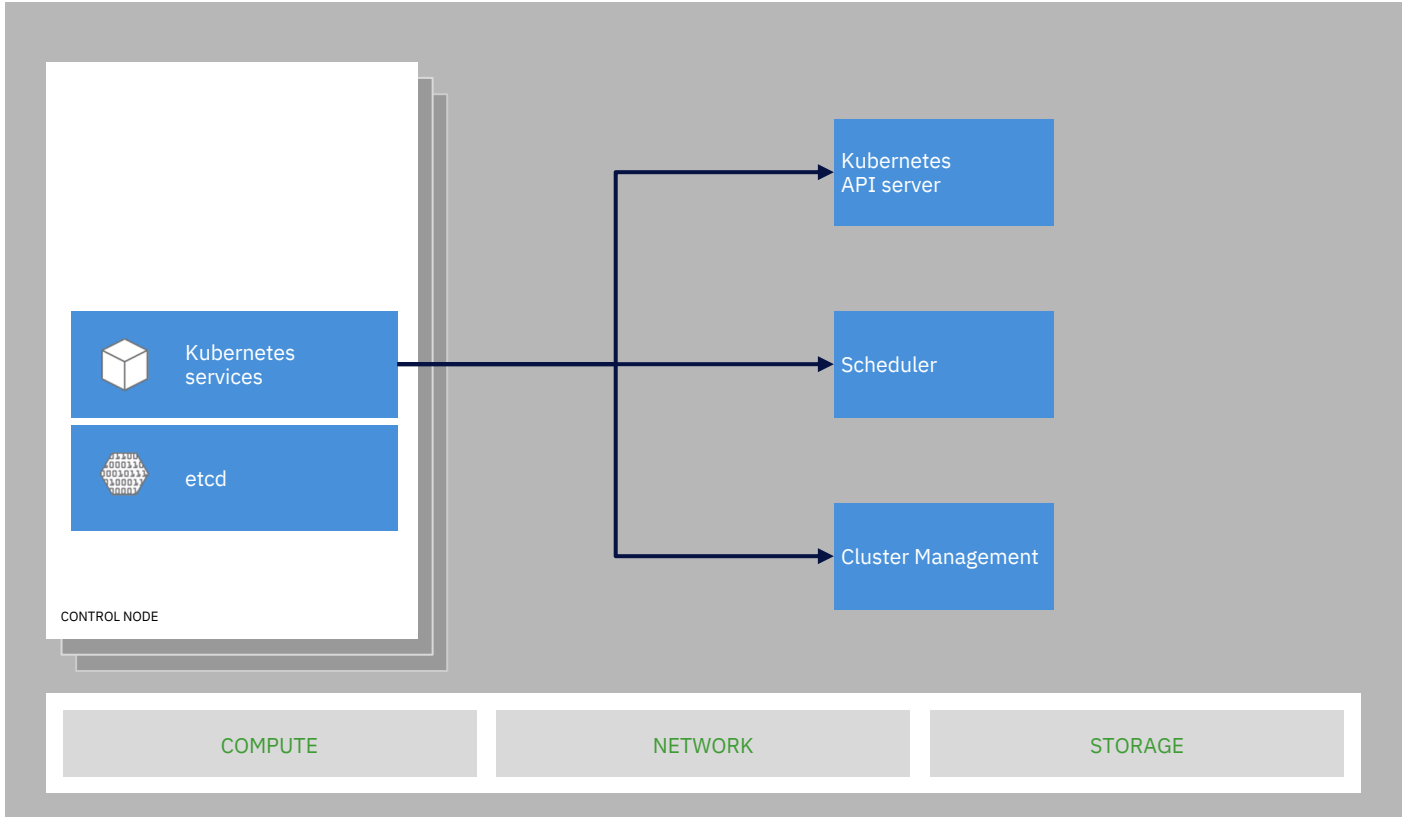
Compute nodes run workloads

Control nodes

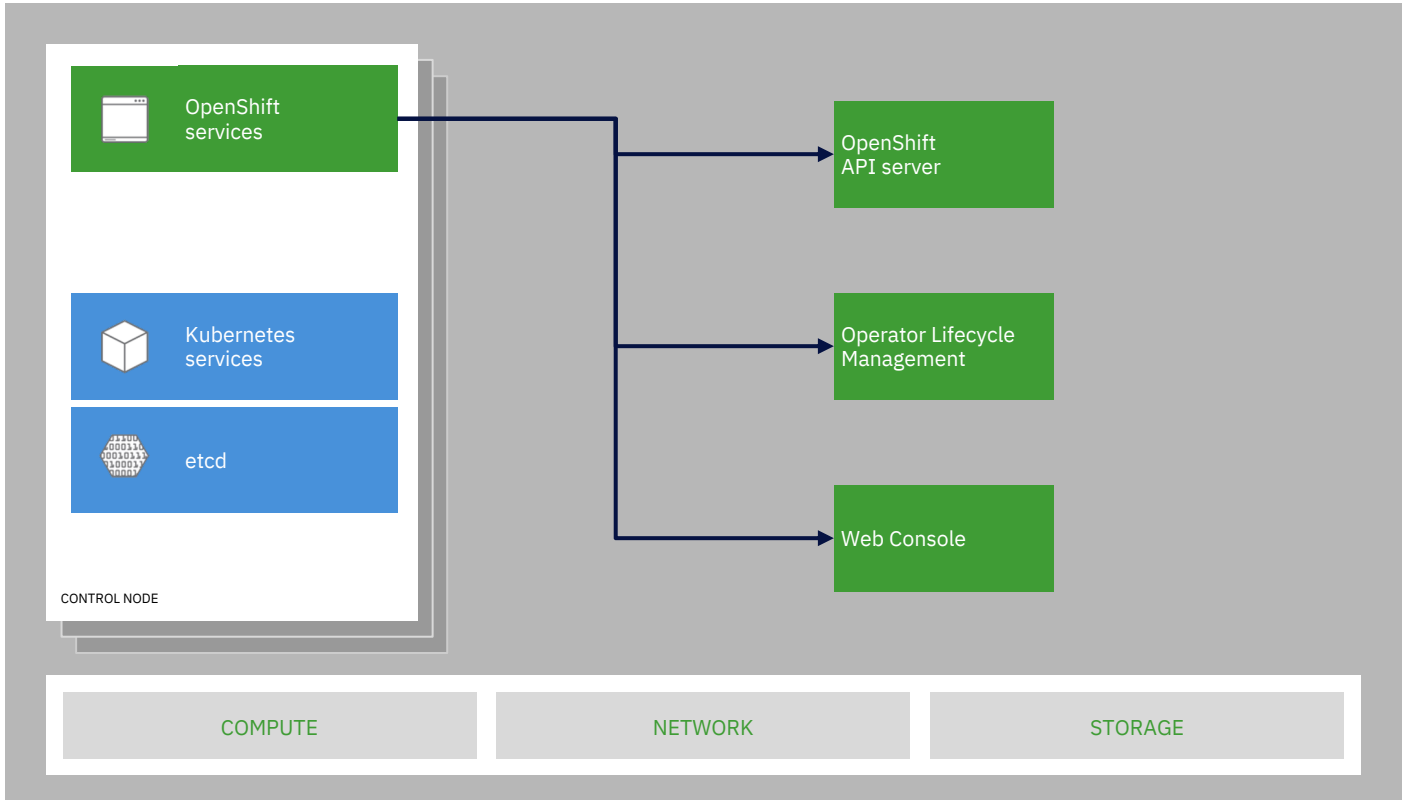


everything runs in pods

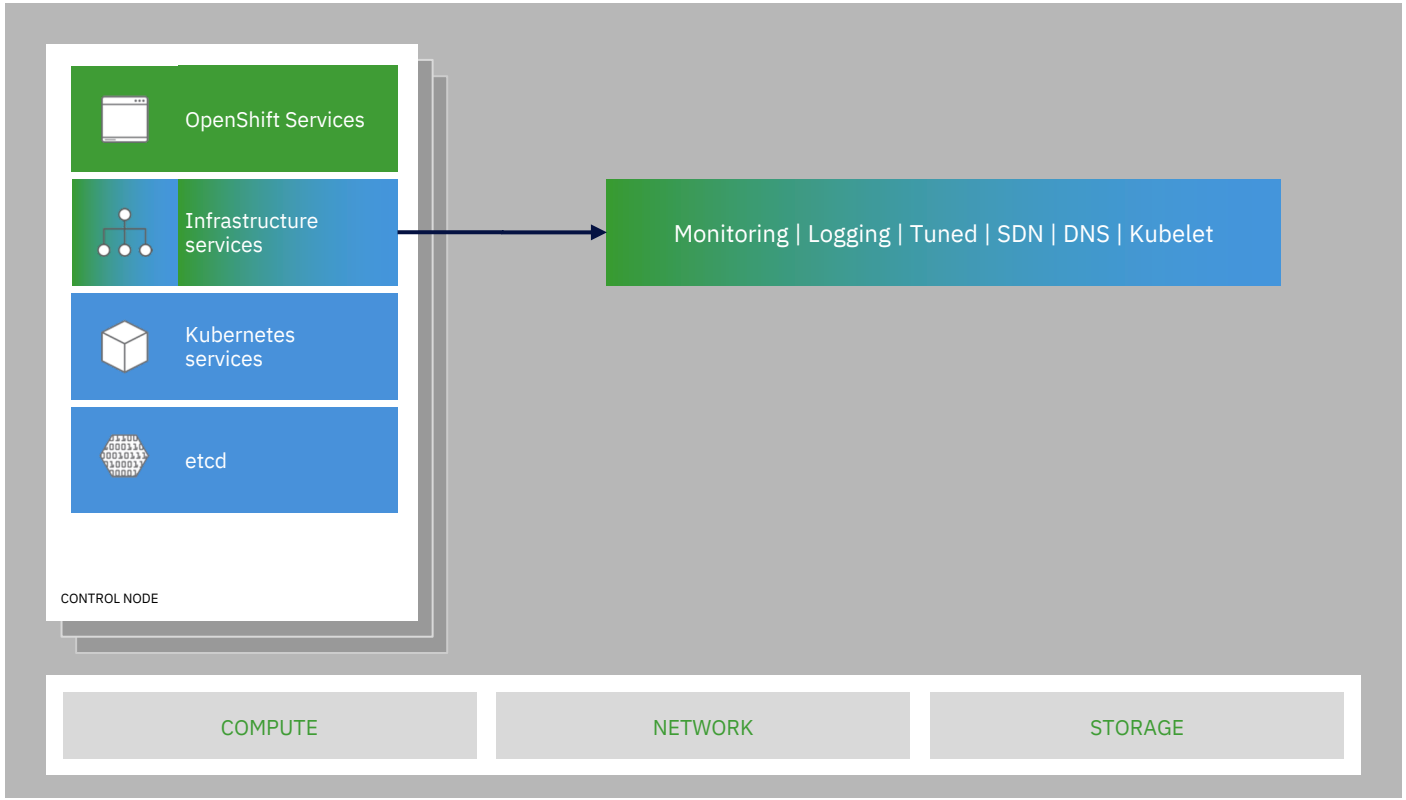
etcd



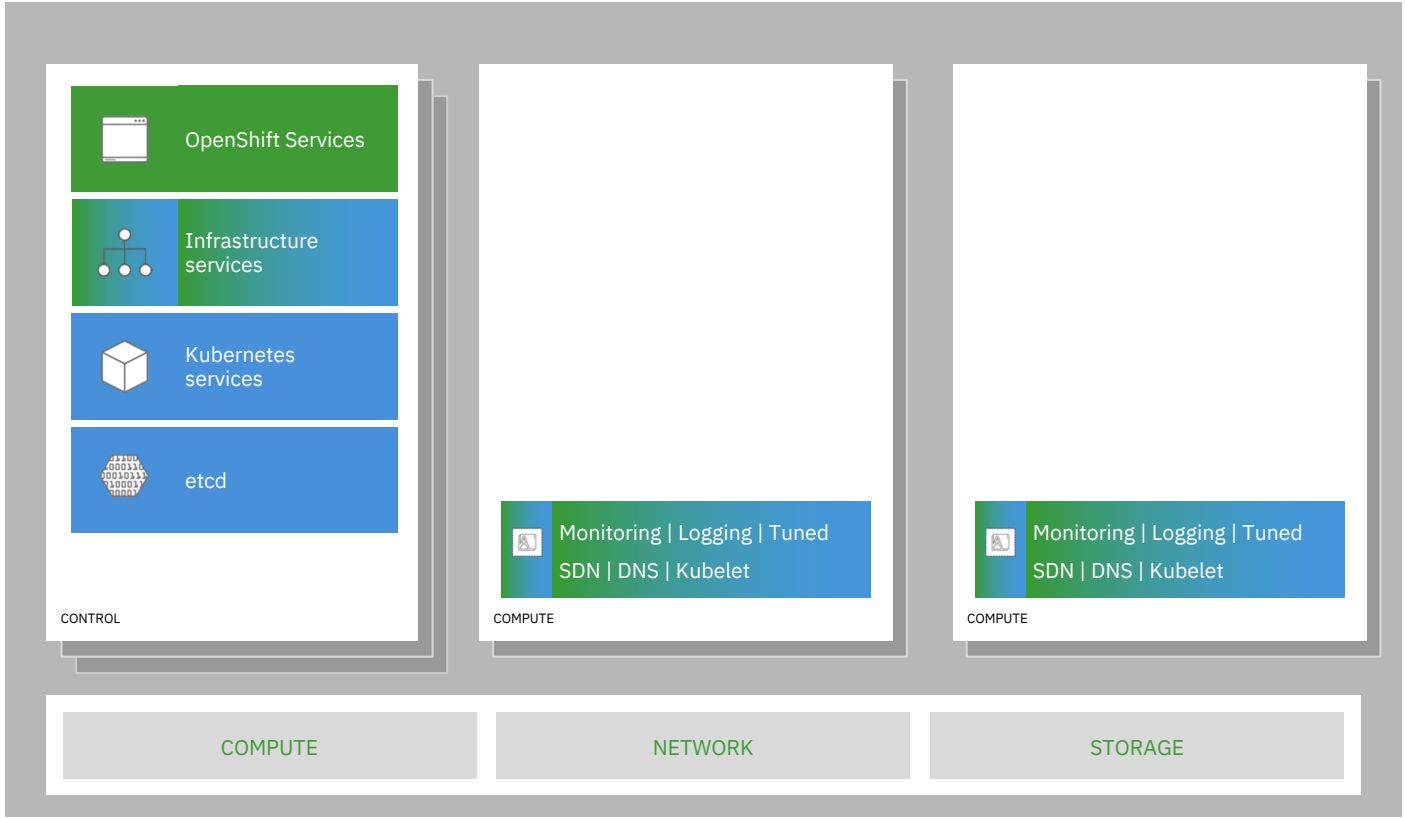
core kubernetes components



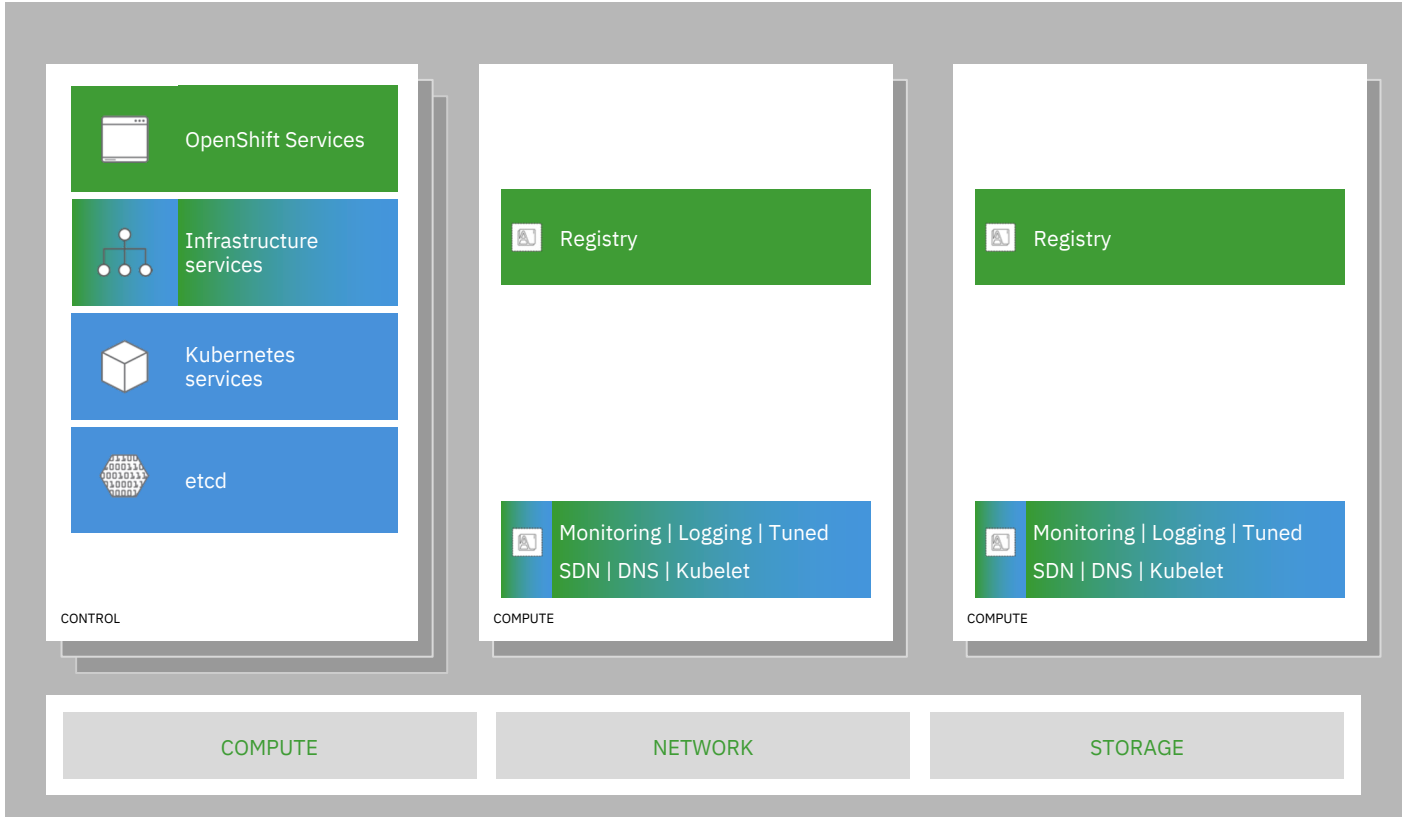
core OpenShift components



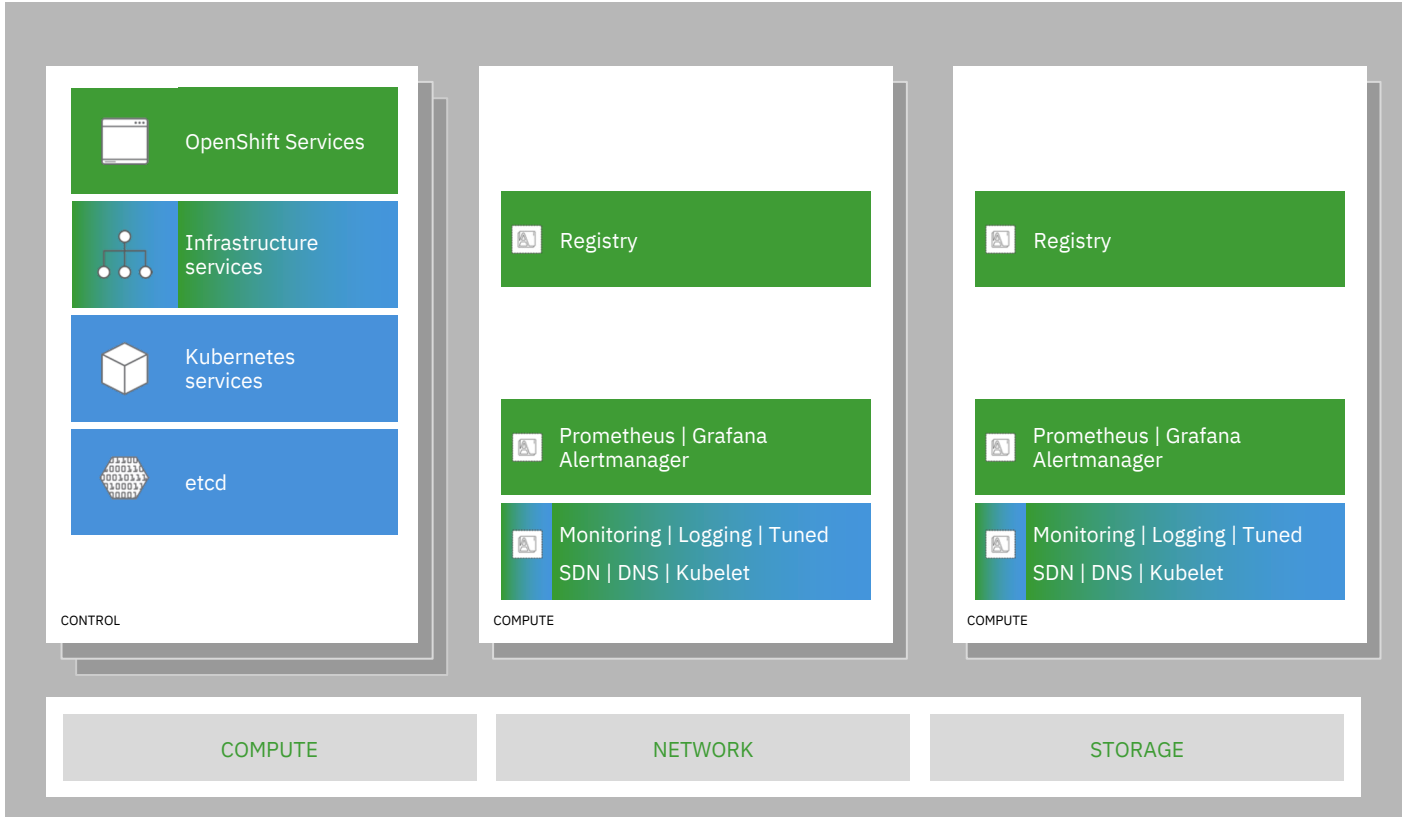
internal and support infrastructure services



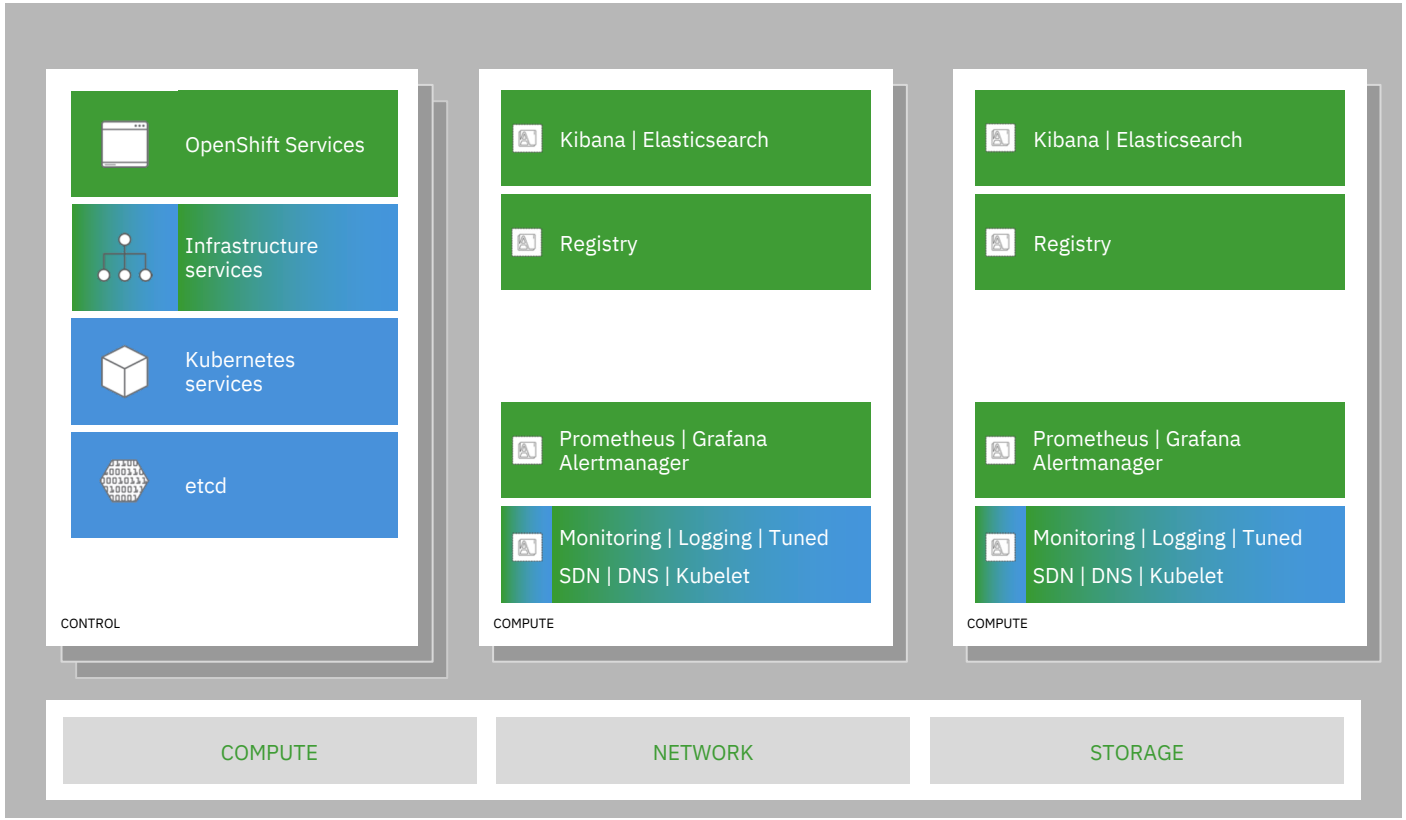
run on all hosts



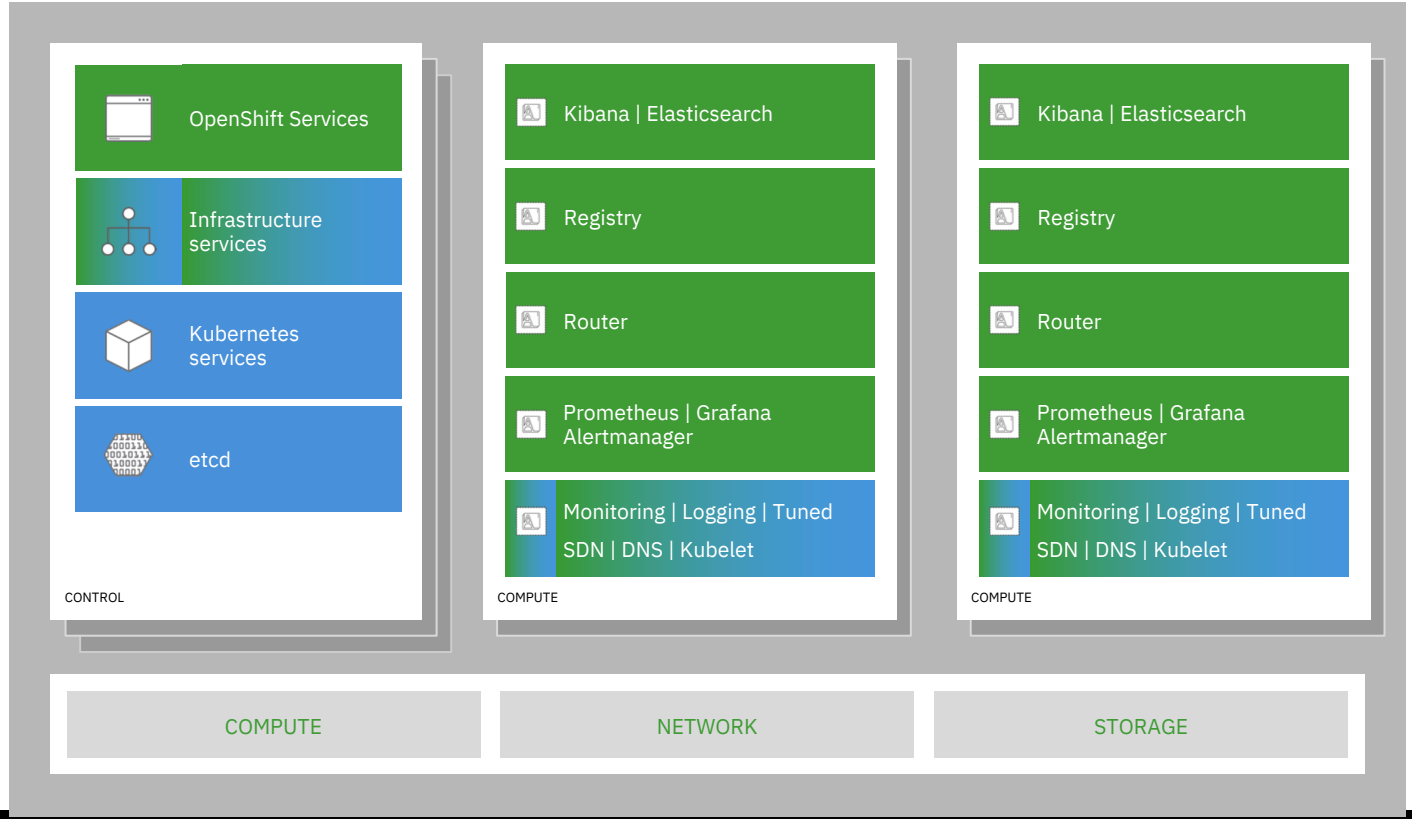
integrated image registry



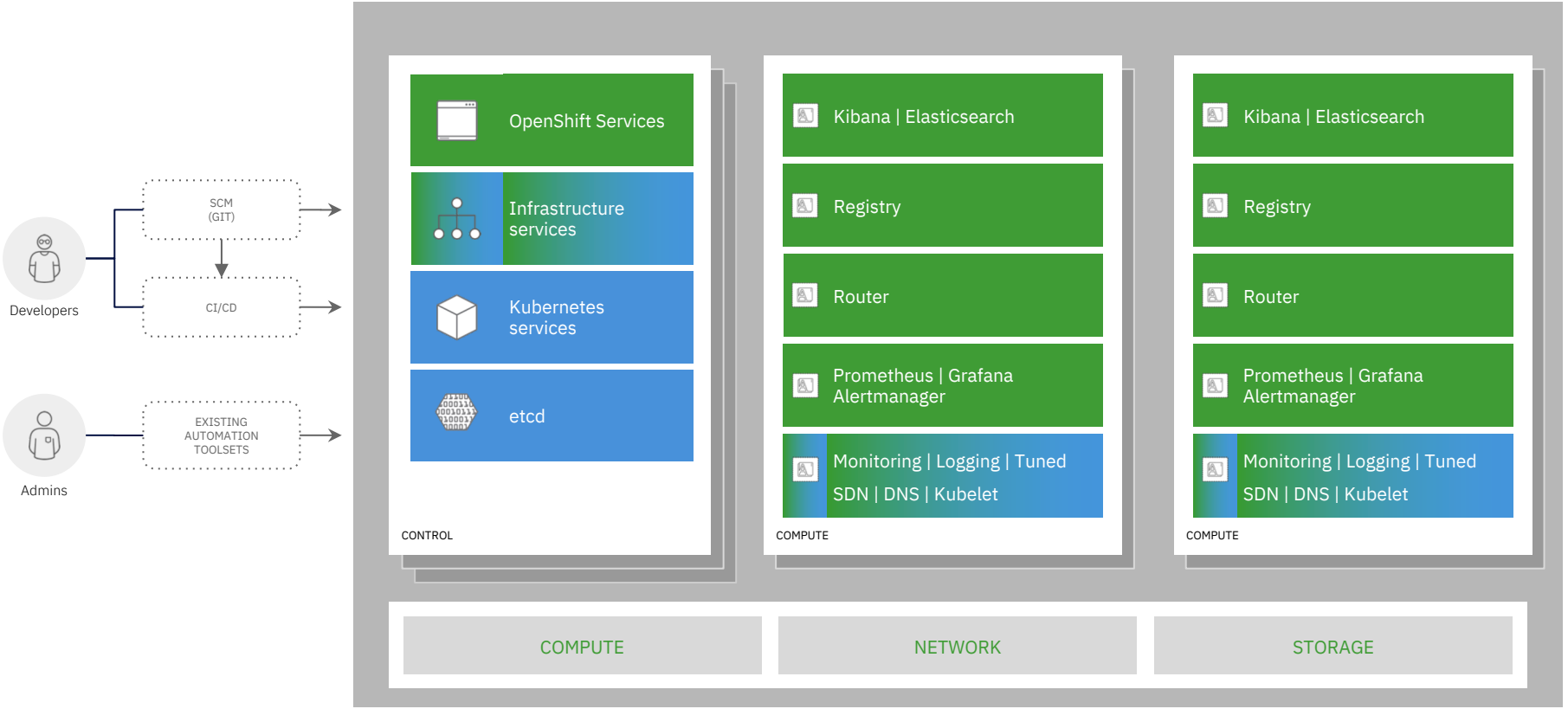
cluster monitoring



log aggregation



integrated routing



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

Good practices and lessons learned

Creating multi-architecture deployments is good practice and should be considered mandatory for your container journey.

Fit for purpose is a fundamental criterion

What should (or shouldn't) I think about containerizing?

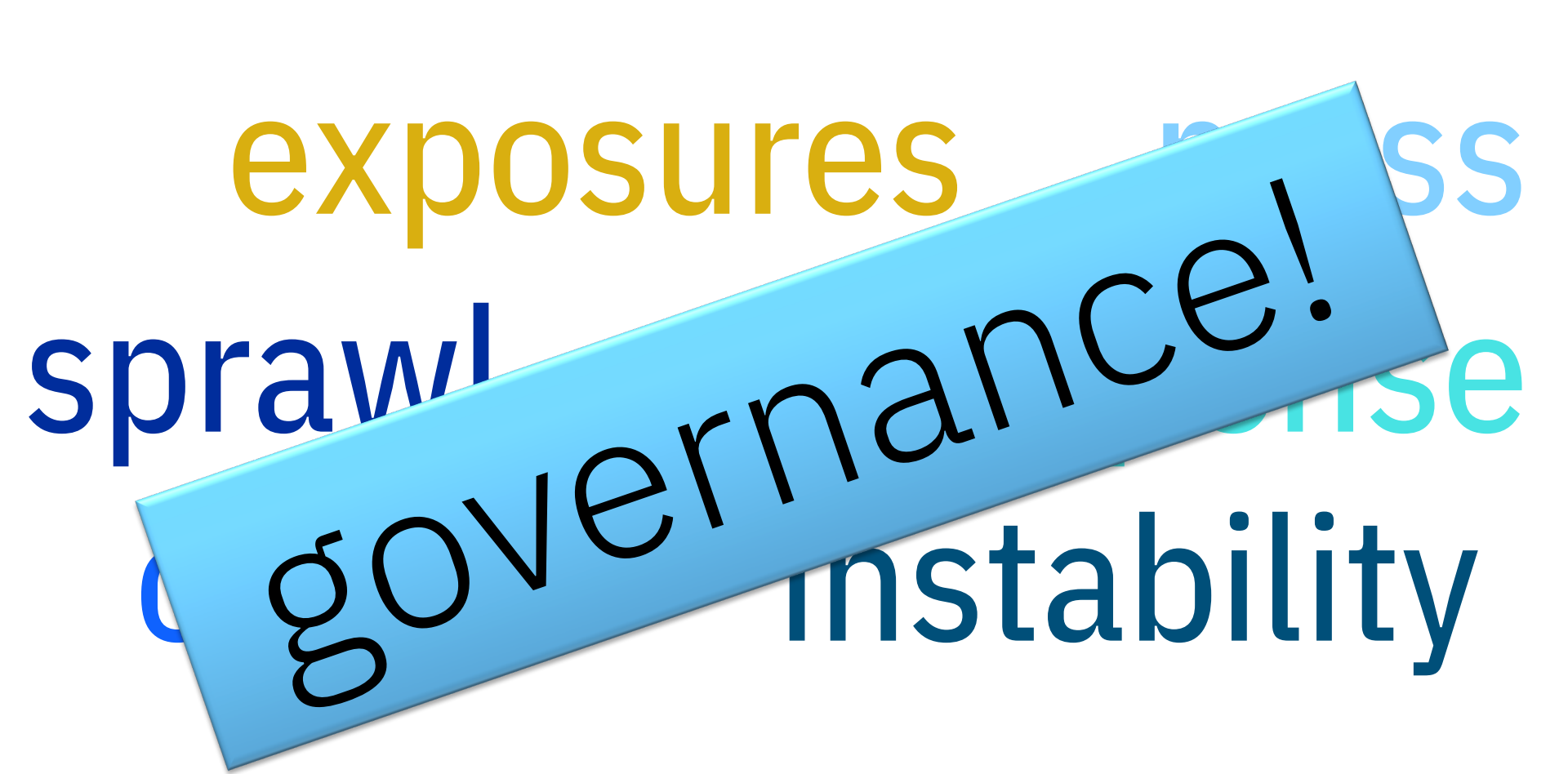
Web middleware / J2EE

Messaging and integration such as Kafka / EventStreams

HTTP content

Anything that needs to be able to rapidly scale up to handle a burst in demand, and then gracefully scale back down to a steady state after the increased demand has subsided.

- Relational databases and other types of warehouses are exceptionally **unlikely** candidates
- Putting everything into containers because all the cool kids are doing it is a terrible plan.
- Moving monolithic applications into containers and saying that you've begun a transformation into microservices and containers is as truthful as stating that you own the Brooklyn bridge.
- "Lift and shift" is and will always be a recipe for unnecessary grief and instability



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!

https://corporate-timecard-hrapps.apps.ocpzcl125.ciocloud.example.com/timecardapp/login.jsp

OCP for Web applications belongs behind a proxy

Multitenancy

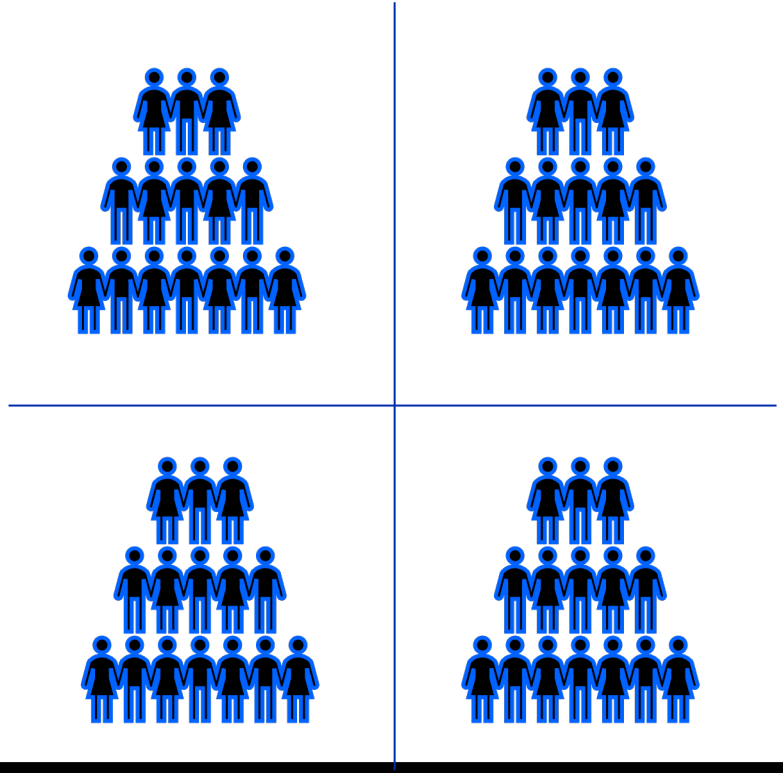
The main prerequisite: Thoughtful planning

- 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.



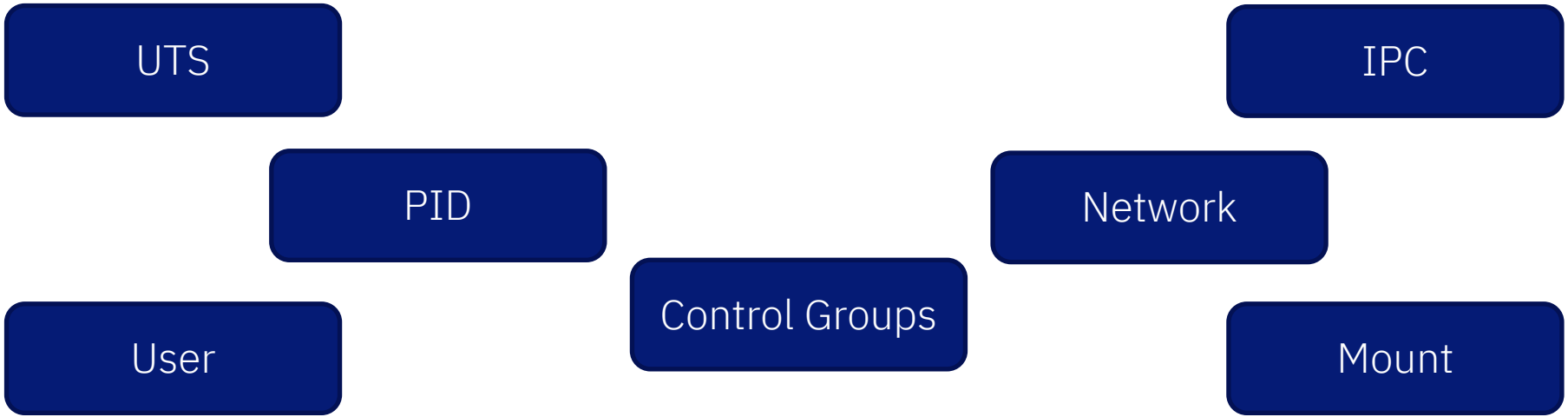
Red Hat OpenShift supports multitenancy through a combination of:

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

Linux kernel Namespaces must not be conflated with Kubernetes Namespaces.

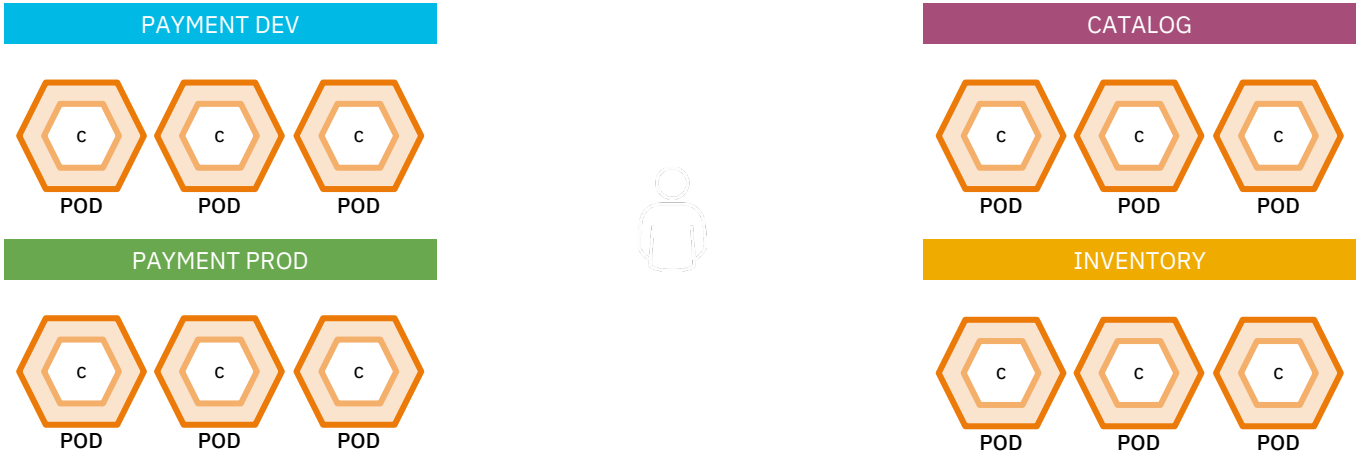
They are entirely different.

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.

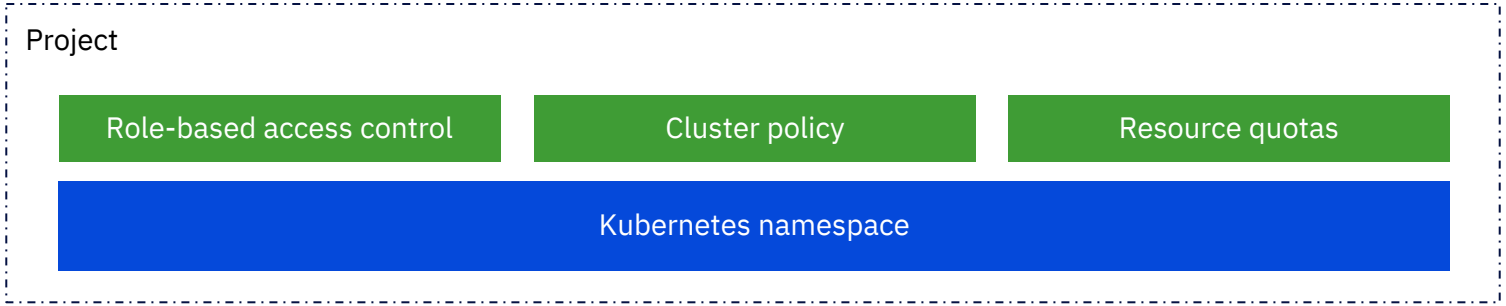
Do not rely on namespaces as a security feature outside of cluster internals within trusted domains.

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

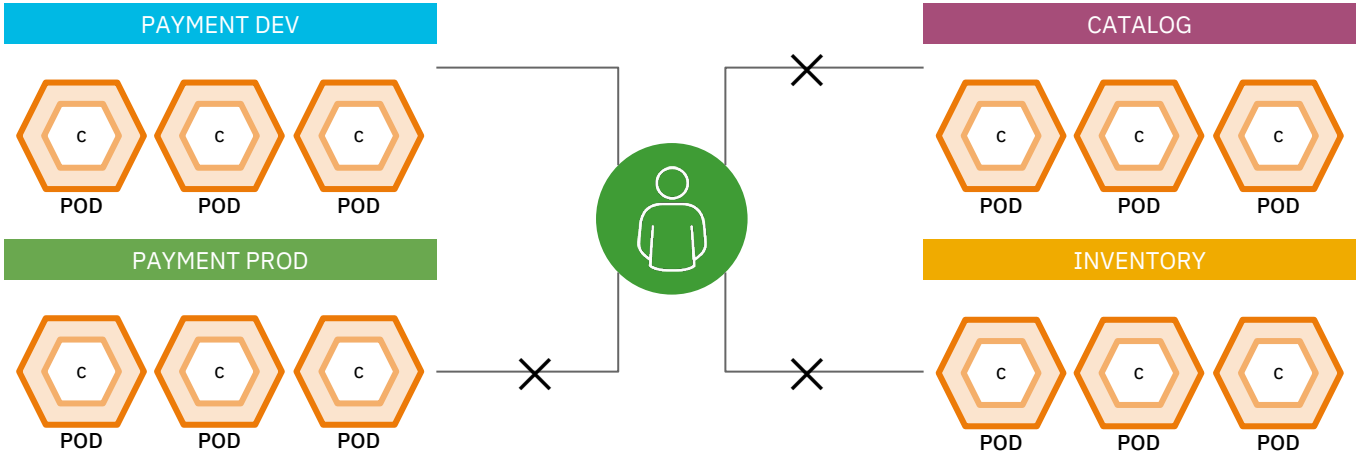
- NIST special publication 800-190: Use a container-optimized OS for additional security
- RHEL CoreOS
  - OS base for Red Hat OpenShift
  - Reduces the attack surface by minimizing the host environment and tuning it for containers
  - Only contains the packages necessary to run Red Hat OpenShift
  - 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.

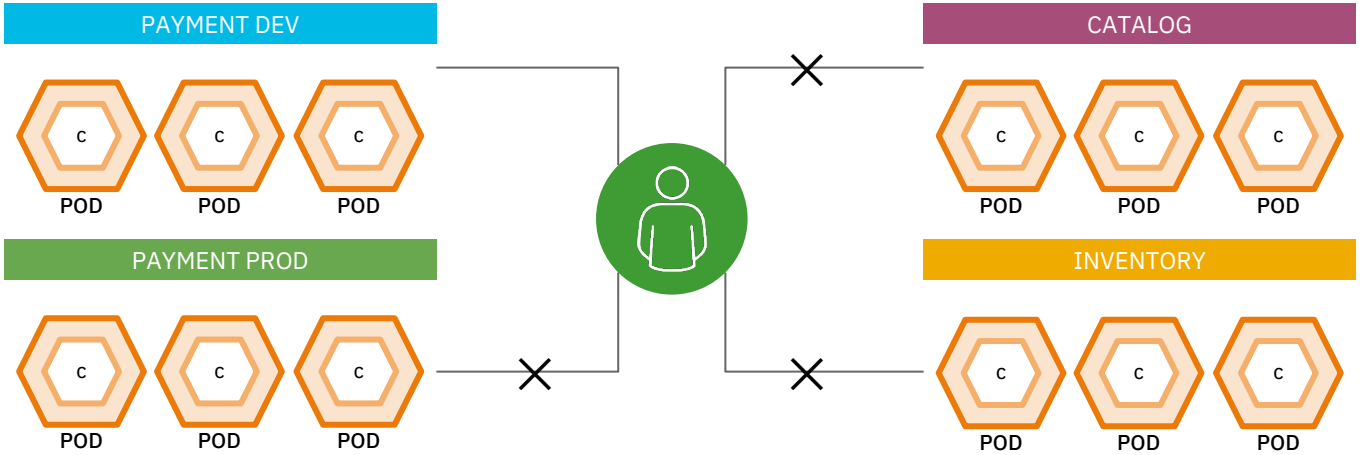
A namespace plus the RBAC layer and some other enhancements is a project



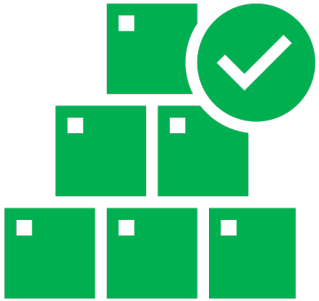
Projects isolate apps across environments, teams, groups and departments in a secure way.

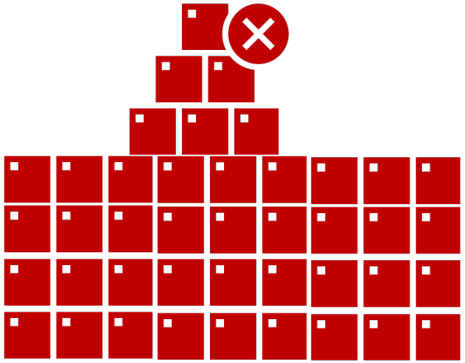


IBM Z and LinuxONE are the only platform where SECURE multi-tenant usage is possible



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





Red Hat OpenShift Container Platform on IBM Z & LinuxONE

All workshop materials can be found here:
https://mmondics.github.io/ocp-cloudpak-workshop/

On break until 11:15

Schedul

Start 10:00 AM Eastern

Presentation
- Overview of OpenShift

Short Break

Short Break

Presentation
- Mini Poll
- Installation
- Requirements
- Options

Short Break

Lab exercises
- Mini Poll
- Self-paced
- Non-sequential

End

4:00 PM Eastern

Workshop Content

Part Three:
Red Hat OpenShift Lifecycle, Installation, Upgrades, and Options

Empowerment Promise

By the end of this segment, you will...

1) Understand the different options for deploying OpenShift on IBM zSystems.

2) Understand the requirements for a successful deployment.

3) Understand the maintenance and upgrade process.

OPENSHIFT CONTAINER PLATFORM

Installer-Provisioned Infrastructure

Simplified opinionated "Best Practices" for cluster provisioning

Fully automated installation and updates including host container OS.

Red Hat Enterprise Linux CoreOS

User-Provisioned 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.

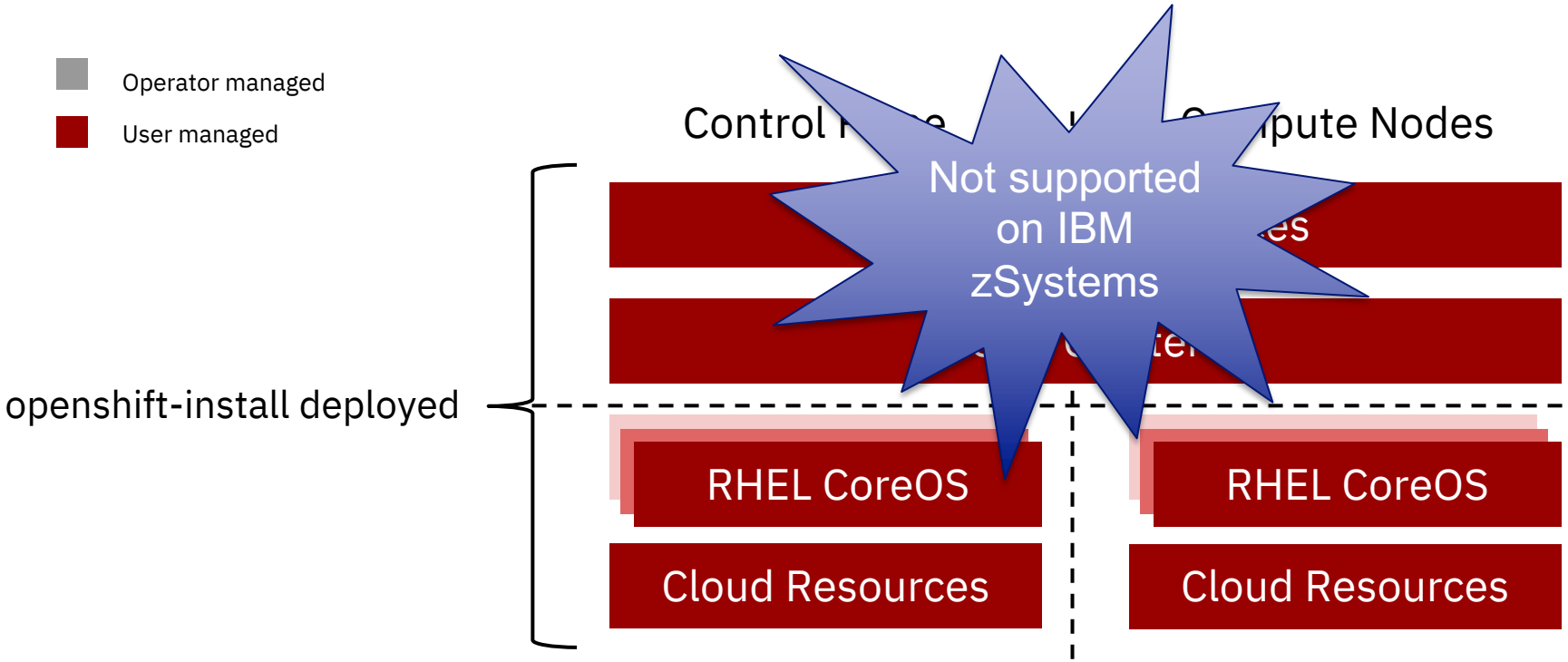
Azure Red Hat OpenShift

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

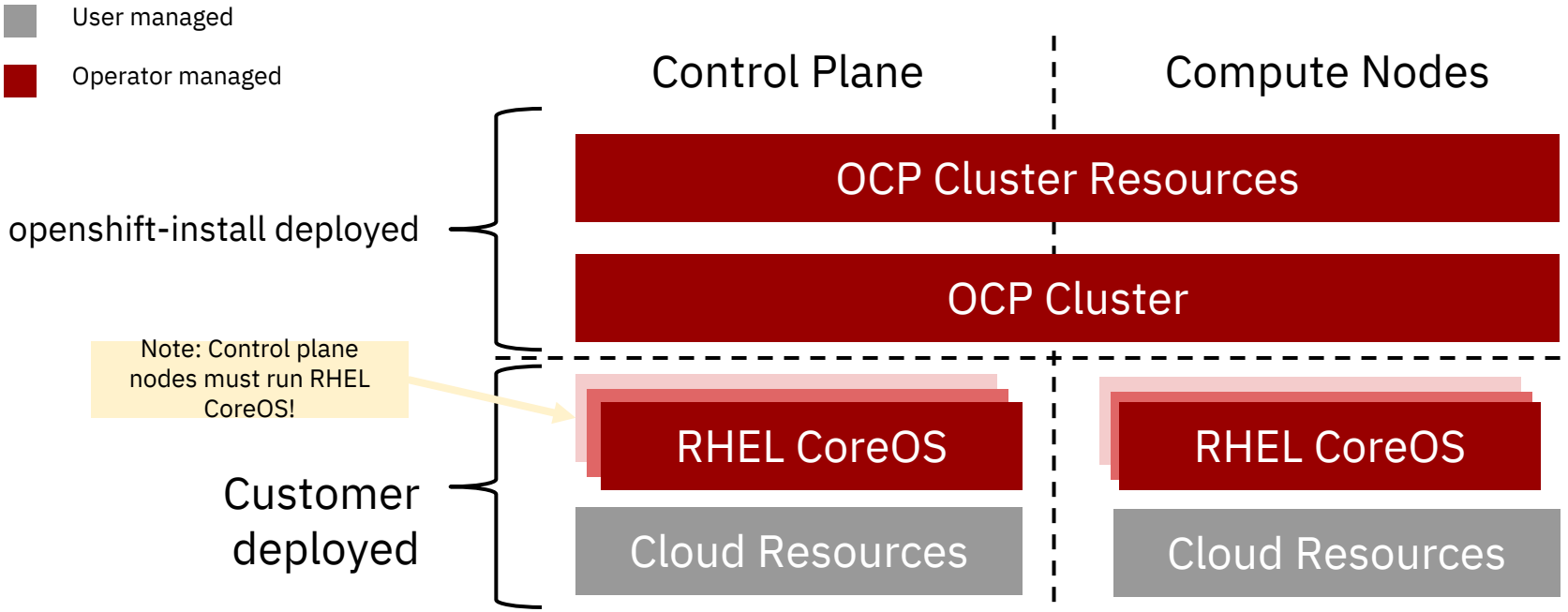
OpenShift Dedicated

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

Installation Paradigms



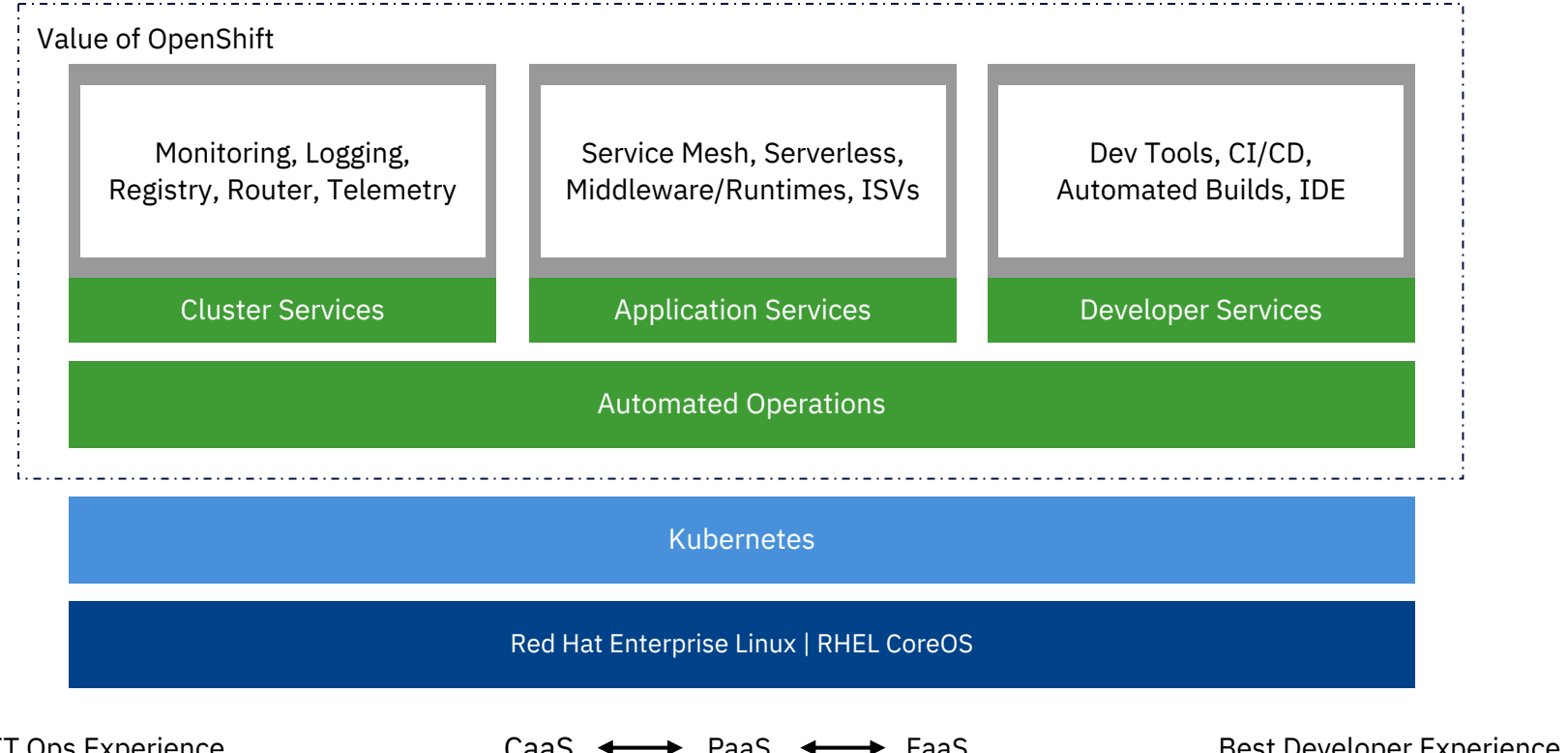
Installer-Provisioned Infrastructure (IPI) Installation



User-Provisioned Infrastructure (UPI) Installation

	Installer-Provisioned Infrastructure	User-Provisioned 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

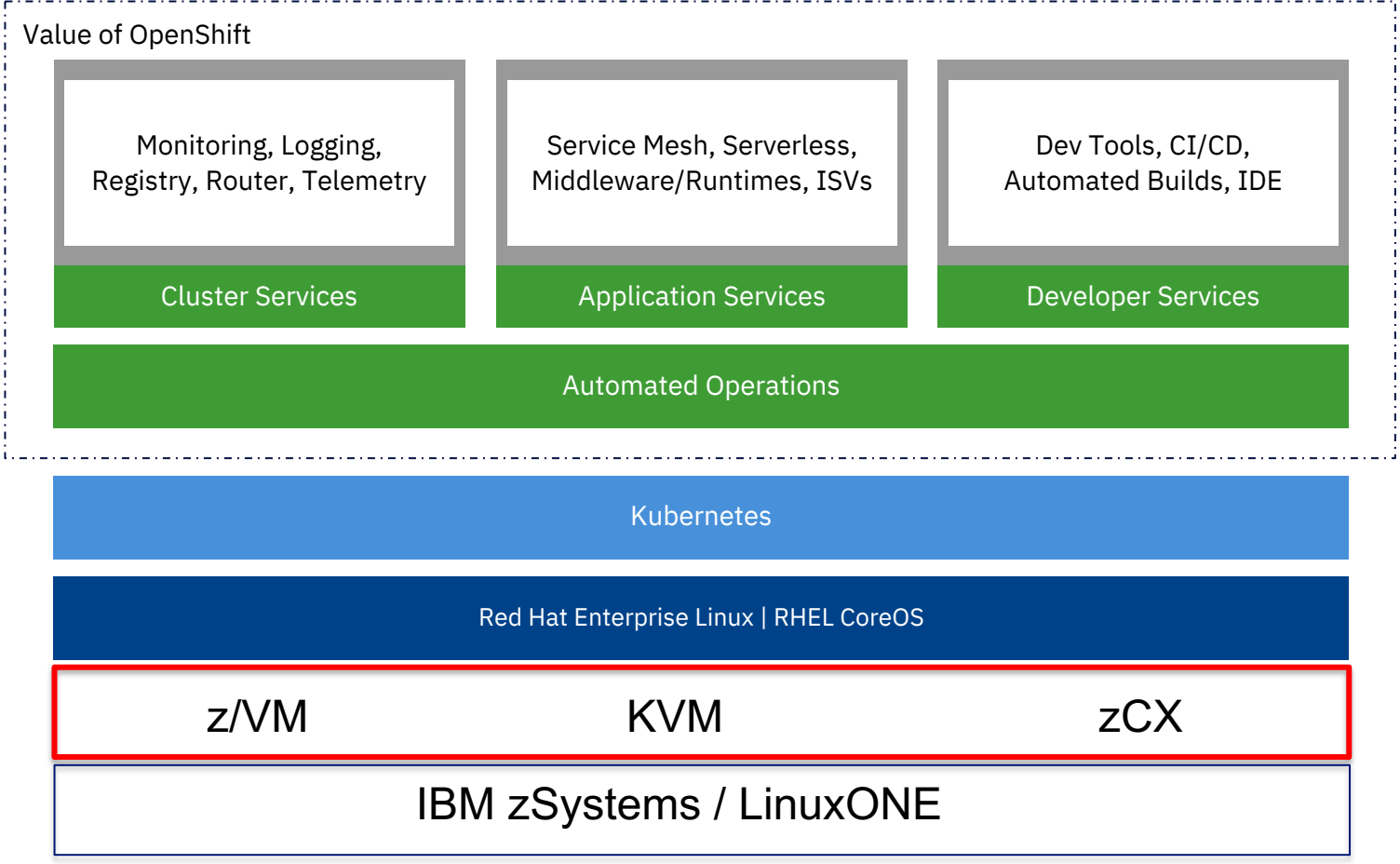


Best IT Ops Experience

CaaS ⟷ PaaS ⟷ FaaS

Best Developer Experience

A best-of-breed concept



Hypervisor

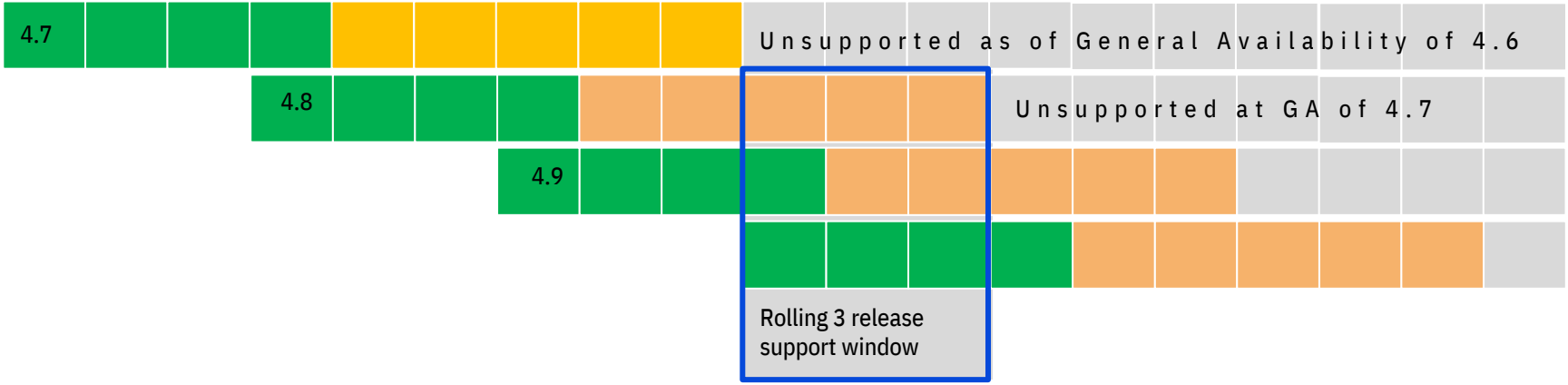
Method	Hypervisor	Paradigm	Automated?	Supported?	Instructions
Manual z/VM	z/VM	UPI	No	Yes	Link
Manual KVM	KVM	UPI	No	Yes	Link
Manual zCX	zCX	UPI	No	Yes	Link
ICIC	Z/VM & KVM	UPI	Yes	Yes	Link
VM ESI	z/VM	UPI	Yes	No	Link
KVM IPI	KVM	IPI	Yes	No	Link
OAAKZ	KVM	UPI	Yes	Yes*	Link

*Once cluster is up it is supported, but the installation method itself is not.

Deployment Options

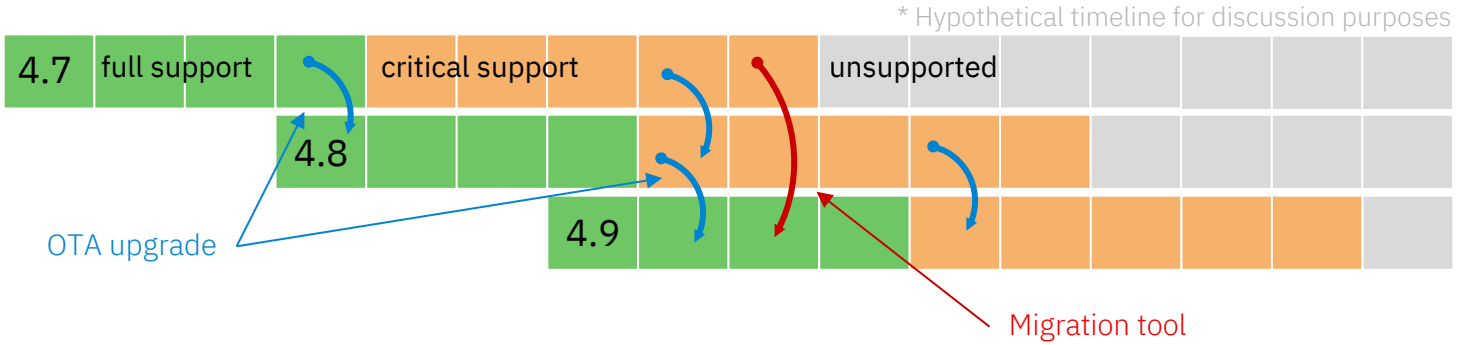
Supported paths for upgrades and migrations

* Hypothetical timeline for discussion purposes



New release support model
EUS release planned

Support Timelines



OTA Upgrades

Happy path = migrate through each version

Optional path = migration tooling

Current minor release

Previous minor release

Upgrades vs. Migrations

Empowerment Promise

By the end of this segment, you will...

1) Understand the different options for deploying OpenShift on IBM zSystems.

2) Understand the requirements for a successful deployment.

3) Understand the maintenance and upgrade process.

3 Key Takeaways

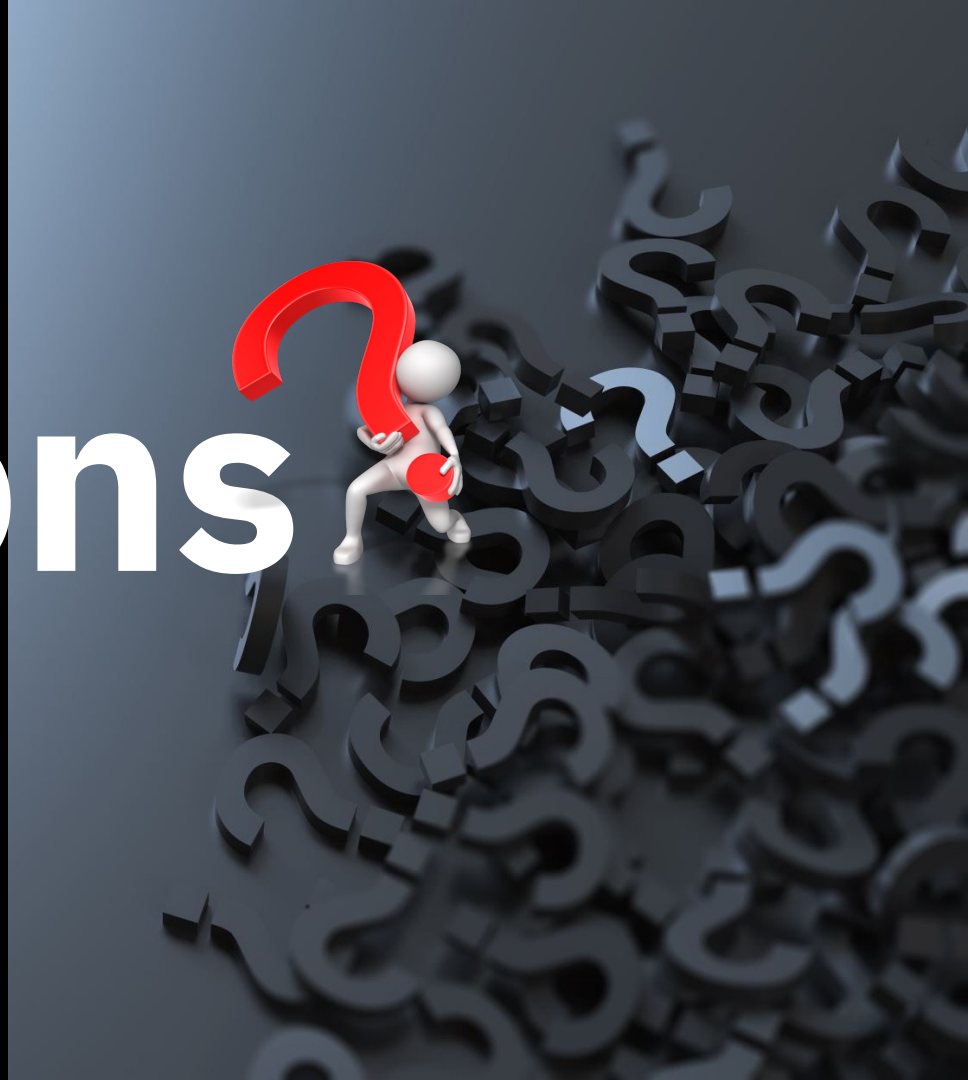
Checking back in on my empowerment promise.

1) Tailor the hypervisor options to suit the client's needs and existing infrastructure.

2) Since UPI must be used, give ample time for deployment. Have a plan in place, all necessary parties on board, and an empowered deployment team.

3) Staying up-to-date with OpenShift releases is critically important for maintaining proper support, and easy to do – so don't fall behind! Make sure your clients have a plan in place for cluster maintenance.

Questions?



Thank you

Jacob Emery
Technical Sales Enablement Specialist - OpenShift on IBM zSystems and Ansible
—
jacob.emery@ibm.com
IBM Washington Systems Center (WSC)

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.

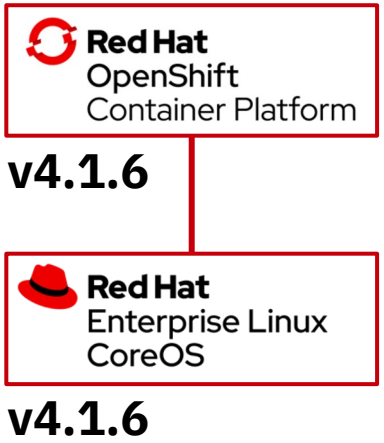
Red Hat Enterprise Linux CoreOS is managed by the cluster

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)

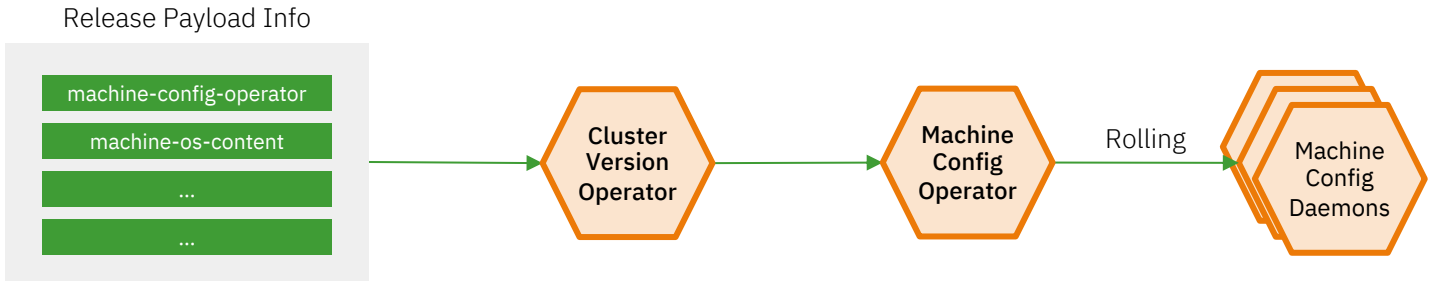
- Terraform provisions initial masters on public cloud / full-stack automated installs. BaNWIS / Bastion on Z and LinuxONE
- Machine API adopts existing masters post-provision
- Each master is a standalone Machine object
- Termination protection (avoid self-destruction)

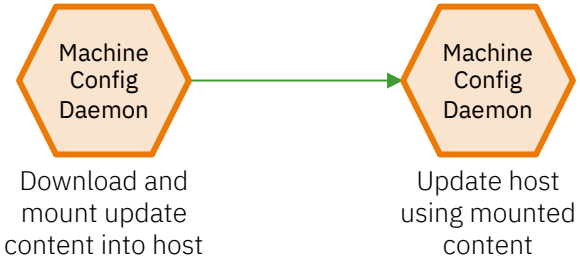
Computes

- 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.

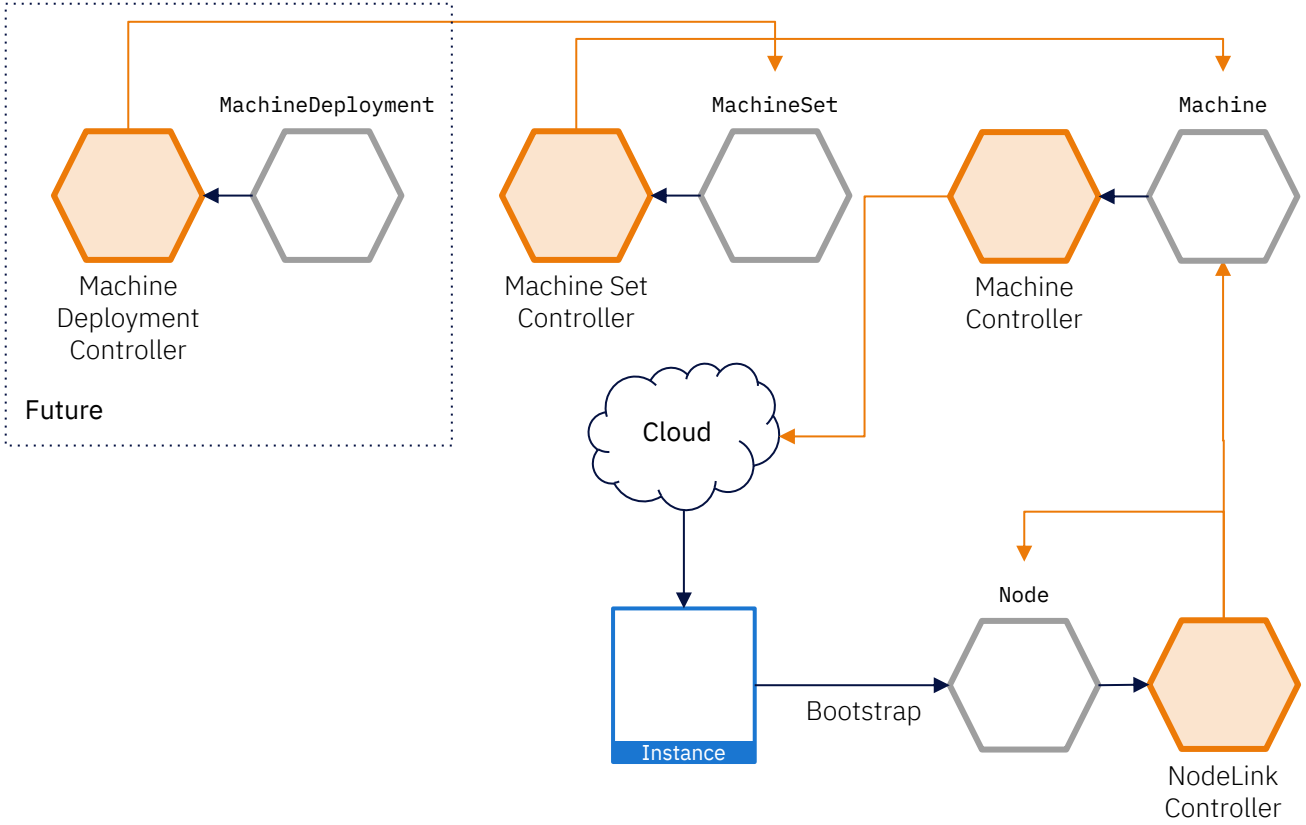
How everything deployed comes under management

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

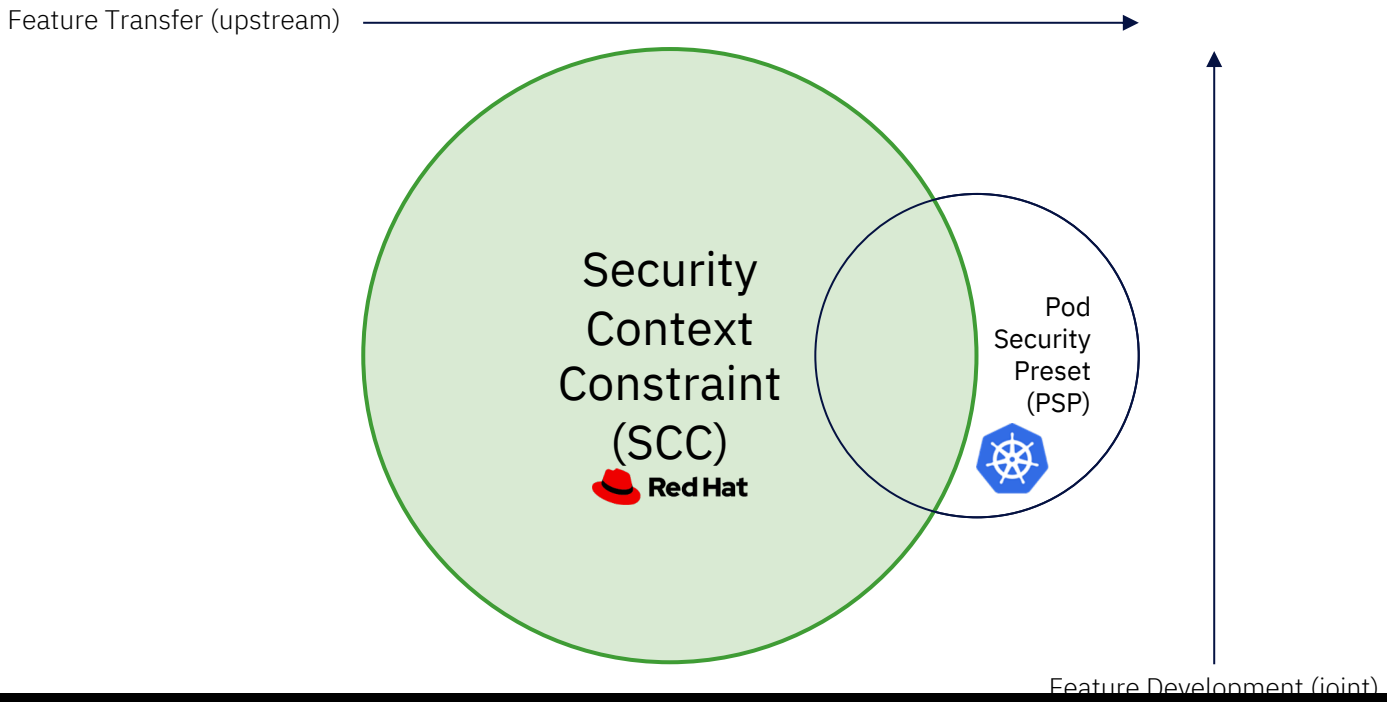
DEFEND
Infrastructure

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

EXTEND

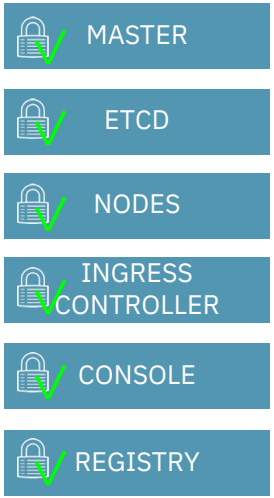
- Security Ecosystem

Security is built in

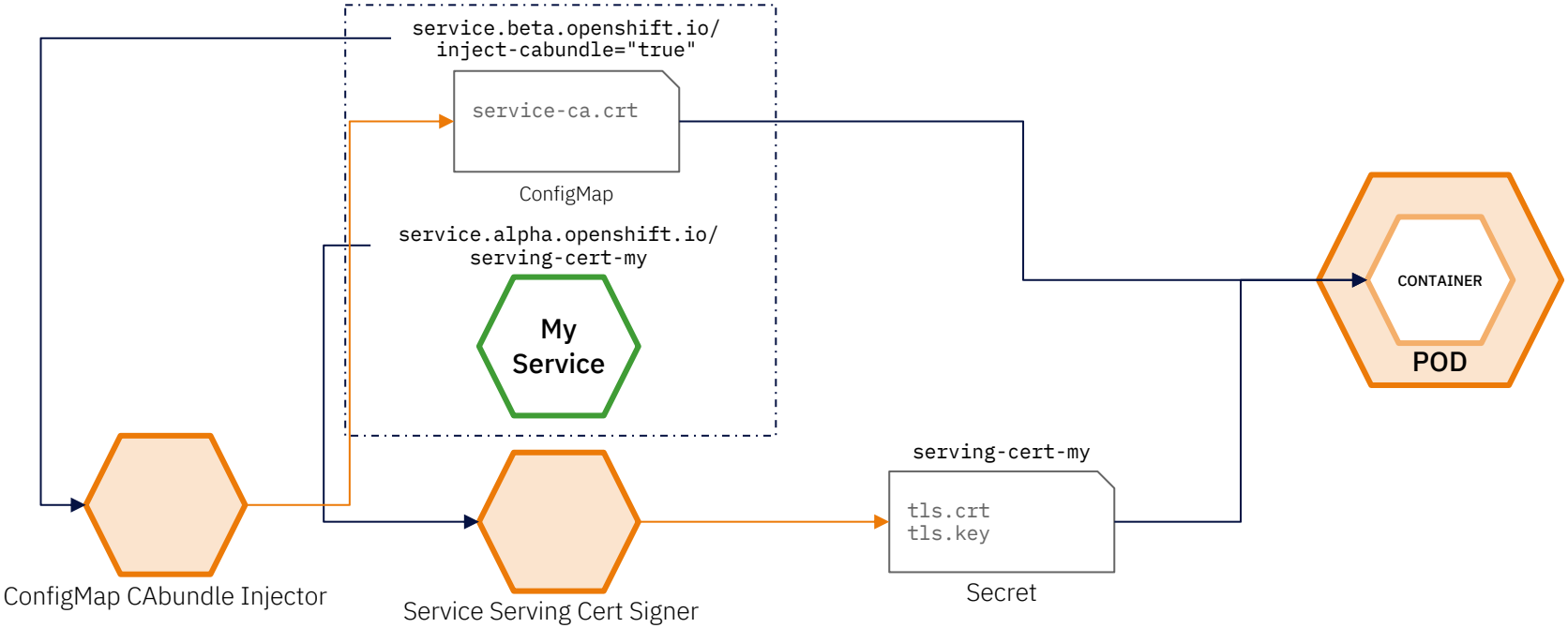


Extended Depth of Protection

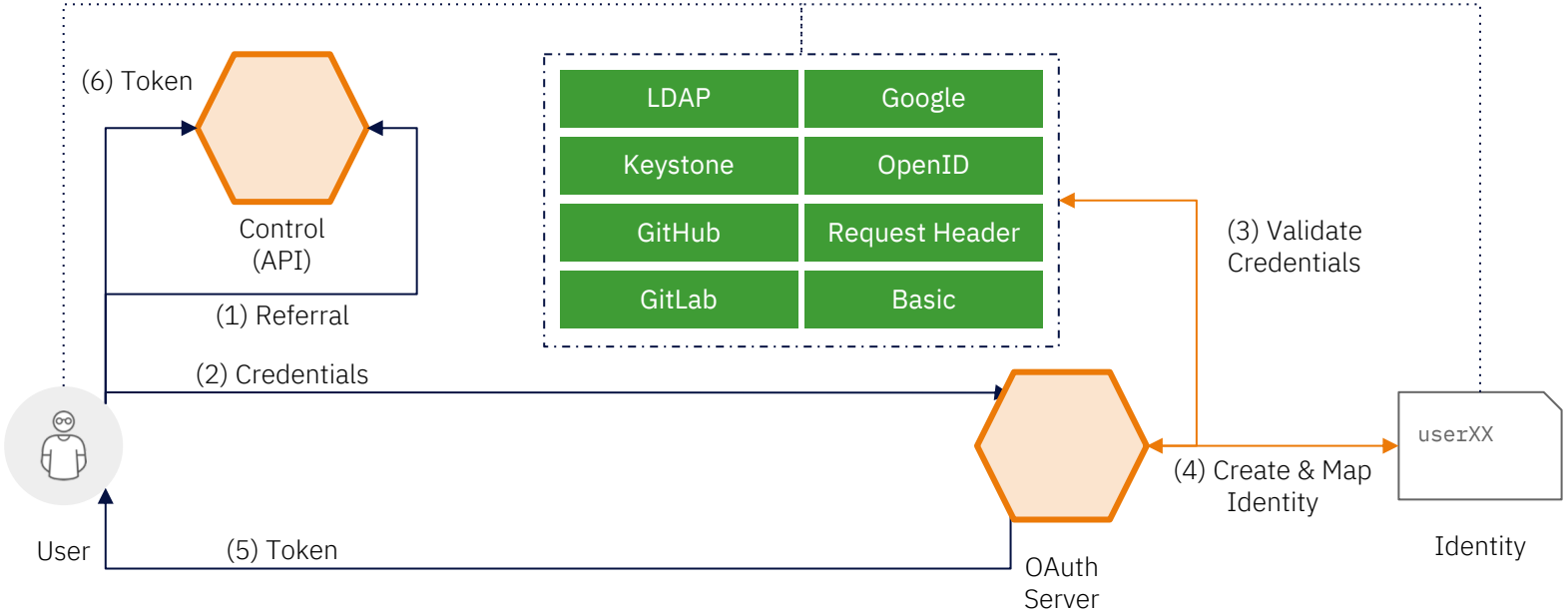
- OpenShift provides its own internal CA
- Certificates are used to provide secure connections to
  - master (APIs) and nodes
  - Ingress controller and registry
  - etcd
- Certificate rotation is automated



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

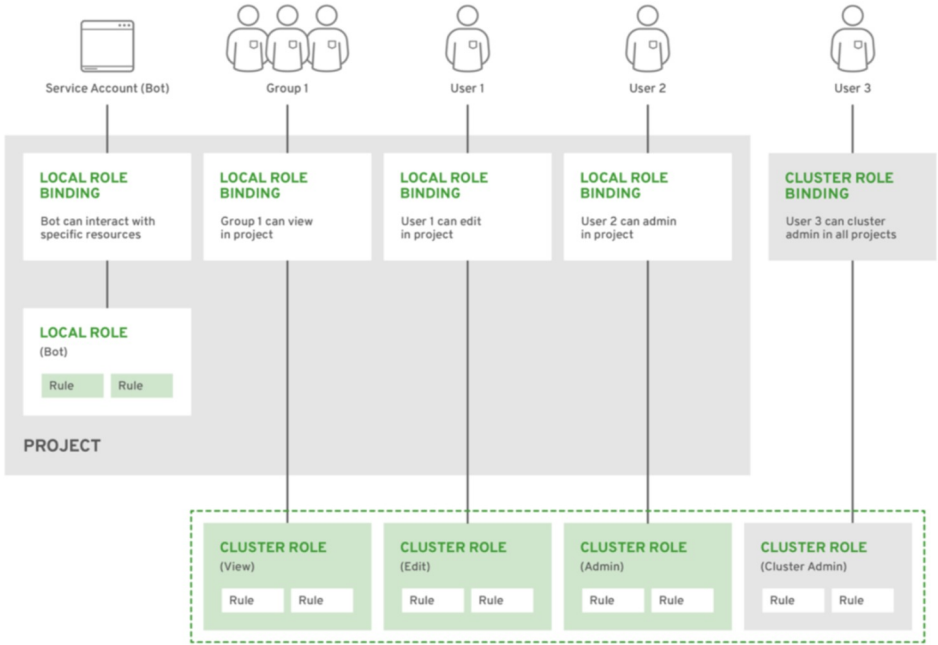


Figure 12 - Authorization Relationships

Fine-Grained RBAC

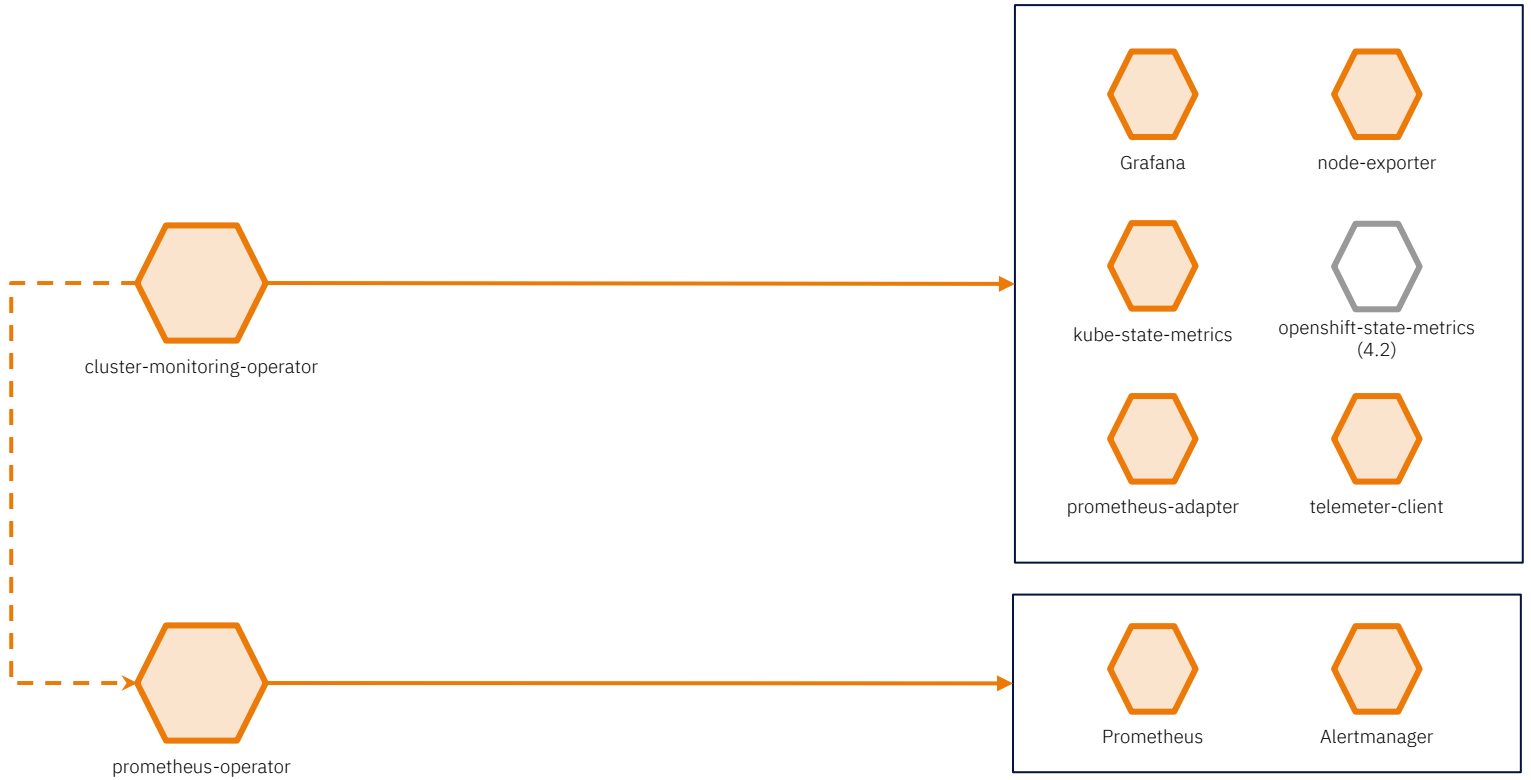
An integrated cluster monitoring and alerting stack

OpenShift Cluster Monitoring

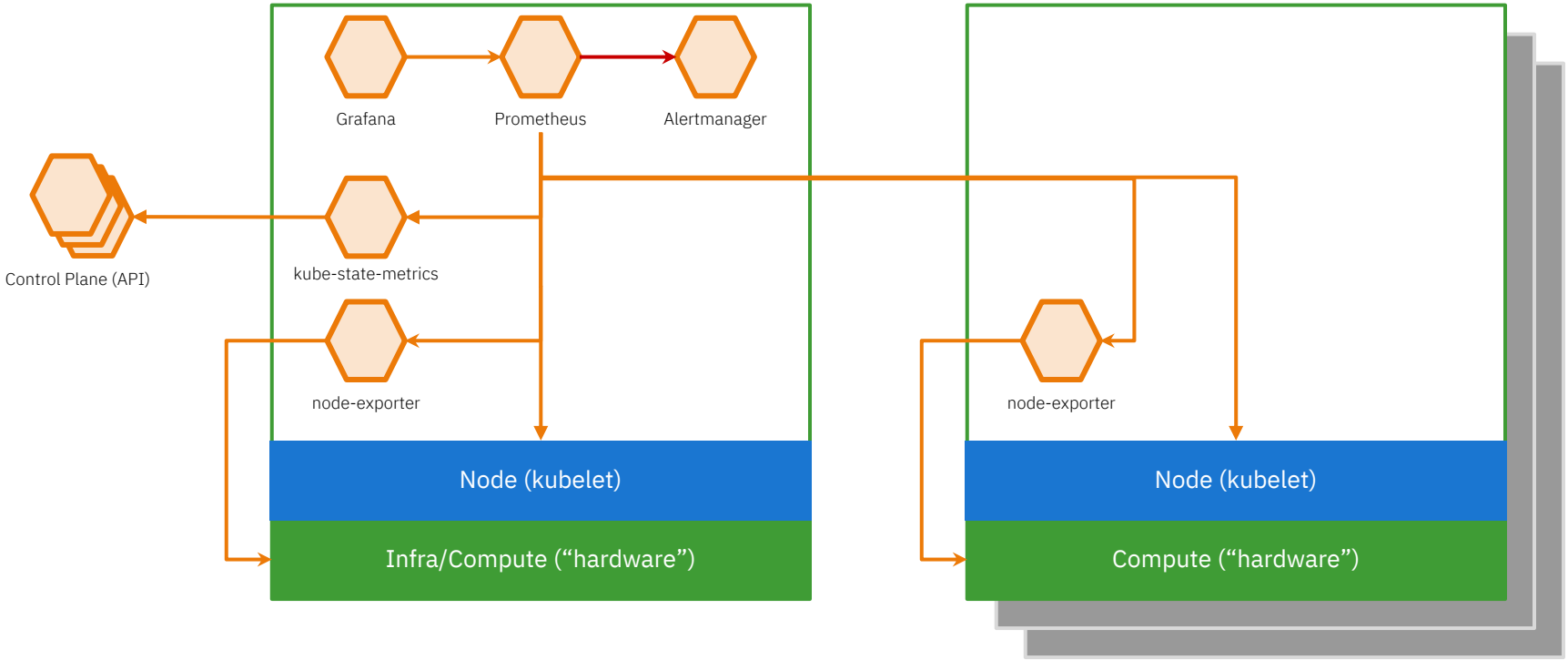
Metrics collection and storage via Prometheus, an open-source monitoring system time series database.

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

Metrics visualization via Grafana, the leading metrics visualization technology.



Monitoring relationships



The “plumbing”

An integrated solution for exploring and corroborating application logs

Components

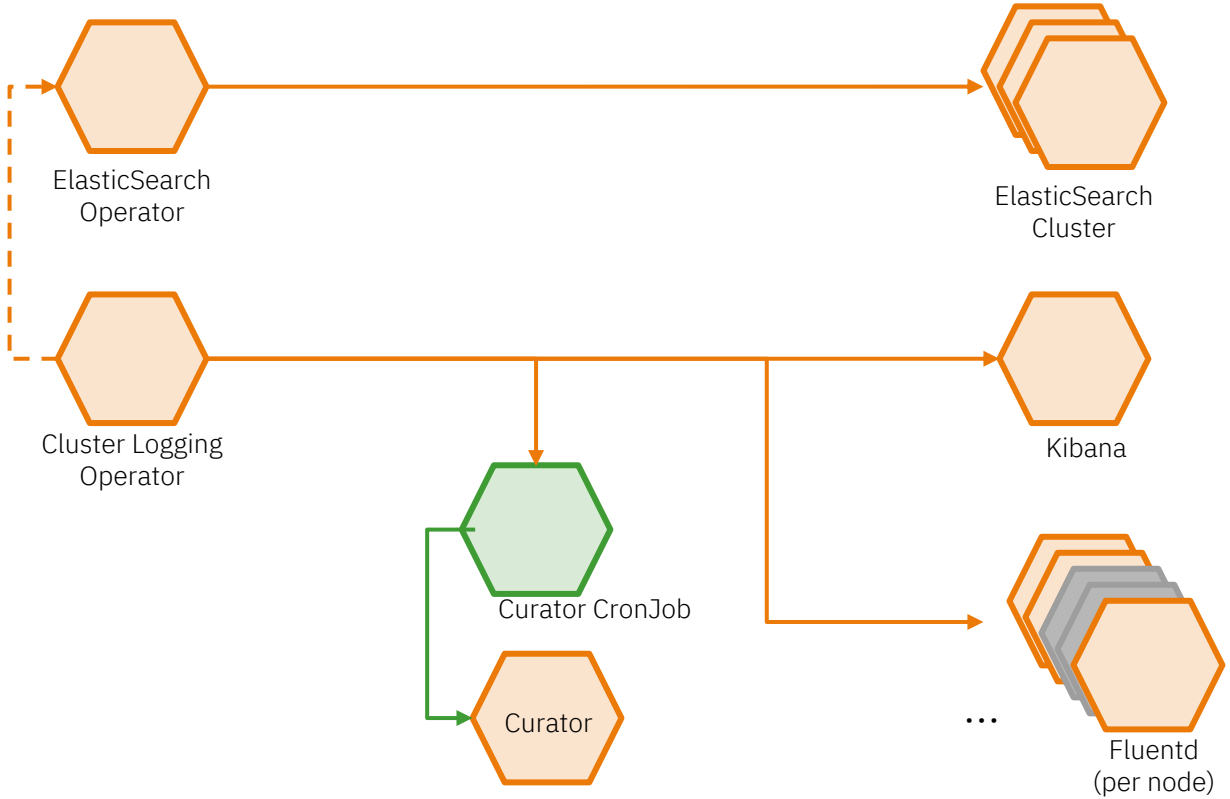
- **Elasticsearch:** a search and analytics engine to store logs
- **Fluentd:** gathers logs and sends to Elasticsearch.
- **Kibana:** A web UI for Elasticsearch.

Access control

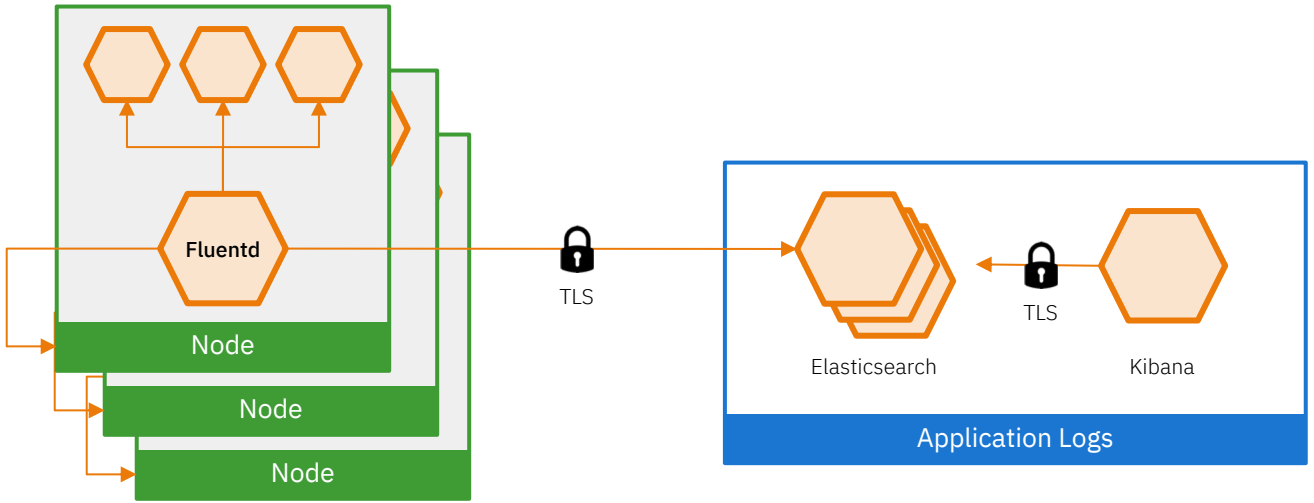
- Cluster administrators can view all logs
- Users can only view logs for their projects

Ability to forward logs elsewhere

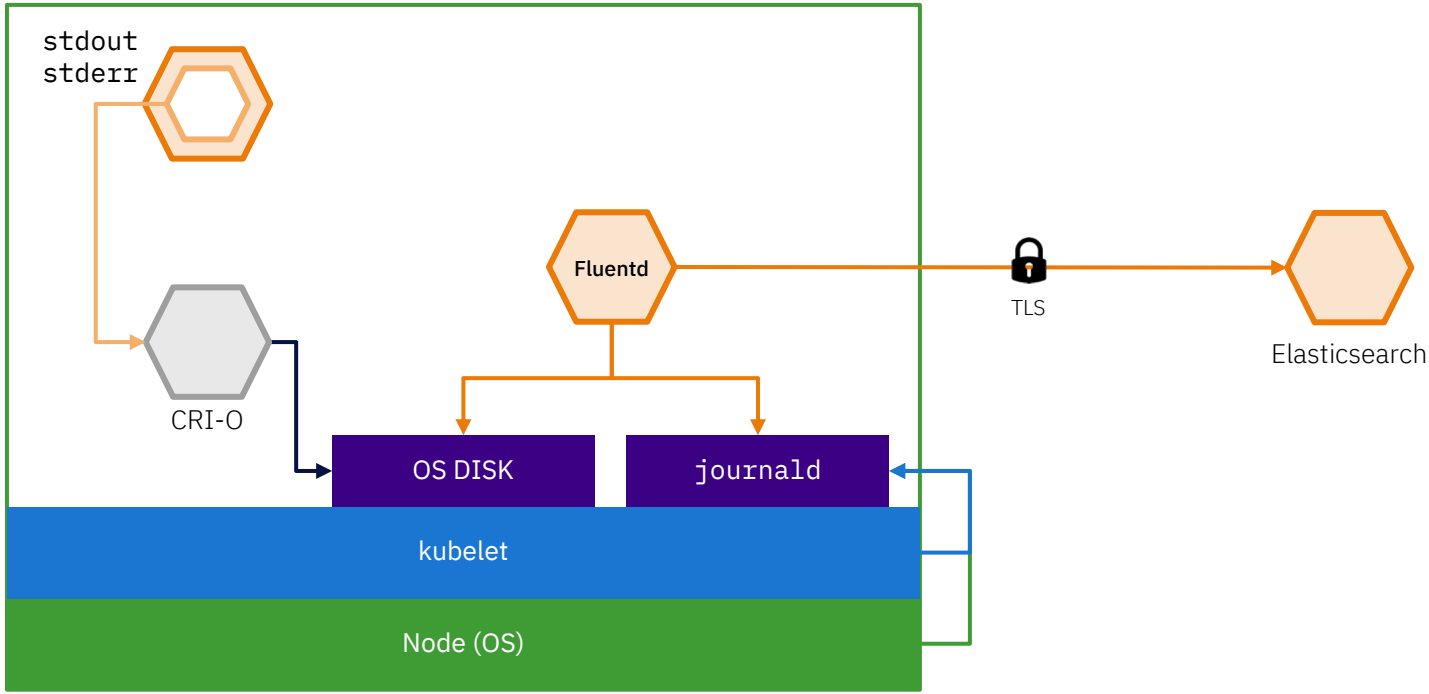
- External elasticsearch, Splunk, etc



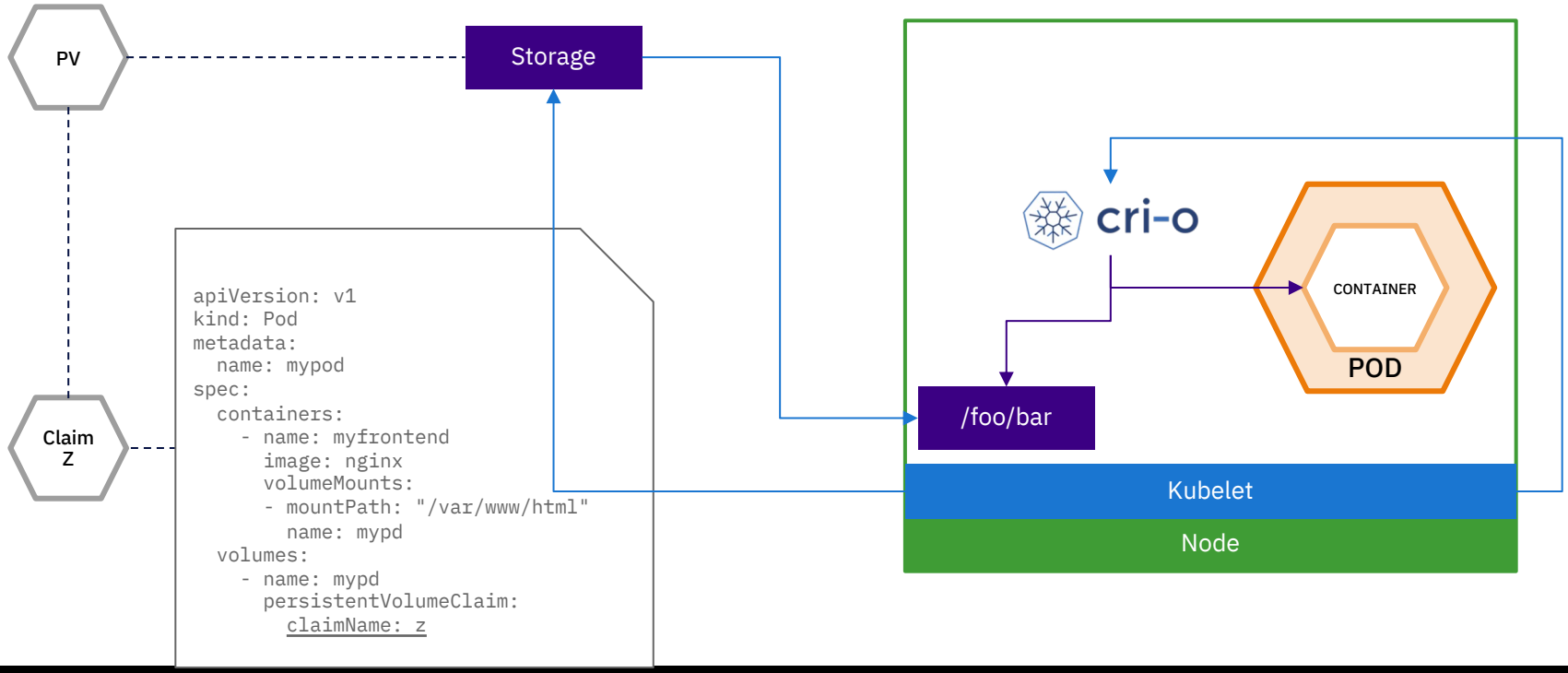
Relationships for logging



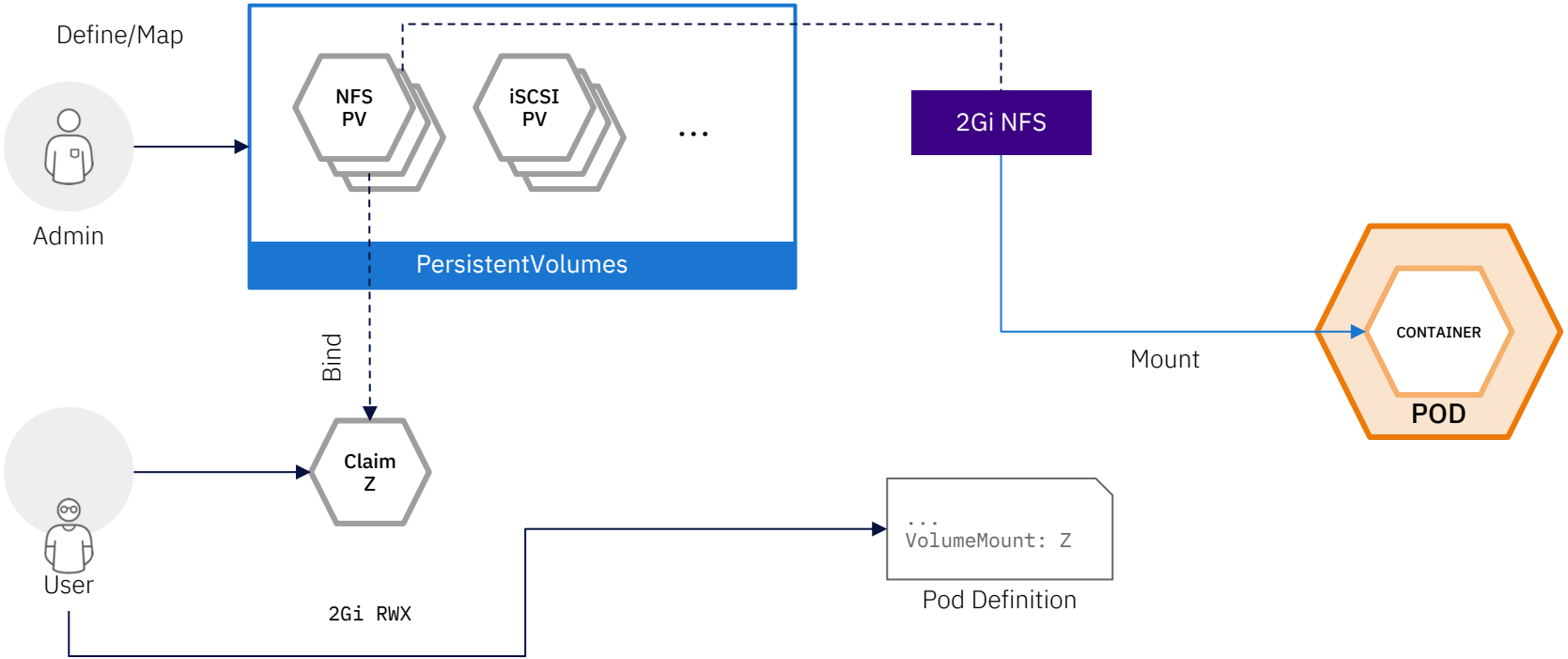
Log data flow in OpenShift



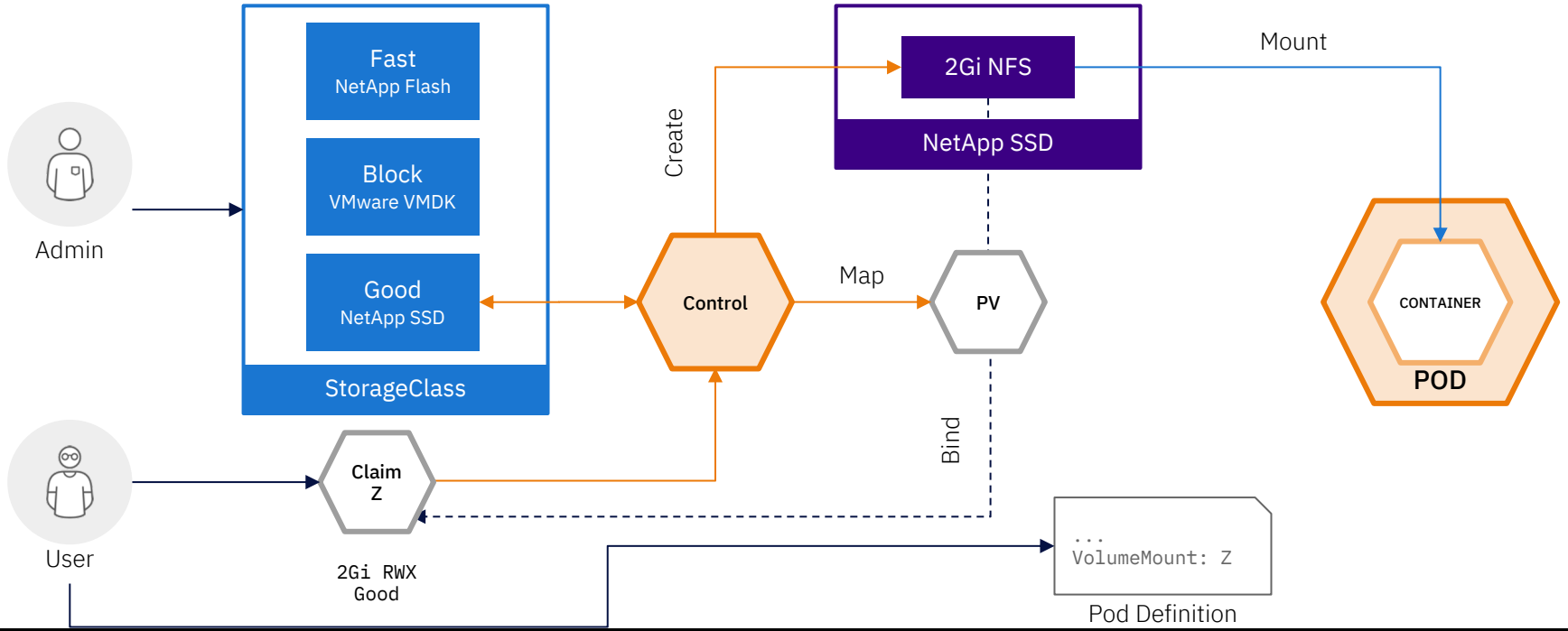
Log data flow in OpenShift



PV Consumption

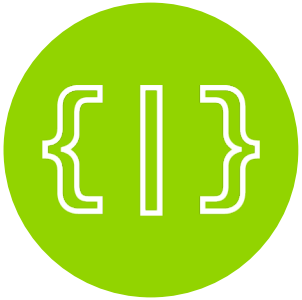


Static Storage Provisioning



Dynamic Storage Provisioning

Tools and automation that makes developers productive quickly

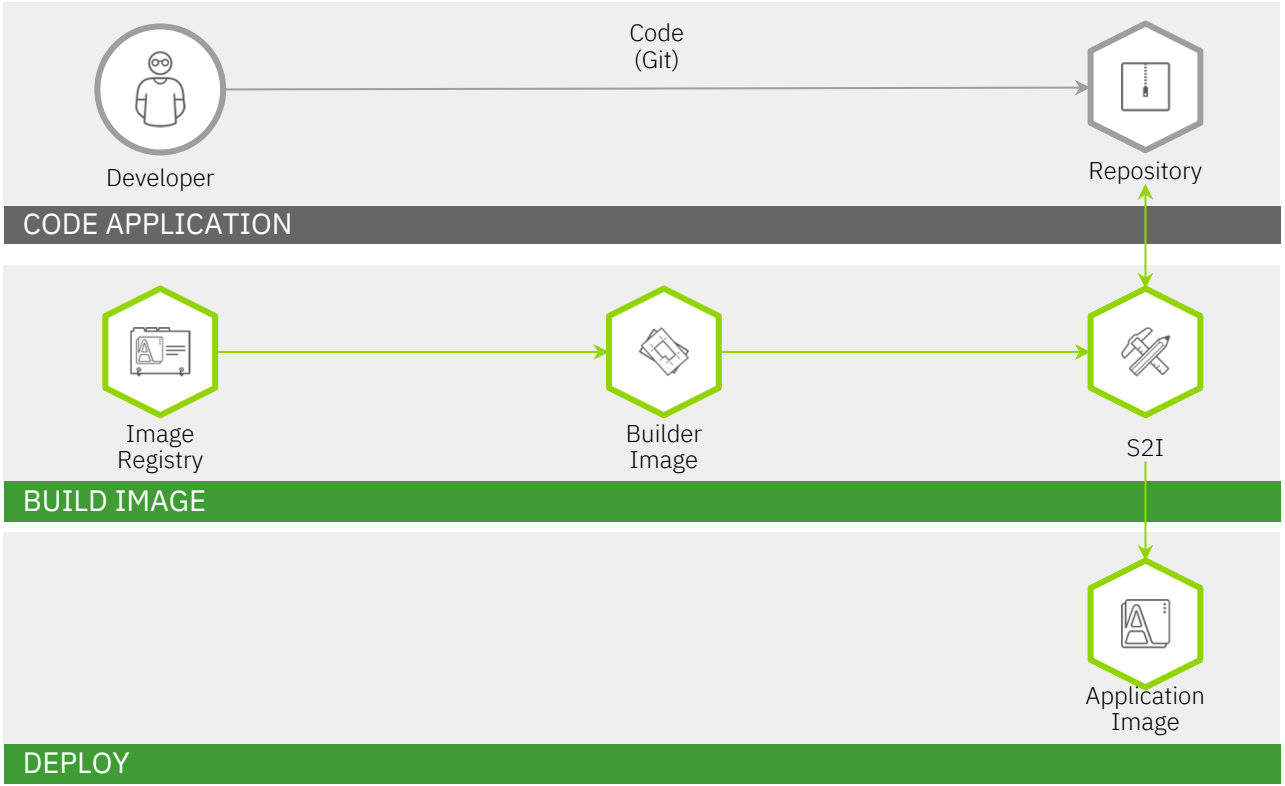


DEPLOY YOUR SOURCE CODE

DEPLOY YOUR APP BINARY

DEPLOY YOUR CONTAINER IMAGE

Methodologies to build and deploy



The Source-to-Image concept

Red Hat OpenShift Container Platform on IBM Z and LinuxONE

All w can be found here:
https:/ io/ocp-cloudpak-workshop/

On break until 12:15

Schedule for the day

Start 10:00 AM Eastern

- Presentation
  - Overview of OpenShift
- Short Break
- Presentation
  - Mini Poll
  - OpenShift on Z technical deep dive
- Short Break
- Presentation
  - Mini Poll
  - Installation
  - Requirements
  - Options
- Short Break
- Lab exercises
  - Mini Poll
  - Self-paced
  - Non-sequential

End 4:00 PM Eastern

Good practices and lessons learned

Red Hat OpenShift Container Platform on IBM Z & LinuxONE

Wildfire Technical Hands-On Workshops
Washington Systems Center

All workshop materials can be found
https://mmondics.github.io/ocp-clou

On break until 1:15

Schedule for the day

Start 10:00 AM Eastern

- **Presentation**
  - Overview of OpenShift
- Short Break
- **Presentation**
  - Mini Poll
  - OpenShift on Z technical deep dive
- Short Break
- **Presentation**
  - Mini Poll
  - Installation
  - Requirements
  - Options
- Short Break
- **Lab exercises**
  - Mini Poll
  - Self-paced
  - Non-sequential

End 4:00 PM Eastern

Time for labs

Red Hat OpenShift Container Platform on IBM Z & LinuxONE

Wildfire Technical Hands-On Workshops
Washington Systems Center

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
- Short Break
- **Presentation**
  - Mini Poll
  - OpenShift on Z technical deep dive
- Short Break
- **Presentation**
  - Mini Poll
  - Installation
  - Requirements
  - Options
- Short Break
- **Lab exercises**
  - Mini Poll
  - Self-paced
  - Non-sequential

End 4:00 PM Eastern

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

IBM

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

IBM

Jacob Emery

IBM

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

IBM WSC washington systems center

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