

# ORCHESTRATING IBM BLOCKCHAIN PLATFORM FOR AGILITY WITH RED HAT OPENSHIFT CONTAINER PLATFORM

GARRETT WOODWORTH

[GARRETT.LEE.WOODWORTH@IBM.COM](mailto:GARRETT.LEE.WOODWORTH@IBM.COM)

TECHNICAL SPECIALIST – BLOCKCHAIN & KUBERNETES

IBM Z

# SESSION ROADMAP

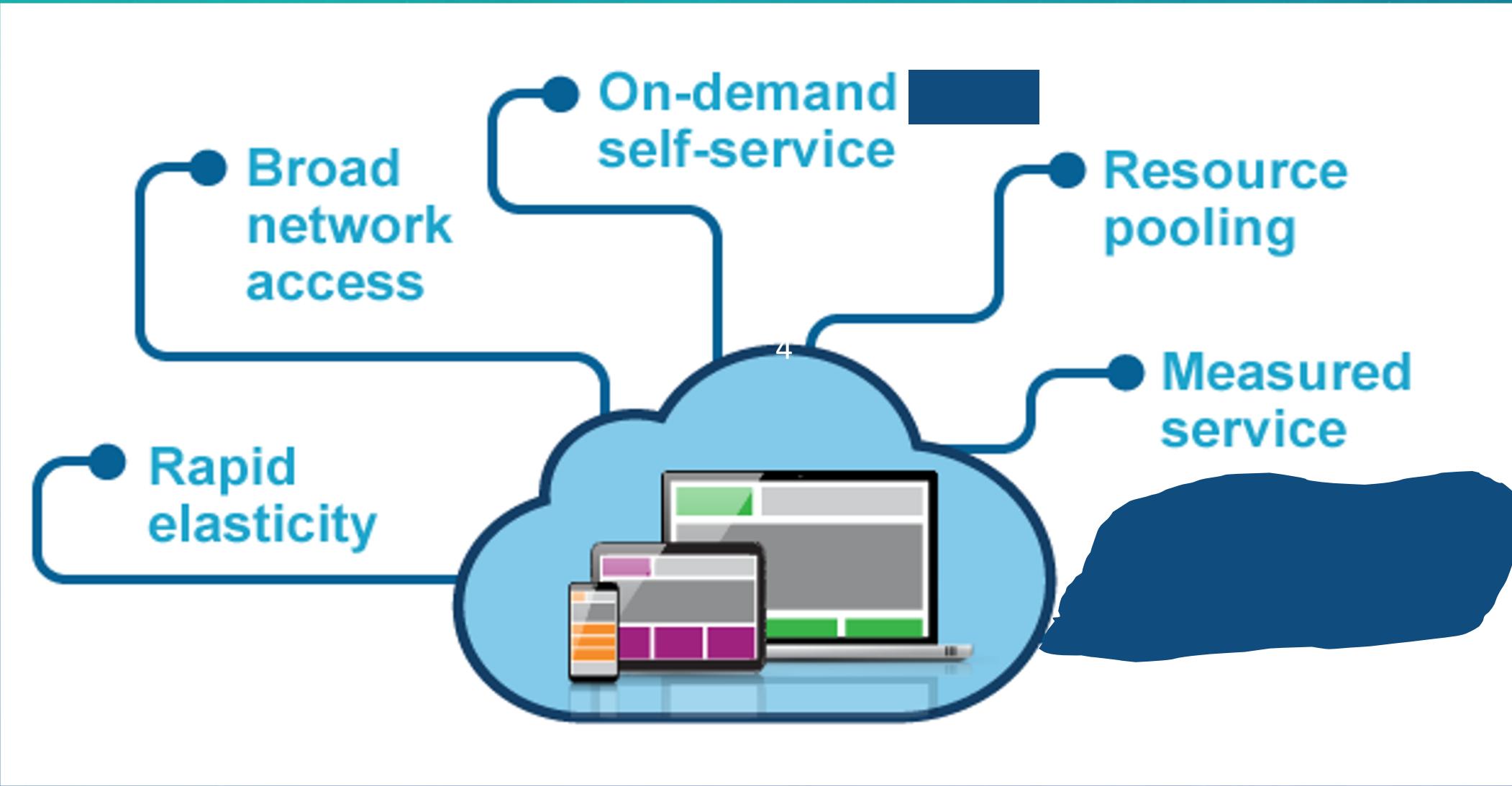
**Evolution of  
cloud and  
containerized  
workloads  
(microservices)**

**What does  
this mean  
for the IBM  
Blockchain  
Platform?**

**Red Hat  
OpenShift  
Container  
Platform  
(OCP)**

# 1. EVOLUTION OF CLOUD AND CONTAINERIZED WORKLOADS (MICROSERVICES)

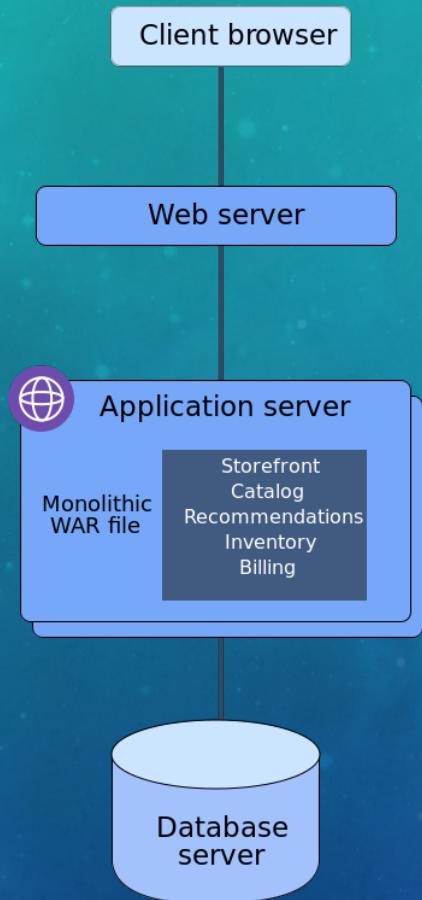
According to NIST cloud is



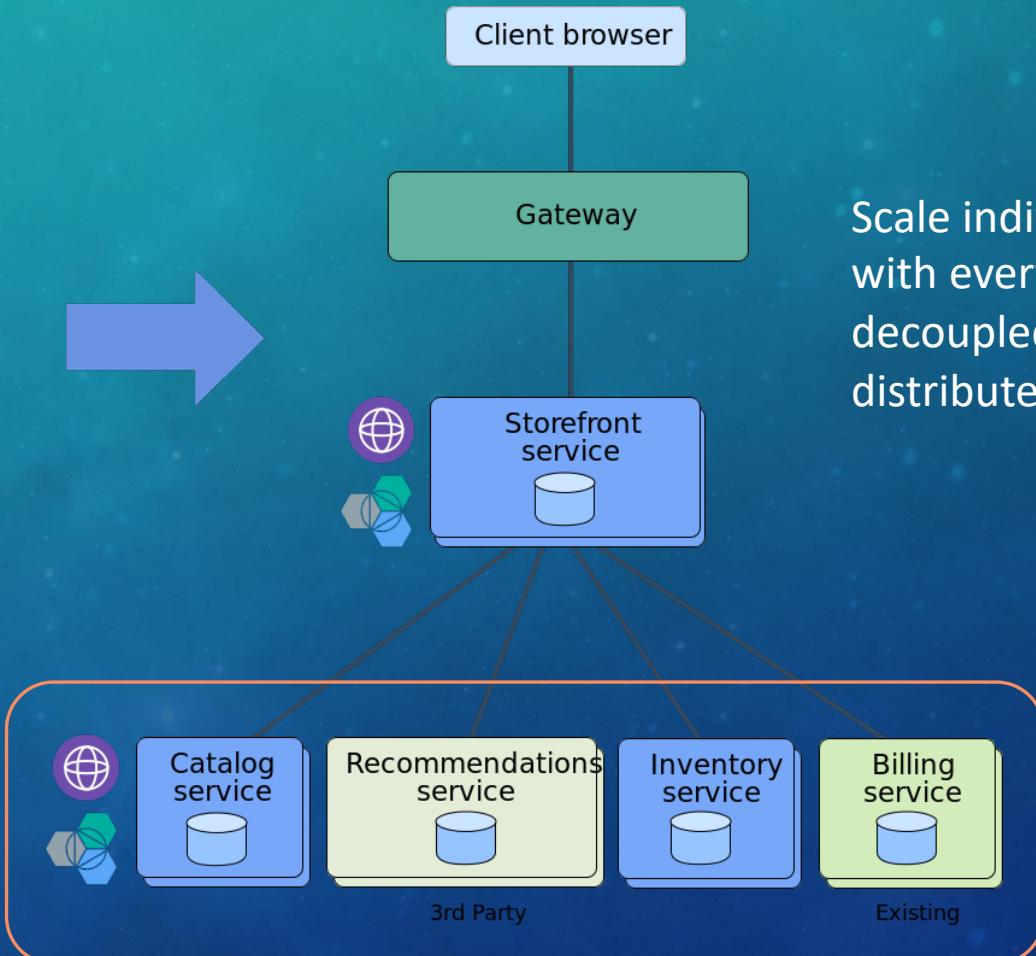
# Need Apps to match scalability and speed of cloud ...

Scale by increasing overall capacity (server and storage) with everything tightly coupled together

Monolithic



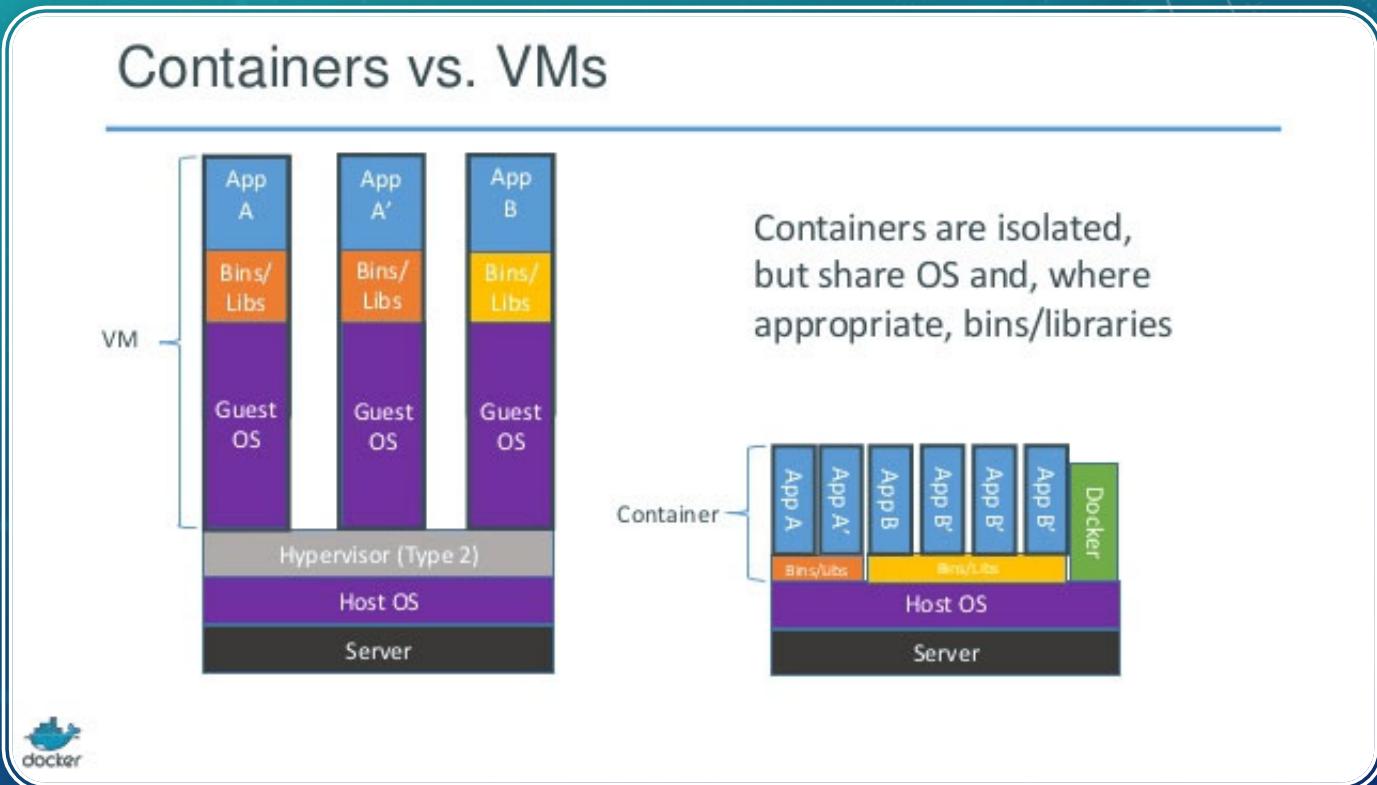
Cloud-Native (Microservices)



Scale individual services with everything decoupled and distributed across nodes

# AND CLOUD NATIVE-ARCHITECTURE (CONTAINERS)

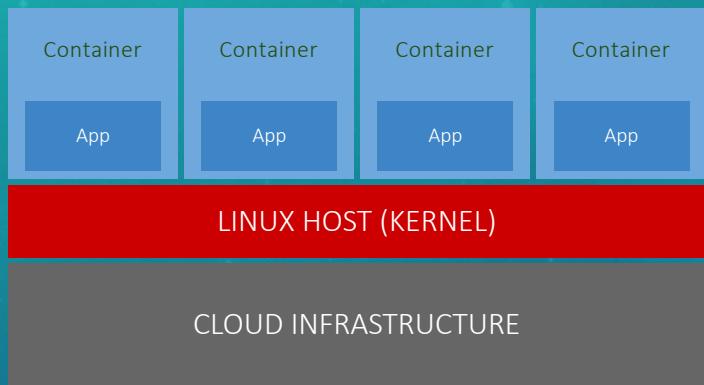
- **Portable**, independently packaged images for apps/services which can be used across Linux distros
- **Lightweight** Namespace Isolation and cgroup resource limits for rapid deployment
- **Storage pooling** of host os and applicable bins/libraries
- Manifest lists -> support for multiple architectures (i.e. s390x, power, x86) [up to developers to enable for a specific container]



# What are Containers?

Consistent Tools for Both Developers and IT Operations

## CONTAINERS



- Integrated in Linux OS
- Fully Open Source
- Secure Isolation of Applications
- Eliminates need for VM Hypervisor
- Runs on Any Cloud Platform

## DEVELOPERS

- Cloud-Native Applications
- Simplify Packaging
- Simplify Testing

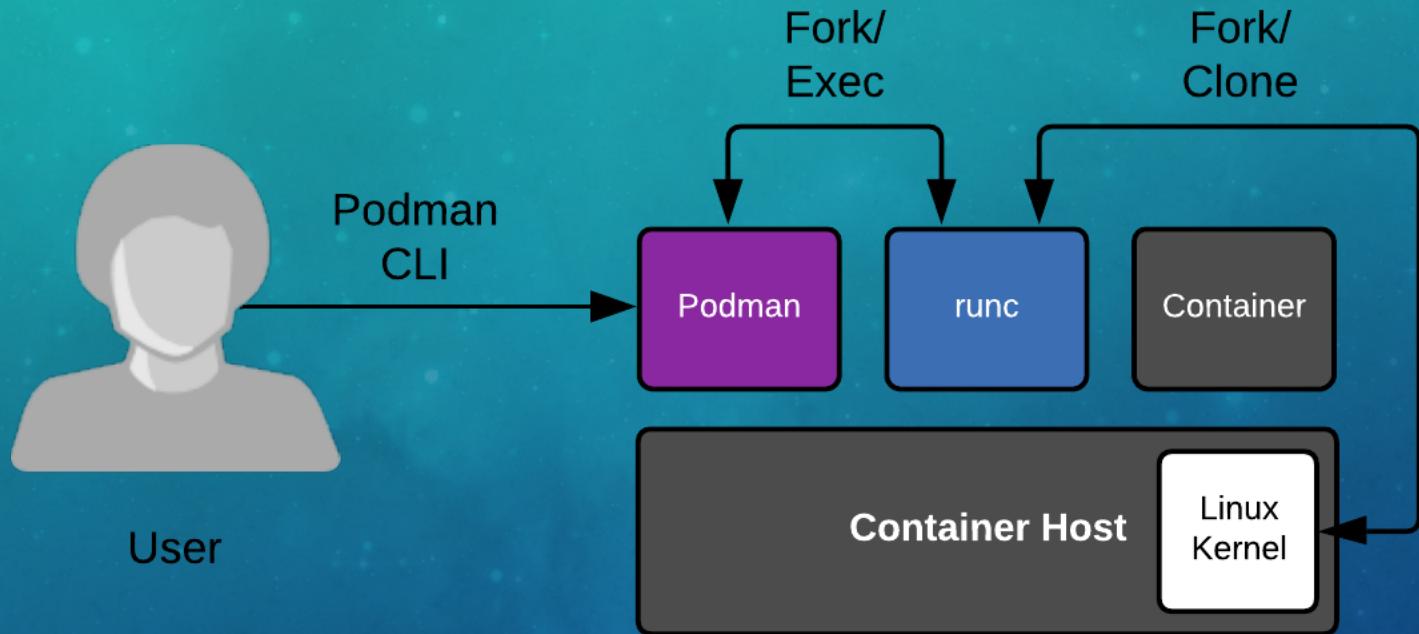
## IT OPERATIONS

- Consistent Application Deploys
- Automated Application Deploys
- Improved Application Performance
- Multi-Cloud Consistency

## BUSINESS LEADERS

- Enable DevOps Culture
- Enable Hybrid Multi-Cloud
- Reduce VM Licensing Costs
- Accelerate App-Dev Cycles

# HOW DOES A CONTAINER RUN (USING PODMAN)



How containers run with a container engine

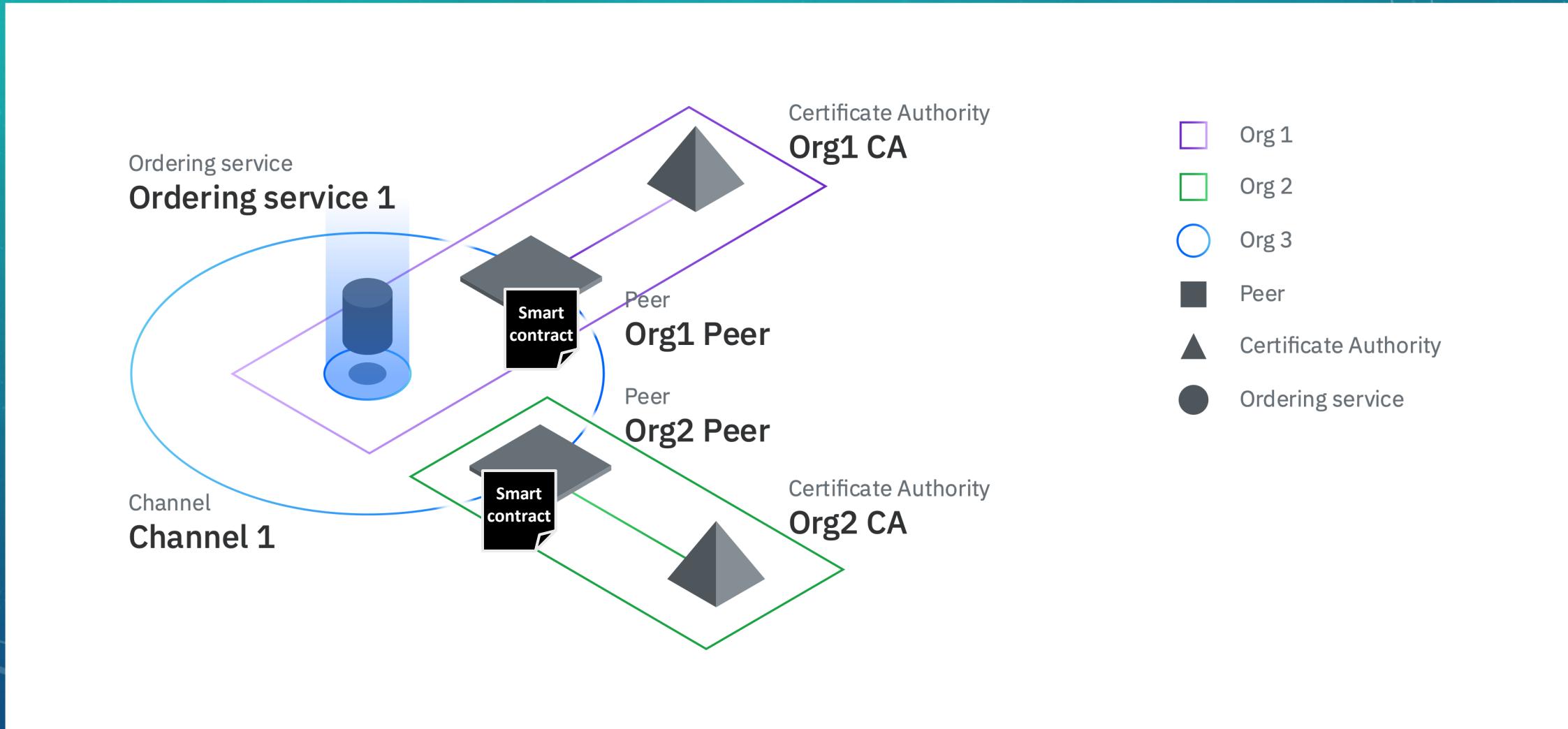
See [RedHat blogpost](#) and [Podman website](#)

# USING PODMAN AND BUILDAH – CAN I STILL RUN MY DOCKER IMAGES?

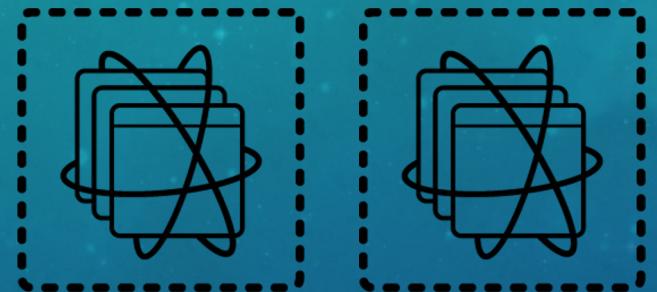
- YES – Podman supports OCI ([Open Container Initiative](#)) [image specification](#) and defaults to runc for running containers (like Docker)
- Podman employs same commands as docker (with Podman instead of Docker) besides some additional flags for convenience such as `--all (-a)` for *podman rm* and *podman rmi* and pod commands
- In fact you can use (alias `docker=podman`) and run existing docker scripts once stopping docker process
- Buildah (think Boston accent) is the builder code for building containers with additional functionality such as using bash scripts instead of Dockerfiles
- Podman uses subset of Buildah code for building images with Podman commands
- See [Podman and Buildah for Docker Users](#) for more information

# EXAMPLE IBM BLOCKCHAIN PLATFORM NETWORK

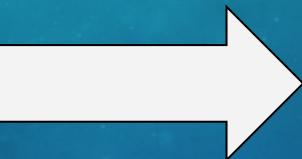
*IT'S ALL RUNNING ON CONTAINERS!*



# Why do Containers need Kubernetes?



CONTAINERIZED APPLICATIONS



## kubernetes

Manage Containers  
Securely

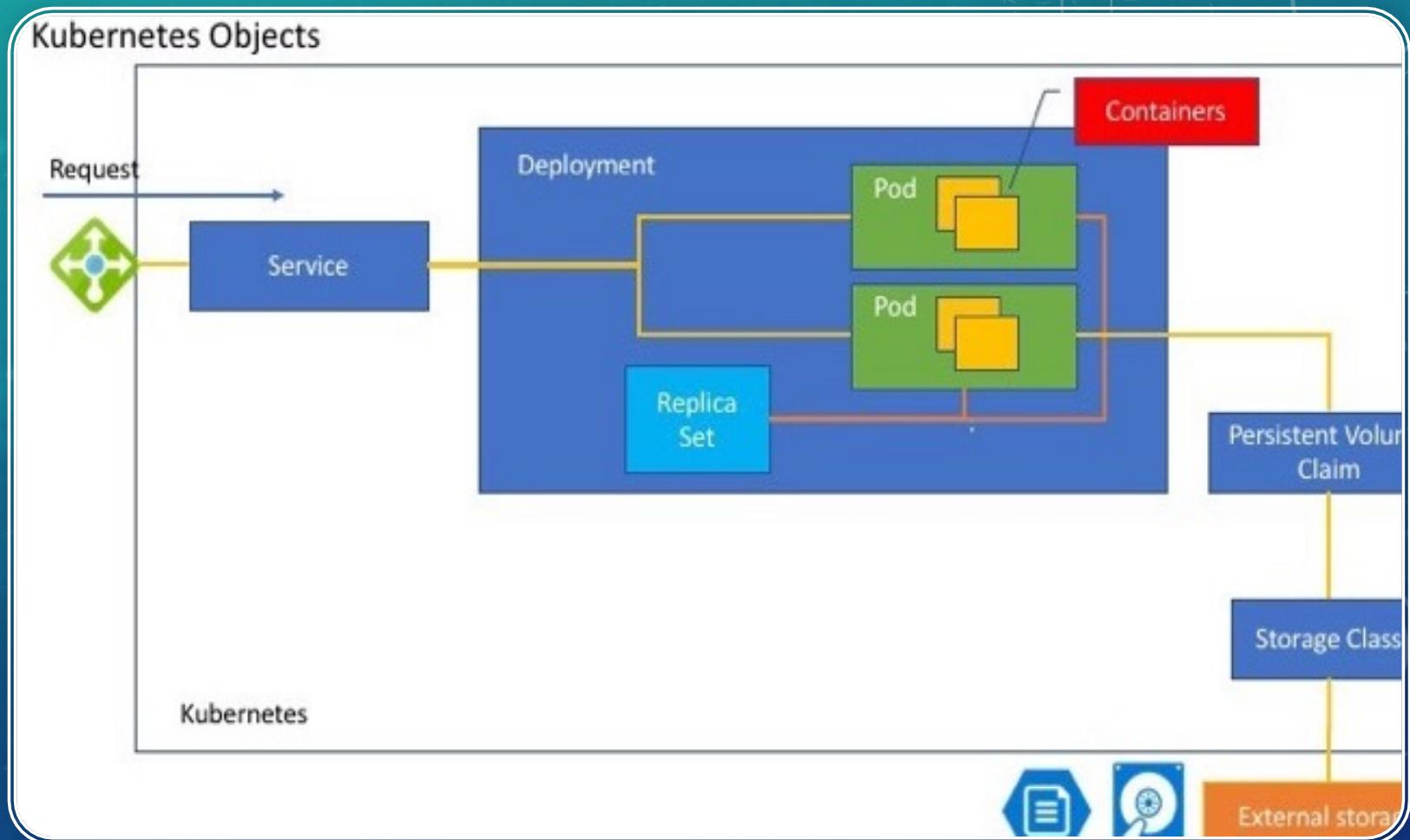
Manage Containers  
At Scale

Integrate IT Operations

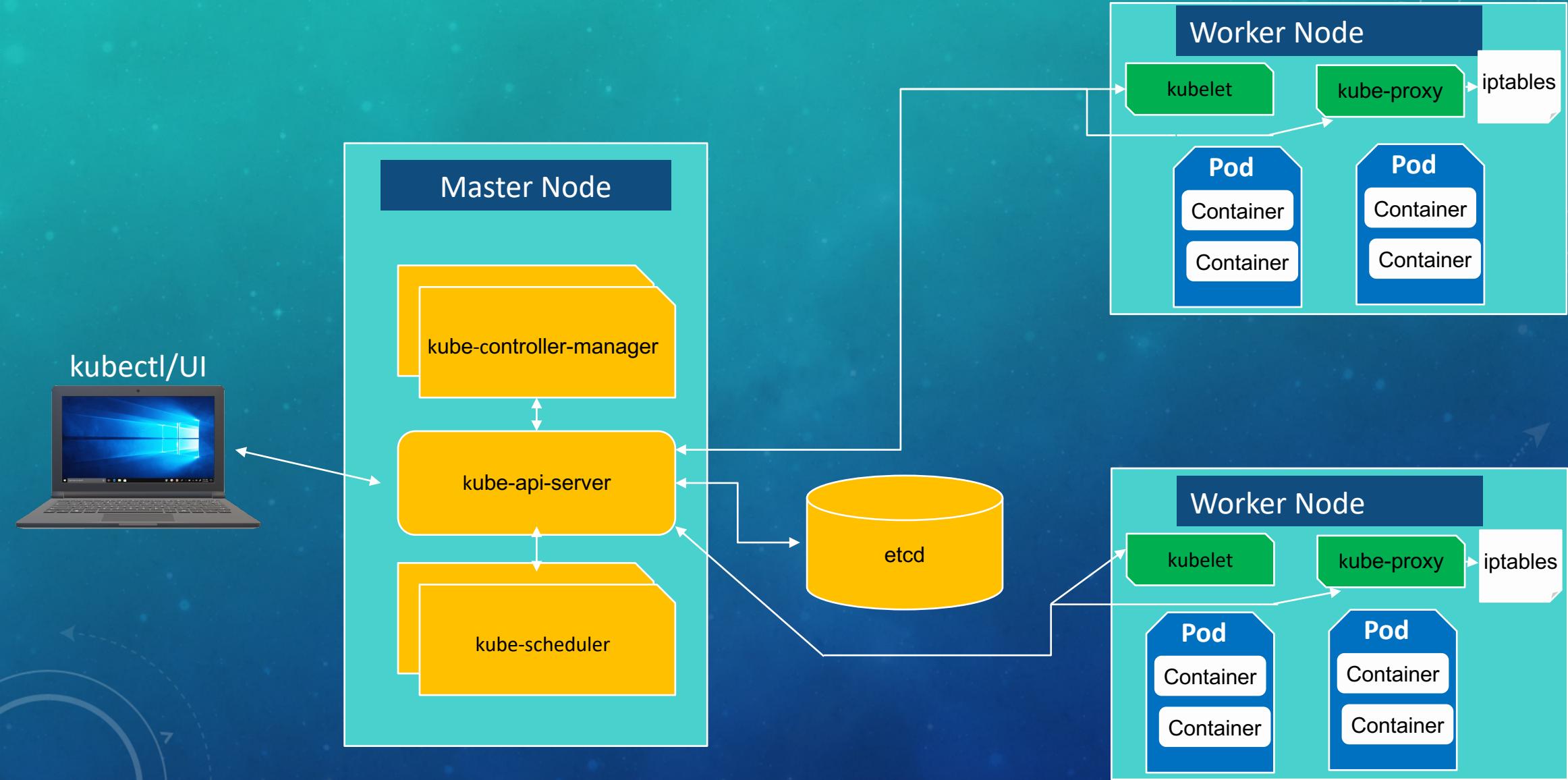
Enable Hybrid Multi-Cloud

# KUBERNETES (HELMMSMAN) – PUTTING THE PIECES TOGETHER

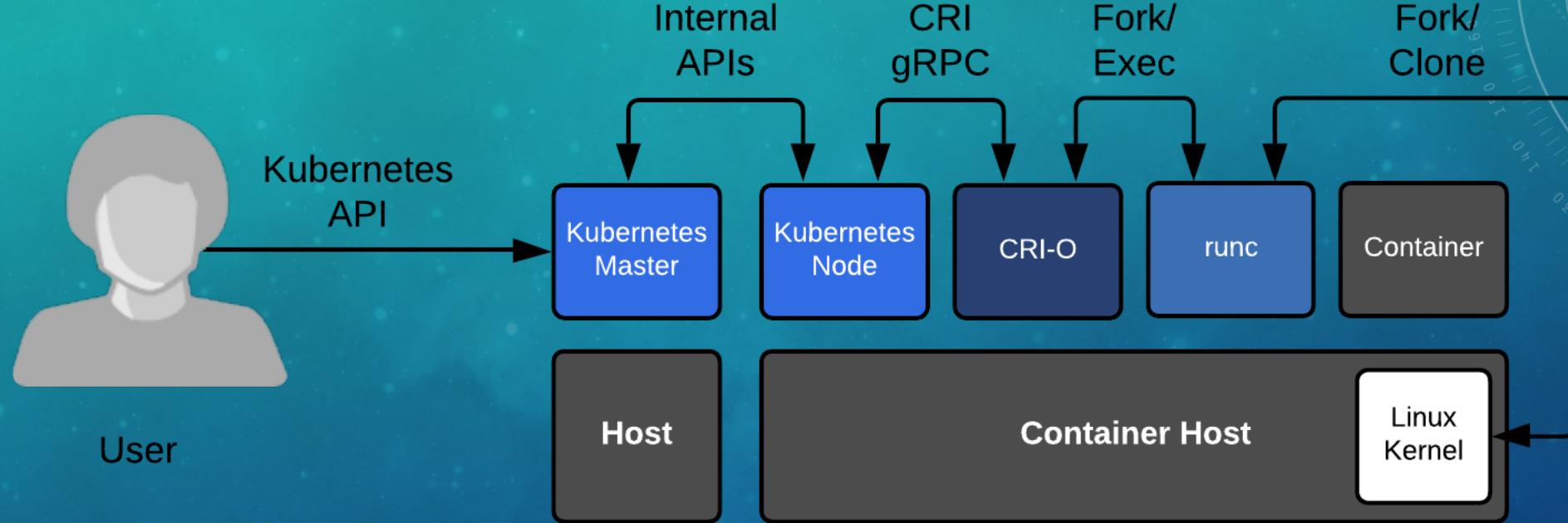
- Pod – Set of containers running in same execution environment/context (smallest unit in Kubernetes) [containers in pod share some Linux namespaces (Network, IPC, and PID if enabled) but each have own cgroup]
- Replica Set – makes sure correct number and types of pods are available
- Deployment –Manages replica sets for ease of new app version rollout.
- Service – Provides access point for pods/deployment as well as load balancing
- Persistent Volume Claim – provides storage volumes to container runtime (i.e. docker) by binding to persistent volumes
- Storage Class – groups storage so that it can be dynamically selected and provisioned
- Persistent Volume - Set of external storage defined to Kubernetes



# BASIC “PHYSICAL” KUBERNETES CLUSTER ARCHITECTURE



# PATH TO RUNNING CONTAINER IN KUBERNETES



How containers run in a Kubernetes cluster

See [RedHat blogpost](#) and [CRI-O homepage](#)

# High-Level Architecture & Kubernetes Features

## Current High-Level Architecture

Underlying Stack

### Application & Developer Services

- IBM Middleware, Open Source, 3<sup>rd</sup> Party

### Management Layer

- UI / Dashboard (for Managing Apps)

### Container Platform

- Kubernetes

### Container Engine

- CRI-O

### Operating System

- CoreOS, RHEL, Ubuntu, SUSE

### Local Virtual & Physical Infrastructure

- Bare Metal, z/VM, KVM

= required for OpenShift

## Features



Horizontal Scaling



Service Discovery & Load Balancing



Automated Rollouts & Rollbacks



Intelligent Scheduling



Self-Healing



Secret & Configuration Management

Kubernetes is an open source Orchestration Tool that allows automation of key container as scaling, upgrades etc.

# KUBERNETES NAMESPACES (NOT LINUX NAMESPACES)

- Virtual clusters all connected to the same “physical” cluster
- End user sees the cluster via current namespace (i.e. `kubectl get pods` will show pods in current namespace)
  - Divide cluster resources between different users
  - Resources (i.e. pods, replicaset, etc.) scoped by namespace (resource names unique within namespace)
- Initial Kubernetes Namespaces
  1. default: objects with no defined namespace
  2. kube-system: objects created by Kubernetes system
  3. kube-public: originally configured as readable by all users (even unauthenticated users) to make certain resources visible cluster-wide
- Make and view namespaces
  - `kubectl create namespace hi` [makes new namespace called hi]
  - `kubectl get namespace` [returns all namespaces on the “physical” cluster]

# OPERATORS: CONTROLLING YOUR CLUSTER FROM THE INSIDE



Write code to extend Kubernetes to  
automate tasks



Uses CRDs (Custom Resource Definitions) to  
define application resources



Popular applications made and deployed via  
operators for best-practices deployment



Ease of sharing via operator hub

# IBM BLOCKCHAIN PLATFORM OPERATOR

- Your blockchain components and console are custom resources using CRDs (custom resource definitions: ibpca, ibppeer, ibporderer, ibpconsole)
- CRDs enable users to control the application specific details (i.e. blockchain component configurations and console configurations) on top of the capabilities offered via regular OpenShift resources (i.e. deployments, routes, etc.)
- They are managed by the IBM Blockchain Platform operator which manages the lifecycles of these components

## 2. RED HAT OPENSHIFT CONTAINER PLATFORM

# CONTAINER CHALLENGES

## Container security

Image scanning, patching, and compliance

## Day 2 management

Installations, upgrades, and maintenance  
Integration of existing enterprise technology

## Application delivery

Monitoring, metering, and management  
Integration of existing developer tools



# Red Hat OpenShift

## Trusted enterprise Kubernetes

Continuous security, world-class support and services, and deep expertise to confidently run any application

## A cloud-like experience, everywhere

Full-stack automated operations on a consistent foundation across on-premises or hybrid cloud infrastructure

## Empowerment for developers to innovate

Ability to get applications to production sooner with a wide range of technologies and streamlined workflows



# Red Hat OpenShift - Hybrid Multi-Cloud platform



Red Hat  
Ansible Automation



Red Hat  
CloudForms



Red Hat  
Satellite



Red Hat  
Insights

# Red Hat OpenShift



IBM Cloud



Private



Edge



Z-Systems



Red Hat  
Enterprise Linux

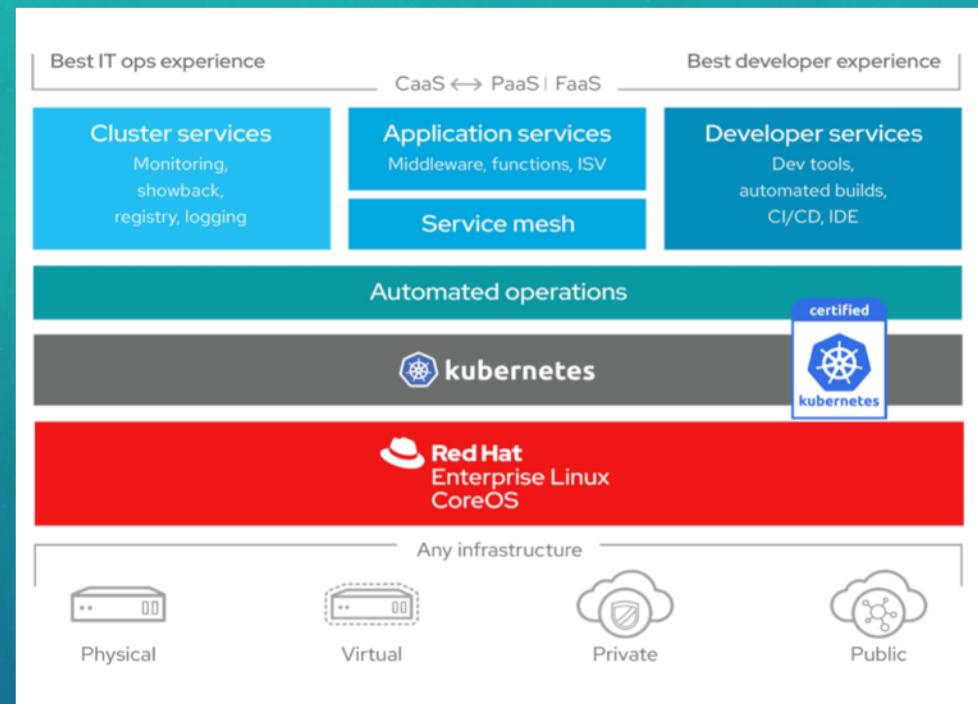


Red Hat  
Storage

CLUSTER STORAGE

CLIFF STORAGE

# OpenShift 4.X – A Smarter Kubernetes Platform



## Seamless Updates

- OpenShift retrieves the list of available updates
- Admin selects the target version
- OpenShift is updated over the air
- Auto-update support

A screenshot of the Red Hat OpenShift web interface. The left sidebar shows navigation options like Home, Catalog, Workloads, Networking, Storage, Builds, Monitoring, Administration, Cluster Settings, and Namespaces. The main panel displays "Cluster Settings" with tabs for Overview, Global Configuration, and Cluster Operators. Under Overview, it shows CHANNEL: fast, UPDATE STATUS: 4.10-0.2, CURRENT VERSION: 4.0-0.2, and CLUSTER ID: 754c289-02ee-4d32-879e-cd4a0e79499c. There is also a note about CLUSTER AUTOSCALER: Create Autoscaler and a prominent "Update" button.

Automated, full-stack installation from the container host to application services

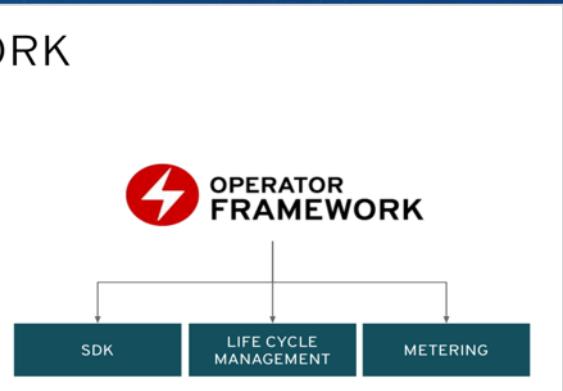
Seamless Kubernetes deployment to any cloud or on-premises environment

Autoscaling of cloud resources

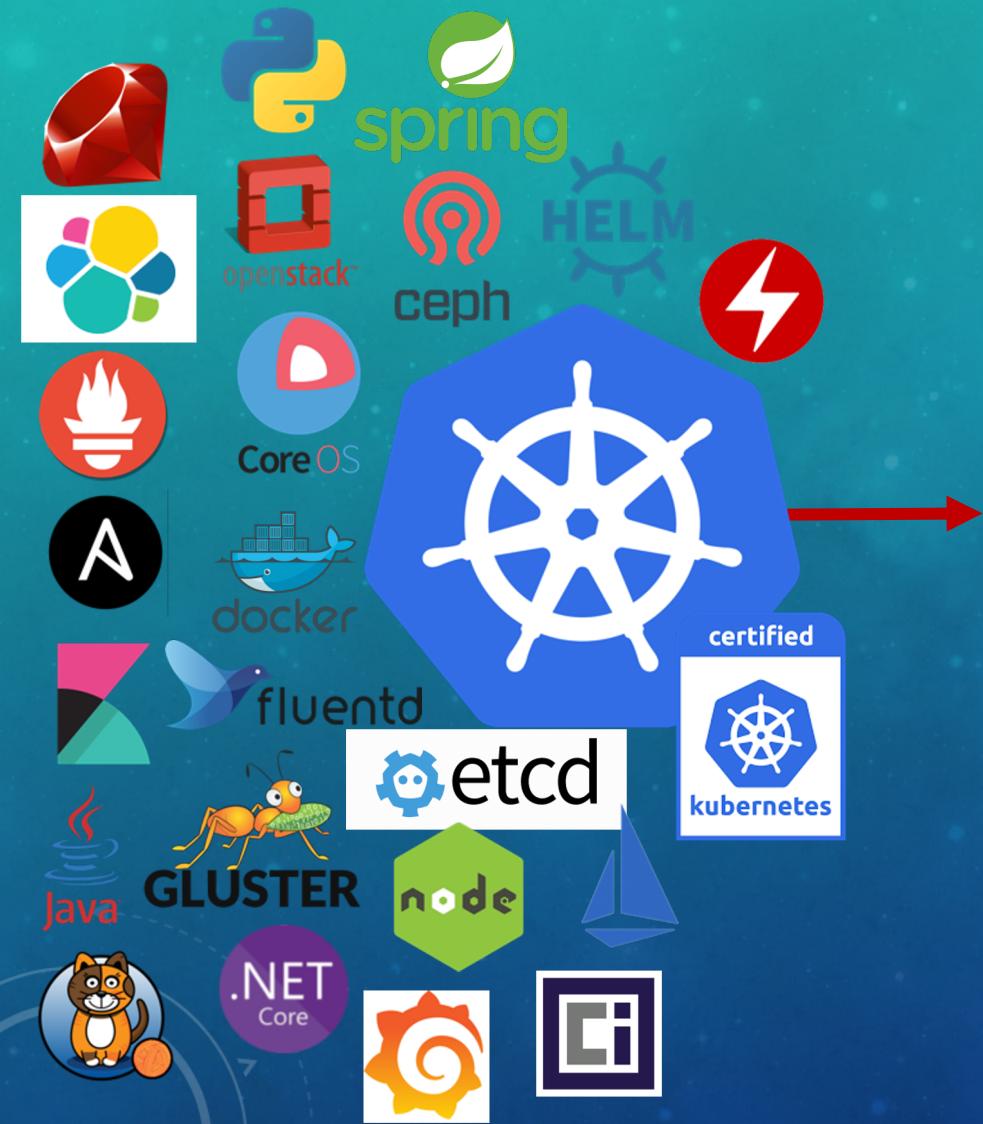
One-click updates for platform, services, and applications

## OPERATOR FRAMEWORK

Operators codify operational knowledge and workflows to automate life cycle management of containerized applications with Kubernetes



# How does RedHat create OpenShift?



okd

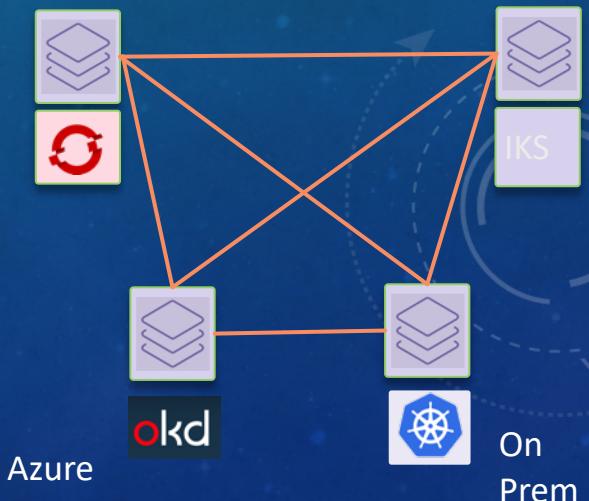
- OKD – Upstream Open Source Software
- Integrate additional OSS projects
- 100+ Integrations
- Validated OSS Innovation
- Partner Integration Platform

→  
openshift  
CONTAINER PLATFORM

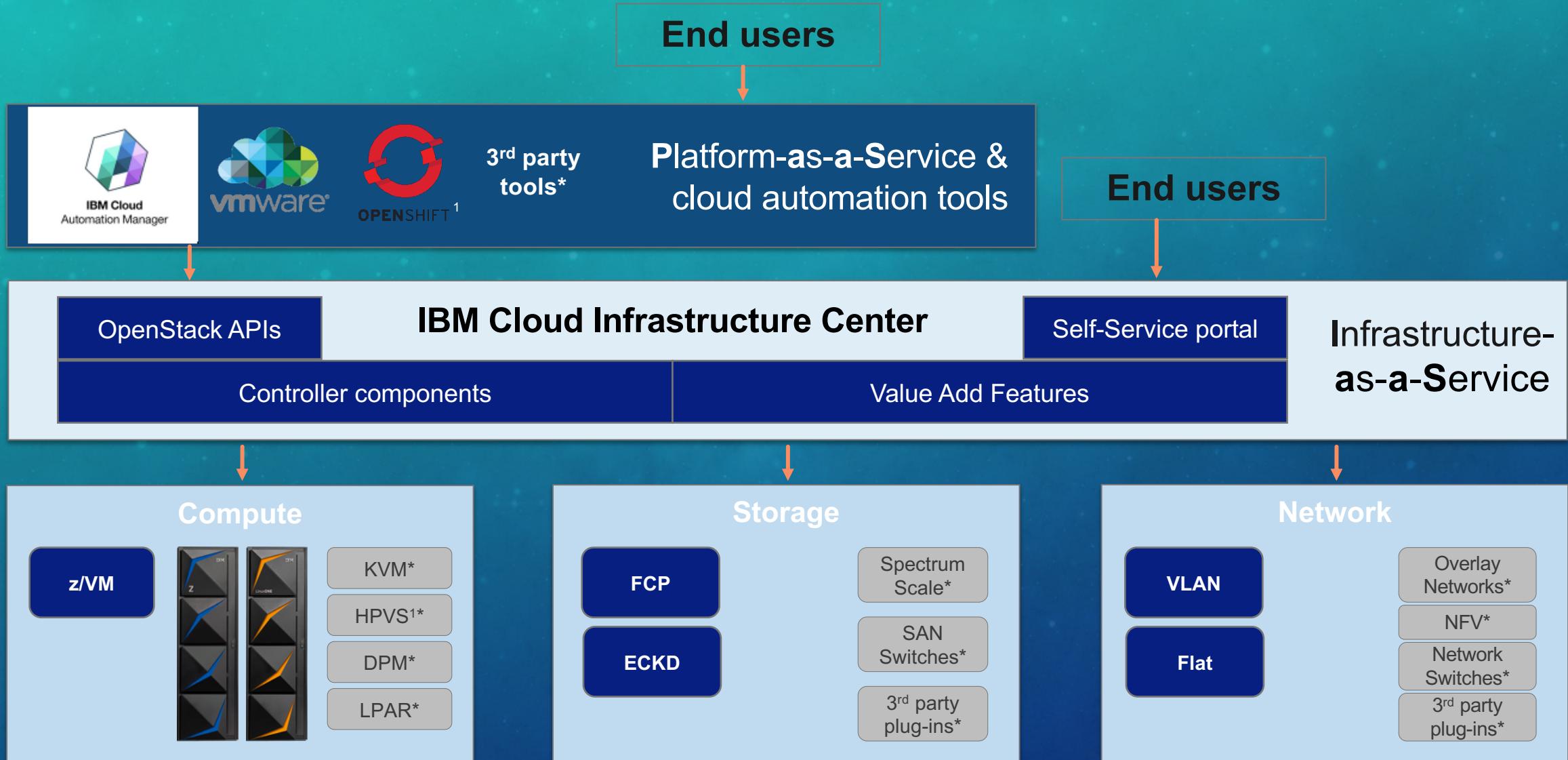


# IBM Blockchain Platform: Deploy anywhere

- IBM Blockchain Platform can be deployed wherever you want
  - **On-premises** or in **other cloud providers** for greater deployment flexibility, using any Kubernetes v1.16+ environment (e.g. Red Hat OpenShift, OKD, AKS, Rancher)
  - **IBM Cloud** for an IBM-managed service, using IBM Kubernetes Service
- **Fully heterogeneous**: different components can be deployed in different environments
- Caters for different vendor biases in the business network and **avoids lock-in**



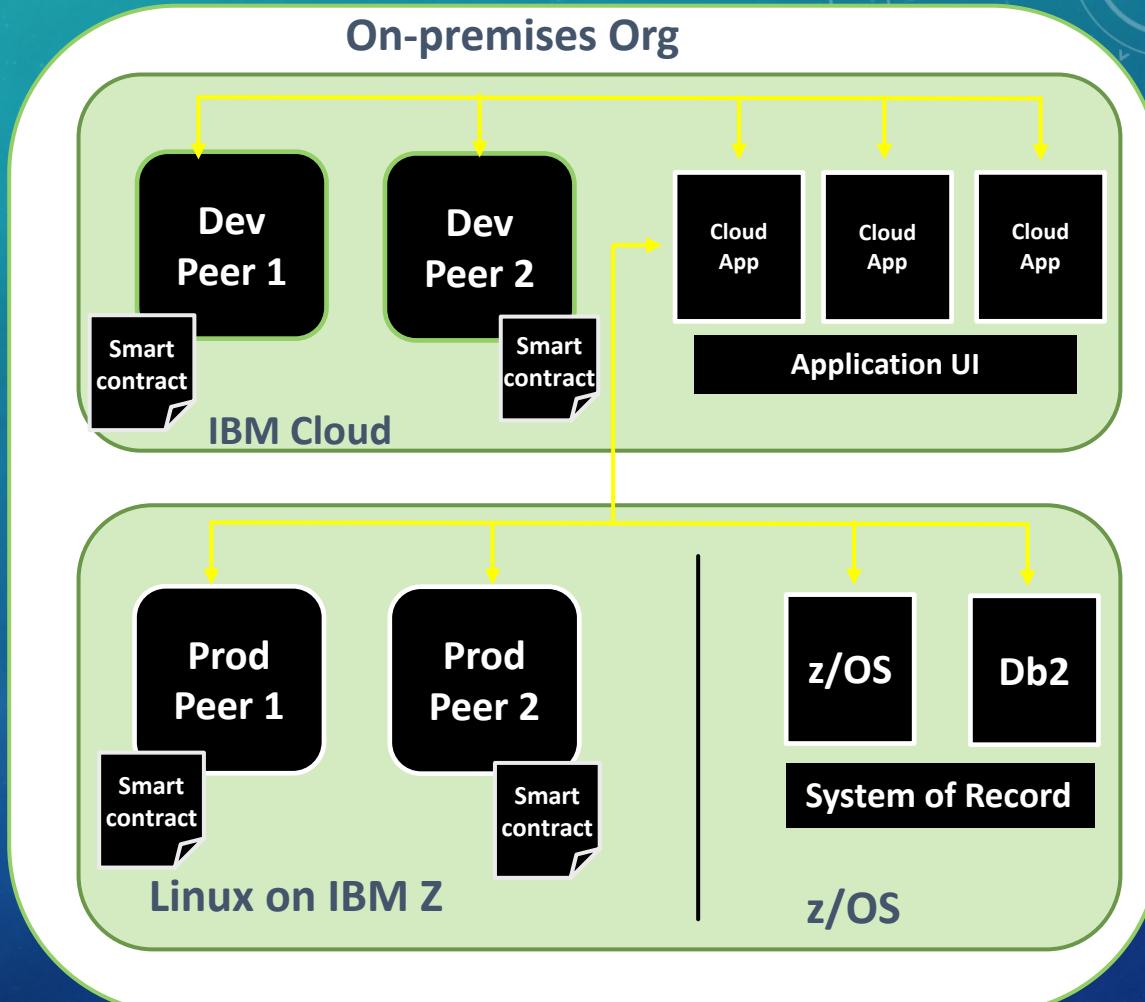
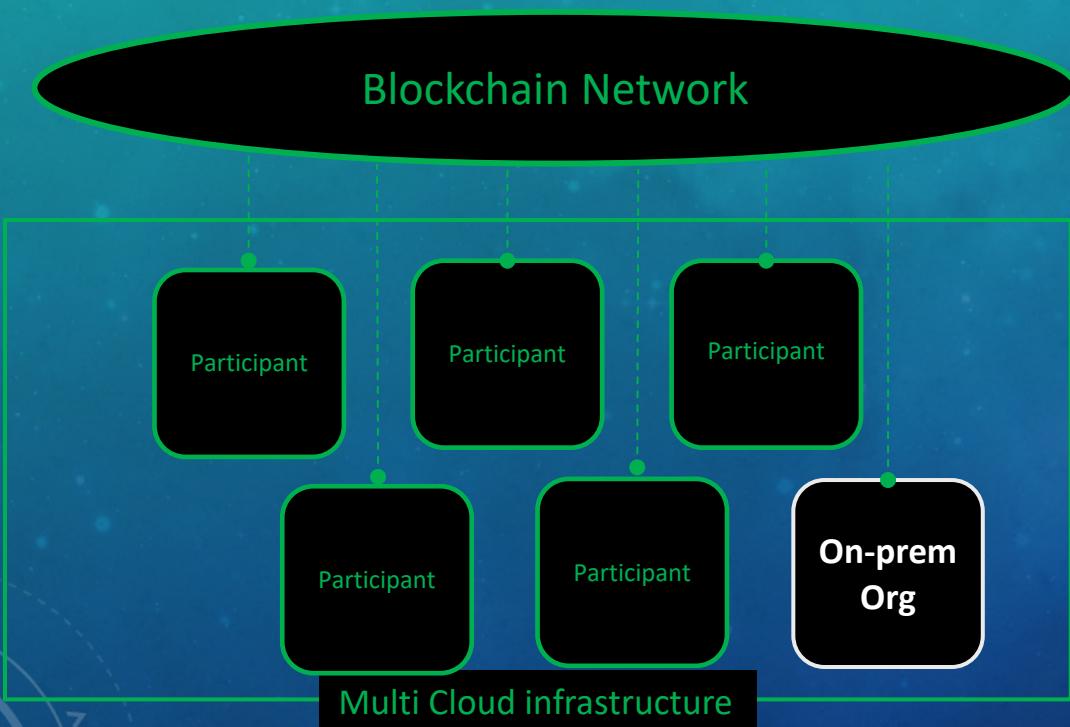
# IBM Z Cloud Big Picture



<sup>1</sup> IBM Hyper Protect Virtual Server

# IBM BLOCKCHAIN PLATFORM

1. IBM Cloud Primary Development Environment for new applications
2. Linux on IBM Z (Linux on Z) Provides collocation option for IBM Z apps with security and performance driven blockchain requirements for production
3. The modular approach enables seamless connection experience across development (dev) and production (prod) environments across multiple clouds

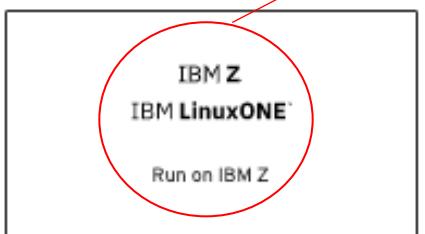
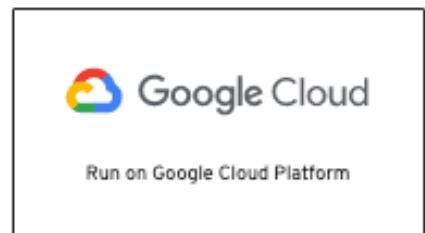


Consistent Development Experience

# TRY IT YOURSELF!

- RedHat OpenShift Container Platform free beta code release
  - Homogenous clusters only (i.e. all Linux on Z or all Linux on x86 nodes)
1. Learn OpenShift for free on the [Interactive Learning Portal](https://learn.openshift.com/introduction/) [<https://learn.openshift.com/introduction/>] with tutorials that spin up an OpenShift cluster for you to go through the exercises hands-on
  2. Try CDK Locally (with free RedHat online account) [<https://developers.redhat.com/products/codeready-containers>] which lets you manage your own OpenShift cluster using minishift to develop things on your local workstation
  3. Install on Linux on Z with trial [<https://cloud.redhat.com/openshift/install>]
  4. [Installing OpenShift Container Platform on Linux on Z Documentation](https://docs.openshift.com/container-platform/4.6/installing/installing_ibm_z/installing-ibm-z.html) [[https://docs.openshift.com/container-platform/4.6/installing/installing\\_ibm\\_z/installing-ibm-z.html](https://docs.openshift.com/container-platform/4.6/installing/installing_ibm_z/installing-ibm-z.html)]

# Available Now – OpenShift Container Platform 4.6



The screenshot shows the Red Hat OpenShift Container Platform 4.6 installation interface. The URL in the browser is <https://cloud.redhat.com/openshift/install/ibmz/user-provisioned>. The page title is "Install OpenShift Container Platform 4". On the left, there's a sidebar with "Clusters", "Subscriptions", "Documentation", "OperatorHub.io", "Cluster Manager Feedback", and "Report an OpenShift Bug". The main content area has a heading "Install on IBM Z with User-Provisioned Infrastructure". It includes a note about telemetry data collection, a link to the official documentation, and a "Get started" button. Below this, there's a "Downloads" section with links for "OpenShift Installer", "Pull Secret", "Red Hat Enterprise Linux CoreOS (RHCOS)", and "Command-Line Interface". Each download link has a "Download" button. A red arrow points from the "Run on IBM Z" box in the bottom-left towards the "Install on IBM Z" section of the interface.