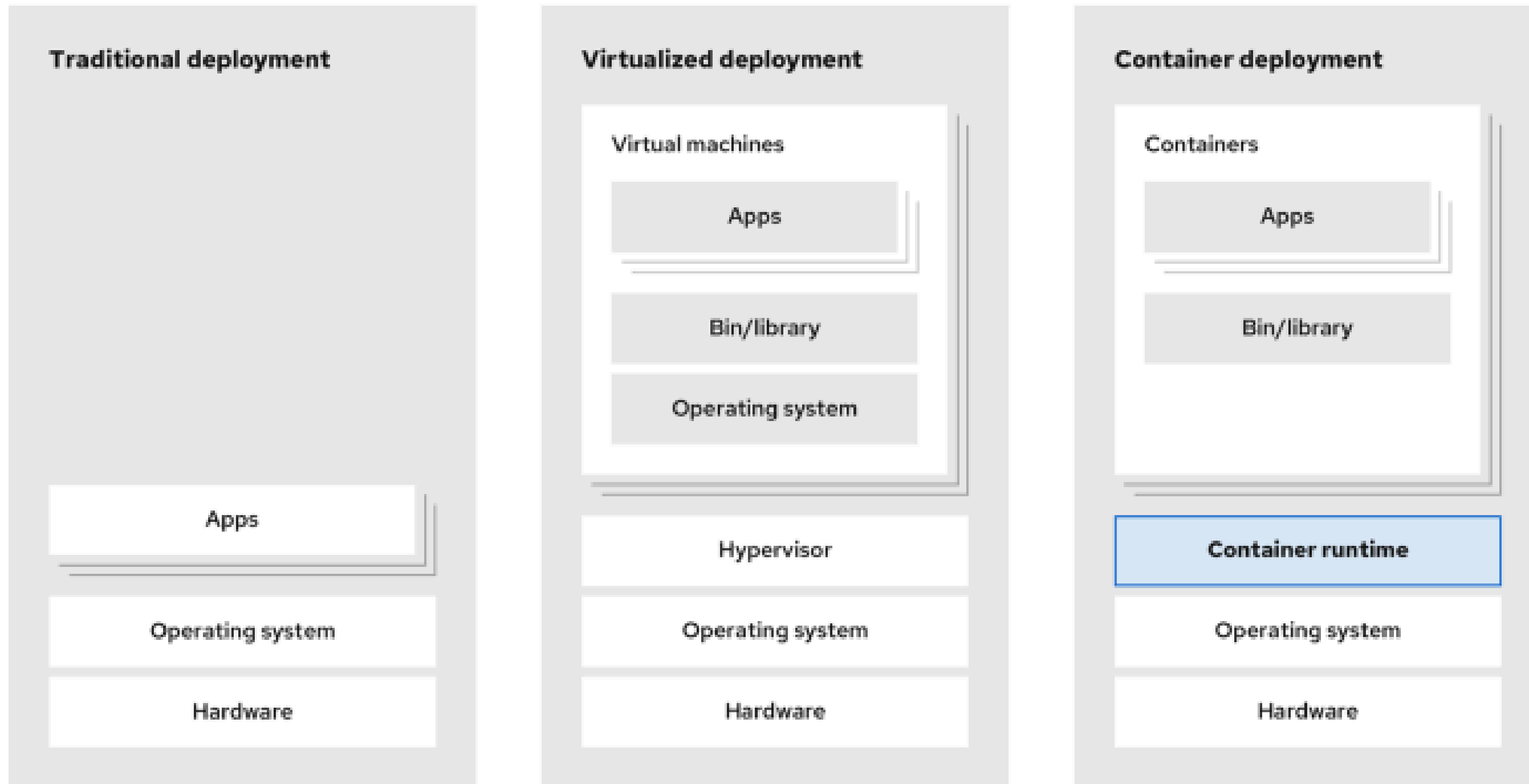


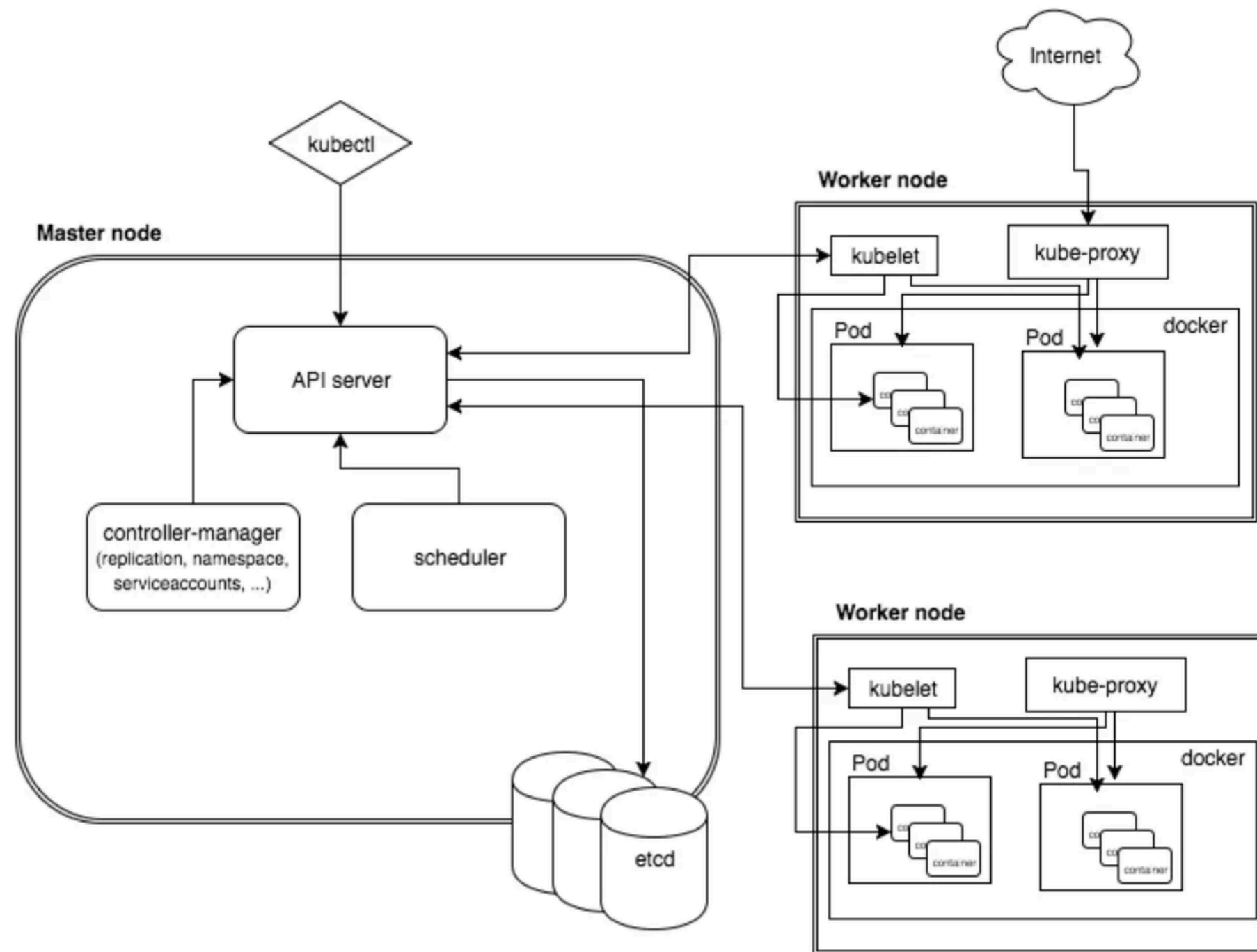
Agenda

- **Kubernetes 101 - Highlevel Overview**
- **OpenShift 101 - Highlevel Overview**
- **Deployment on OCP (OpenShift Container Platform)**
- **Wells Internal Sizing, RACI, Cluster Details**
- **OECM on OCP**
- **Target Product Status**
- **Current Status (WIP)**

Evolution of Container Technology



Kubernetes 101 - Highlevel Overview



Orchestration of Containers: Kubernetes automates the deployment, scaling, and management of containerized applications, making it easier to handle complex applications spread across multiple environments.

Master Nodes

- API Server
- ETC
- Scheduler
- Control Manager

Worker Node

- Kubelet
- Kubeproxy
- Pod

Client: kubectl

Kubernetes 101 - Pod

- Atomic unit or smallest “unit of work” of Kubernetes.
- Pods are one or MORE containers that share volumes, a network namespace, and are a part of a single context.
- They are also ephemeral

```
apiVersion: v1
kind: Pod
metadata:
  name: multi-container-pod-example # Name of the Pod
spec:
  containers:
    # Container 1: Apache HTTP Server (httpd)
    - name: apache-container        # Name of the first container
      image: httpd:latest           # Image for the Apache HTTP server
      ports:
        - containerPort: 80        # Port exposed by the Apache server
      # No custom command is needed; defaults to running httpd service

    # Container 2: Alpine Linux Shell
    - name: alpine-container        # Name of the second container
      image: alpine:latest          # Image for Alpine Linux
      command: ["/bin/sh"]          # Command to run when the container starts
      args: ["-c", "while true; do echo 'Hello, Kubernetes from Alpine!'; sleep 5;
done"]
      # The command runs a shell that outputs a message every 5 seconds
```

Kubernetes 101 - Storage

Pods are useful but often need data sharing between containers or data persistence. To enable this, Kubernetes offers:

- **Volumes:** Shared, temporary storage tied to a Pod's lifecycle, persisting across container restarts but deleted with the Pod.
- **PersistentVolumes (PVs):** Cluster-wide storage that retains data beyond the Pod's lifecycle.
- **PersistentVolumeClaims (PVCs):** Requests for storage used by Pods to access PersistentVolumes.
- **StorageClasses:** Define and manage different storage types, enabling dynamic provisioning of PersistentVolumes.

Kubernetes 101 - Volume

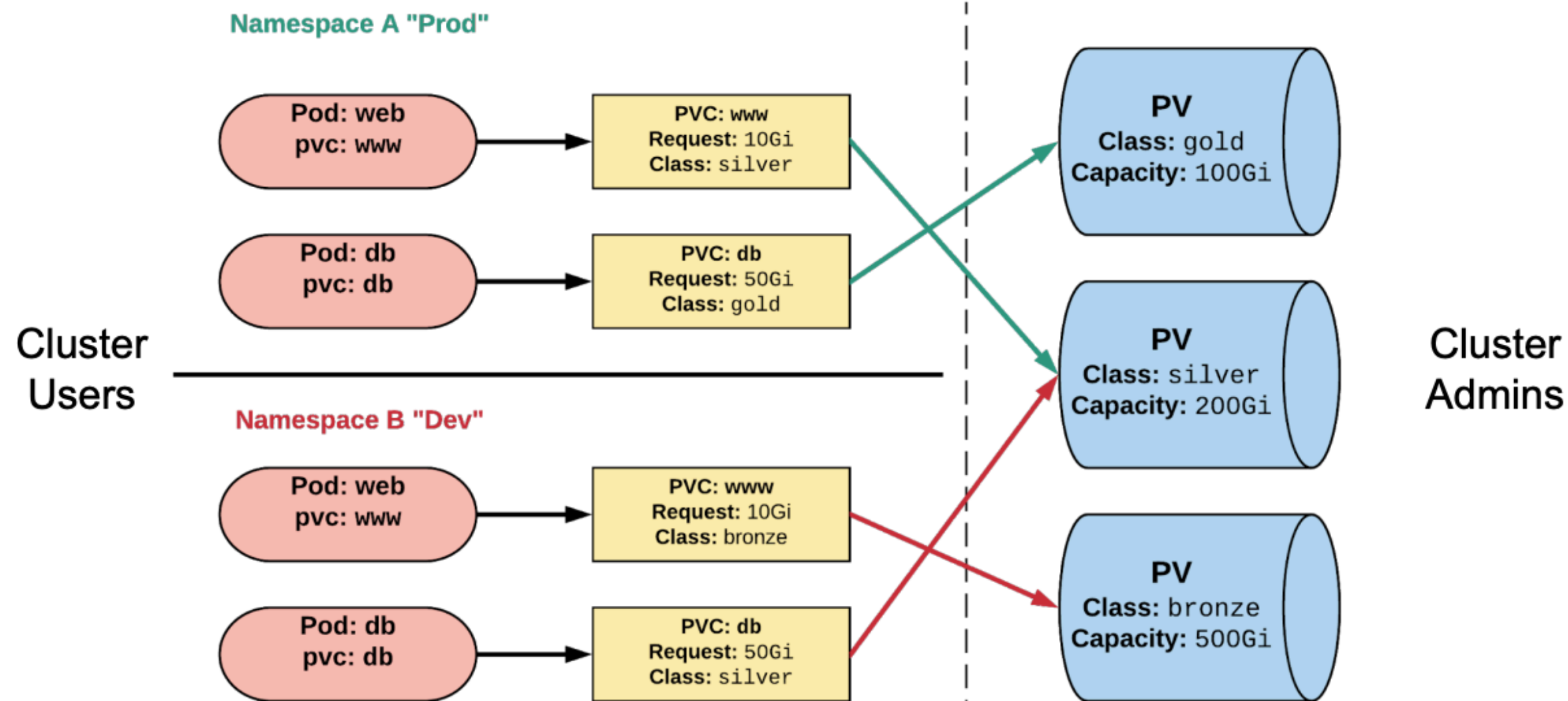
- **Storage linked to the Pod's lifecycle.**
- **A Pod can have multiple types of volumes.**
- **Usable by any container within the Pod.**
- **Survives Pod restarts; durability depends on the volume type.**

```
apiVersion: v1
kind: Pod
metadata:
  name: multi-container-volume-example # Name of the Pod
spec:
  containers:
    # Container 1: Apache HTTP Server (httpd)
    - name: apache-container          # Name of the first container
      image: httpd:latest             # Image for the Apache HTTP server
      volumeMounts:
        - name: shared-data          # Referencing the Pod volume by name
          mountPath: /usr/local/apache2/htdocs # Path inside the container where the volume is mounted

    # Container 2: Alpine container for writing data to the shared volume
    - name: alpine-writer-container   # Name of the second container
      image: alpine:latest            # Image for Alpine Linux
      command: ["/bin/sh", "-c"]      # Command to run when the container starts
      args:
        - while true; do echo "Hello from Alpine at $(date)" >> /data/index.html; sleep 5; done
      # Command writes a timestamped message to the shared file every 5 seconds
      volumeMounts:
        - name: shared-data          # Referencing the same Pod volume by name
          mountPath: /data            # Path inside the container where the volume is mounted

  volumes:
    - name: shared-data              # Defining the volume used by both containers
      emptyDir: {}                  # Type of volume that is created as an empty directory
```

Kubernetes 101 - Persistent Volume



PersistentVolumeClaims (PVCs):

- Namespaced requests for storage within a Kubernetes cluster.
- Specify requirements rather than directly mapping to a specific storage resource.
- Ensure portability of storage claims, allowing applications to access storage across various backends or providers.

Persistent Volumes (PVs):

- Represent a storage resource within a Kubernetes cluster.
- Linked to a backing storage provider such as NFS, GCEPersistentDisk, or RBD.
- Typically provisioned by an administrator.
- Managed independently from Pod lifecycles.
- Cannot be attached directly to a Pod and must be accessed via a PersistentVolumeClaim (PVC).

Kubernetes 101 - PV/PVC Spec File

PersistentVolume (PV) Specification

apiVersion: v1

kind: PersistentVolume

metadata:

name: nfsserver # Name of the PersistentVolume

spec:

capacity:

storage: 50Gi # Total available storage capacity

volumeMode: Filesystem # Type of volume (Filesystem or Block)

accessModes:

- ReadWriteOnce
- ReadWriteMany

persistentVolumeReclaimPolicy: Delete # Behavior when PVC is deleted (Delete or Retain)

storageClassName: silver # Storage class name for PVCs to reference

mountOptions:

- hard # Ensures hard mount for stability
- nfsvers=4.1 # Specifies NFS version

nfs:

path: /exports # Path on the NFS server

server: 172.22.0.42 # NFS server IP address

PersistentVolumeClaim (PVC) Specification

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: pvc-sc-example # Name of the PersistentVolumeClaim

spec:

accessModes:

- ReadWriteOnce # Access mode requested by the PVC

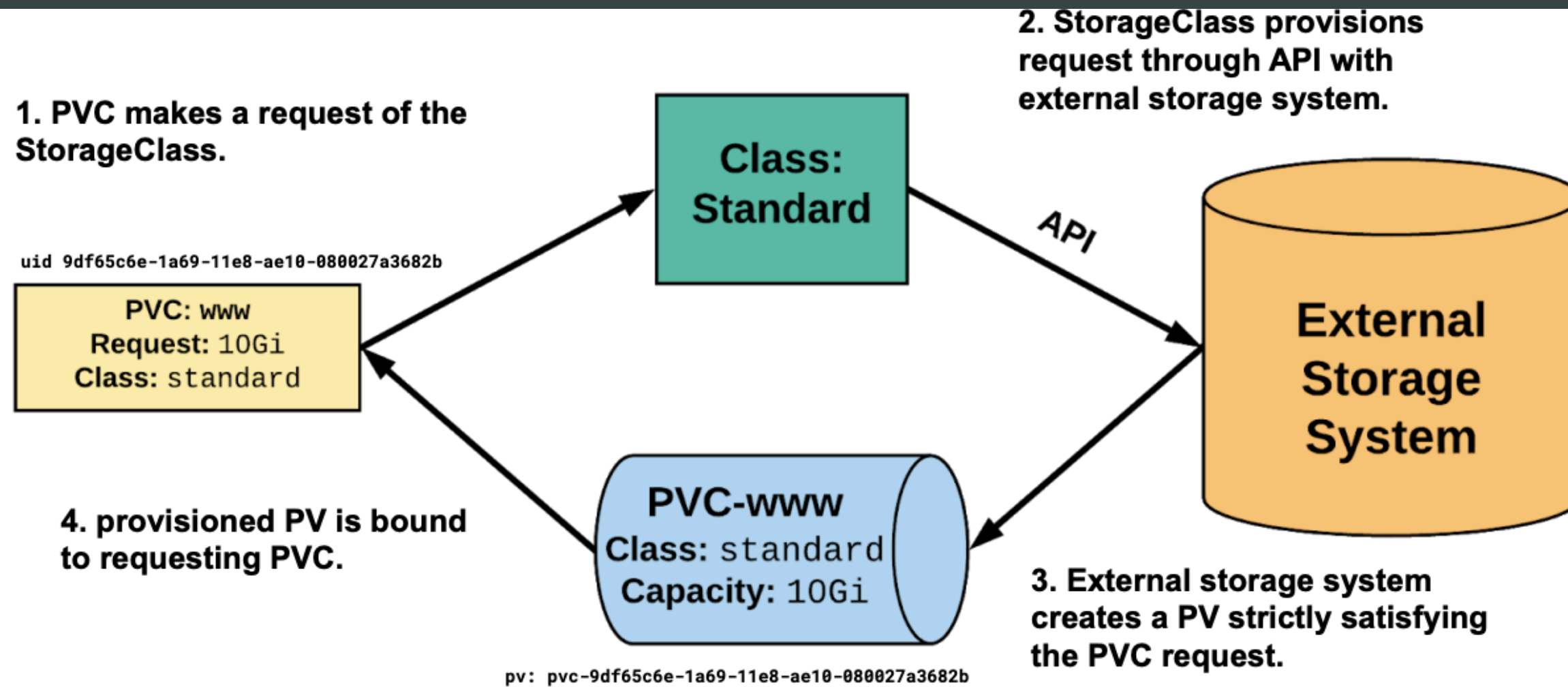
resources:

requests:

storage: 1Gi # Storage size requested by the PVC

storageClassName: silver # References the storage class defined in the PV

Kubernetes 101 - Storage Classes



kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: standard # Name of the storage class

provisioner: kubernetes.io/gce-pd # The driver used for external storage provisioning

parameters:

type: pd-standard # Type of persistent disk (e.g., pd-standard, pd-ssd)

zones: us-central1-a, us-central1-b # Availability zones for the storage

reclaimPolicy: Delete # Policy for reclaiming the storage asset after PVC deletion

Kubernetes 101 - Wells Reference

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: netapp-trident-block # Name of the StorageClass for block storage

provisioner: netapp.io/trident # Provisioner specific to NetApp Trident

parameters:

backendType: ontap-san # Backend type for block storage (e.g., ontap-san)

spaceReserve: thin # Optional: Thin or thick provisioning

encryption: true # Optional: Enable encryption for the volume

fsType: ext4 # Filesystem type for block storage (e.g., ext4, xfs)

protocols: iscsi # Protocol used for block storage (e.g., iscsi, fc)

snapshotPolicy: default # Optional: Snapshot policy to use

igroup: default # Optional: Specifies the initiator group for access

reclaimPolicy: Retain # Policy for reclaiming storage after PVC deletion

volumeBindingMode: Immediate # When the volume should be provisioned (Immediate or WaitForFirstConsumer)

Kubernetes 101 - Deployment / Replicaset

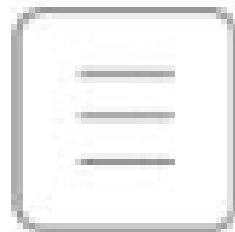
Deployment

Updates and Rollback

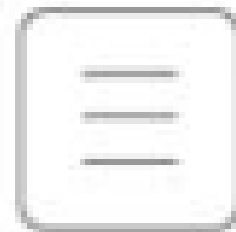
ReplicaSet

Self-healing, scalable, desired state

Pod

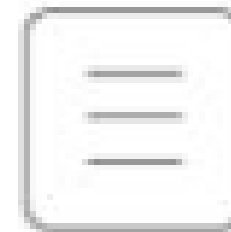


Pod



...

Pod



Kubernetes 101 - Deployment / Replicaset - Spec

apiVersion: apps/v1

kind: Deployment

metadata:

name: deploy-example # Name of the deployment

spec:

replicas: 3 # Number of Pod replicas

revisionHistoryLimit: 3 # Number of old ReplicaSets to retain

selector:

matchLabels:

app: nginx # Label to match for Pod selection

env: prod # Additional label for environment

strategy:

type: RollingUpdate # Deployment strategy type

rollingUpdate:

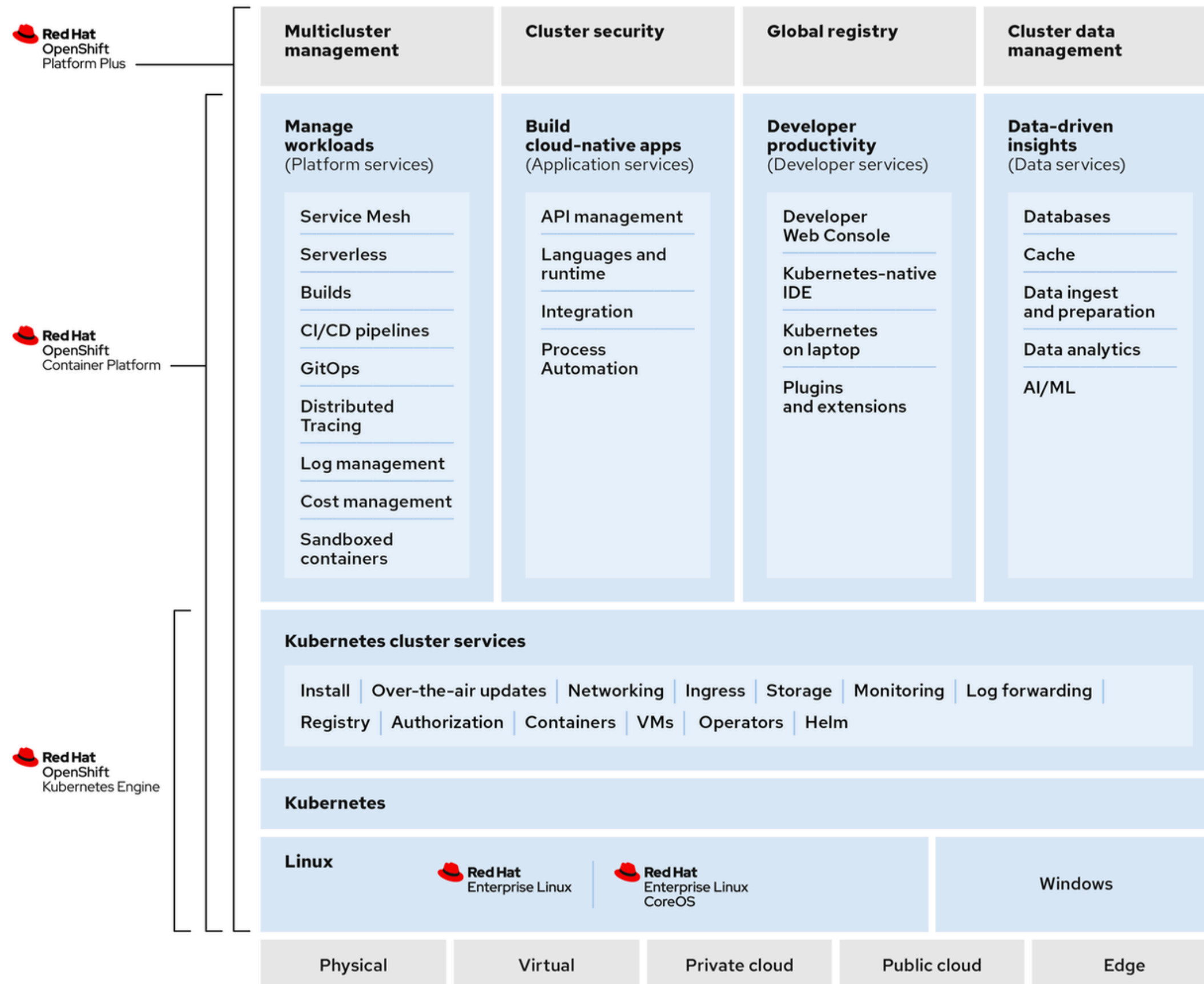
maxSurge: 1 # Maximum number of Pods that can be created above the desired number during update

maxUnavailable: 0 # Maximum number of Pods that can be unavailable during the update


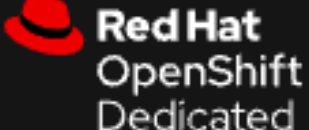
template: # Pod template definition


<pod definition>

OpenShift 101



WellsFargo - Openshift Cluster



 Developer

+Add

Topology

Observe

Search

Functions

Builds



Pipelines


Helm

Project

ConfigMaps

Secrets



 Administrator

Home

Operators

Workloads

Virtualization

Serverless

Networking

Storage

Builds

Pipelines

User Management

Administration

Workloads

Pods

Deployments

DeploymentConfigs

StatefulSets

Secrets

ConfigMaps

CronJobs

Jobs

DaemonSets

ReplicaSets

ReplicationControllers

HorizontalPodAutoscalers

PodDisruptionBudgets

WellsFargo - Openshift Cluster - Operator

Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace.

Name ▾





Redhat

/

Name

Redhat ✕

Clear all filters

Name ▴ ▾	Managed Namespaces ▴ ▾	Status	Last updated	Provided APIs
<div><div><div>Red Hat OpenShift Dev Spaces</div><div>3.16.1 provided by Red Hat</div></div></div>	<div><div>NS</div><div>openshift</div><div>The operator is running in crw but is managing this namespace</div></div>	<div><div>✓</div><div>Succeeded</div></div>	<div><div>🌐</div><div>3 Nov 2024, 23:38</div></div>	<div><div>Red Hat OpenShift Dev Spaces instance Specification</div><div>⋮</div></div>
<div><div><div>Red Hat OpenShift Pipelines</div><div>1.15.2 provided by Red Hat</div></div></div>	<div><div>NS</div><div>openshift</div><div>The operator is running in openshift-operators but is managing this namespace</div></div>	<div><div>✓</div><div>Succeeded</div></div>	<div><div>🌐</div><div>4 Nov 2024, 00:08</div></div>	<div><div>-</div><div>⋮</div></div>
<div><div><div>Red Hat OpenShift AI</div><div>2.15.0 provided by Red Hat, Inc.</div></div></div>	<div><div>NS</div><div>openshift</div><div>The operator is running in redhat-ods-operator but is managing this namespace</div></div>	<div><div>✓</div><div>Succeeded</div></div>	<div><div>🌐</div><div>13 Nov 2024, 19:27</div></div>	<div><div>Data Science Cluster DSC Initialization FeatureTracker</div><div>⋮</div></div>
<div><div><div>Red Hat OpenShift Serverless</div><div>1.34.0 provided by Red Hat</div></div></div>	<div><div>NS</div><div>openshift</div><div>The operator is running in serverless-operator but is managing this namespace</div></div>	<div><div>✓</div><div>Succeeded</div></div>	<div><div>🌐</div><div>3 Nov 2024, 23:31</div></div>	<div><div>Knative Serving Knative Eventing Knative Kafka</div><div>⋮</div></div>

WellsFargo - Openshift Cluster - Wells RACI

WellsFargo - Openshift Cluster - TCOO

WellsFargo - Openshift Cluster - DTI Architecture

WellsFargo - OECM Architecture in Openshift

WellsFargo - Helm Chart - OECM Completed

WellsFargo - Helm Chart - OECM Ongoing Activity

WellsFargo - Helm Chart - OECM Apps Ready (Draft)

WellsFargo - Helm Chart

THANK YOU

