The Db2 Operator: **Present and Future**

Aruna De Silva
Architect, Db2 on CPD/OpenShift/k8s (a.k.a. Db2U)

Janpreet Singh Chandhok
Software Engineer, Db2 on CPD/OpenShift/k8s (a.k.a. Db2U)

Modernization with Db2 on IBM Cloud Pak for Data, Red Hat OpenShift, Kubernetes and Public Cloud Providers

Agenda

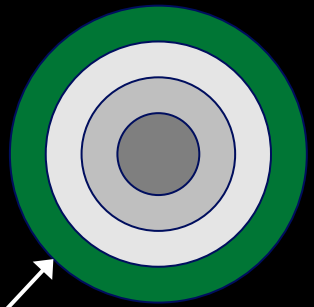
- ❑ The Db2 Universal (Db2U) Container
  - ❑ Overview
  - ❑ Container Hierarchy
- ❑ Db2 “Go” Operator
- ❑ Db2U Certification
- ❑ Db2U Ecosystem
- ❑ Db2U Architecture
  - ❑ Overview
  - ❑ Kubernetes Resource Model
  - ❑ Storage
- ❑ Db2U - Elevated Value
- ❑ Db2U Security Posture
  - ❑ Non-Privileged
  - ❑ SA & SCC/PSP
- ❑ Db2U – A Survey of Current Capabilities and Future Direction
  - ❑ Overview
  - ❑ Limitations in Current Implementation
  - ❑ Next Gen – Core Capabilities
  - ❑ Next Gen – Cloud-native Backup and Restore
  - ❑ Next Gen – Cloud-native Audit Facility
  - ❑ Next Gen – Cloud-native Log Streaming
  - ❑ Next Gen – Public Cloud Provider Alignment

Db2U: Containers, Operator, Certification and Ecosystem

- ❑ The Db2 Universal (Db2U) Container
  - ❑ Overview
  - ❑ Container Hierarchy
- ❑ Db2 “Go” Operator
- ❑ Db2U Certification
- ❑ Db2U Ecosystem

The Db2 Universal (Db2U) Container

- Db2 “Universal” (Db2U) Container – driving Db2 modernization on IBM Cloud Pak for Data, Red Hat OpenShift and Kubernetes
  - Microservice architecture
  - Flexible, tailorable form factor – OLTP (Db2), OLAP (Db2 Warehouse)
    - Transaction & data volumes
    - Query patterns & performance requirements
  - Enable pre-built configurations defining the fabric for “infrastructure as code”
    - Portable, secure & certified
    - Ready in minutes
    - Unified environments (consistency through Dev → Test → QA → Prod, etc.)
  - Core to a growing ecosystem of decoupled services

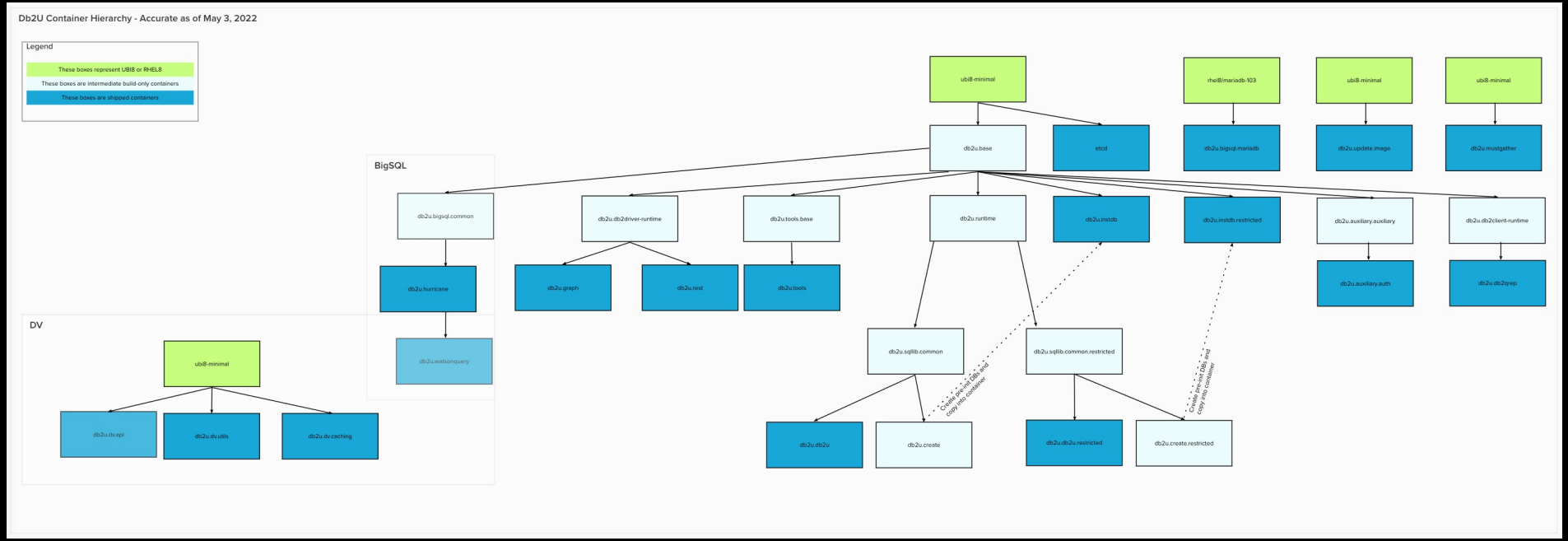


The Db2U Container Hierarchy

Db2U Container Hierarchy

An ecosystem of containers:

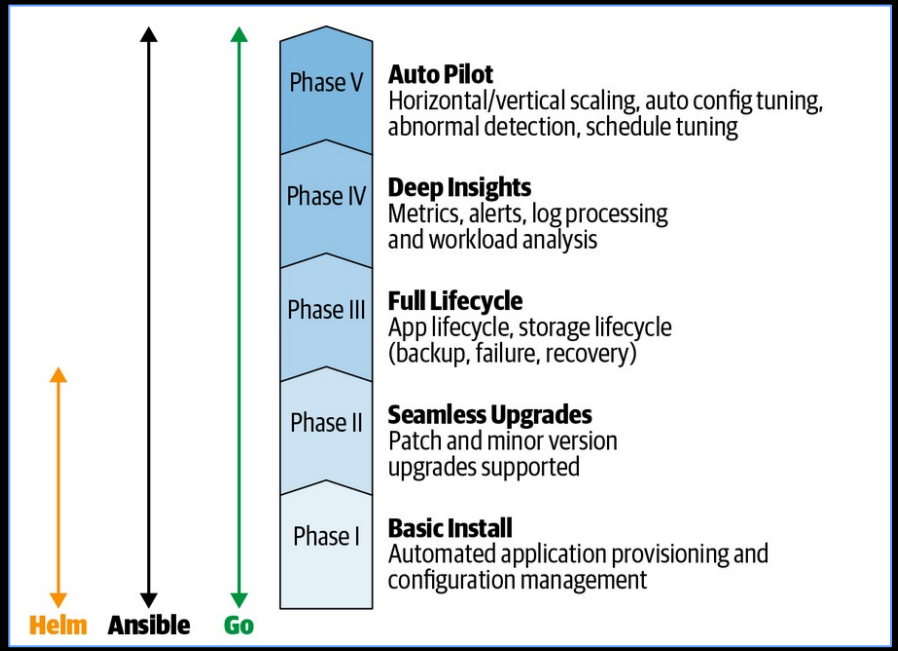
- ❑ Db2U, BigSQL, Data Virtualization (DV)
- ❑ Add-ons: REST, Graph, Replication (Q-rep)
- ❑ Internal: FVT, Storage Certifier, Release Certifier



The Db2 “Go” Operator

- **Operational Management**
  - Package, Configure, Deploy & Manage
  - Deployment
  - Management
- Measured for completeness by a **maturity model**
  - Provides a glimpse at the Db2 Operator roadmap
    - Currently expanding Phase III capabilities
- **Delivery**
  - IBM Operator Catalog
    - Supports Air Gap
  - Red Hat Marketplace

The Operator **Maturity Model**

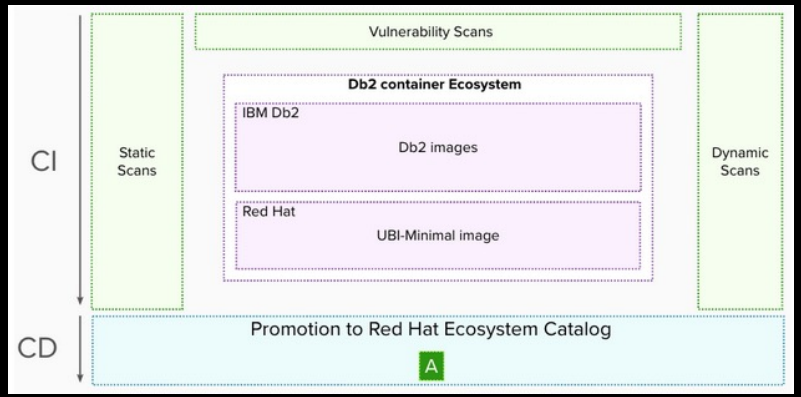


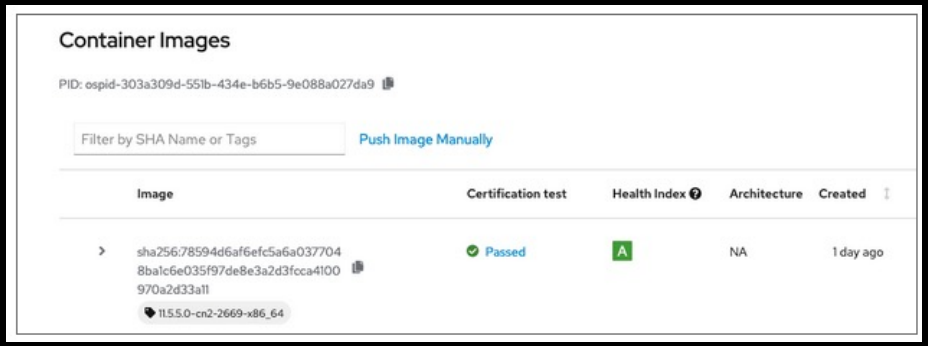
Db2U Certification

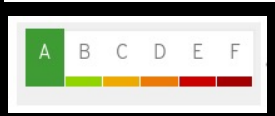
IBM **Db2**
+
RED HAT OPENSHIFT Container Platform

Db2 on OpenShift/Cloud Pak for Data

- IBM Kubernetes Certification: Certified Offering
- Grade in Red Hat Ecosystem Catalog: **A**
- Scanning with *Twistlock* from Sysdig







Db2U Ecosystem

- **Ecosystem of Components**
  - The Db2u Engine Container
  - A complete ecosystem of decoupled services
    No impact to Db2u engine
    Instantiate, manage, upgrade, decommission separately
    Ability to separate for load balancing, performance consistency
    - REST API
    - Unified Console (DMC)
    - Built-in LDAP Service
    - Continuous Availability (HADR / Q-replication)
    - Db2 Graph
  - IBM Internal Ecosystem
    - Performance & validation testing
    - Storage certification

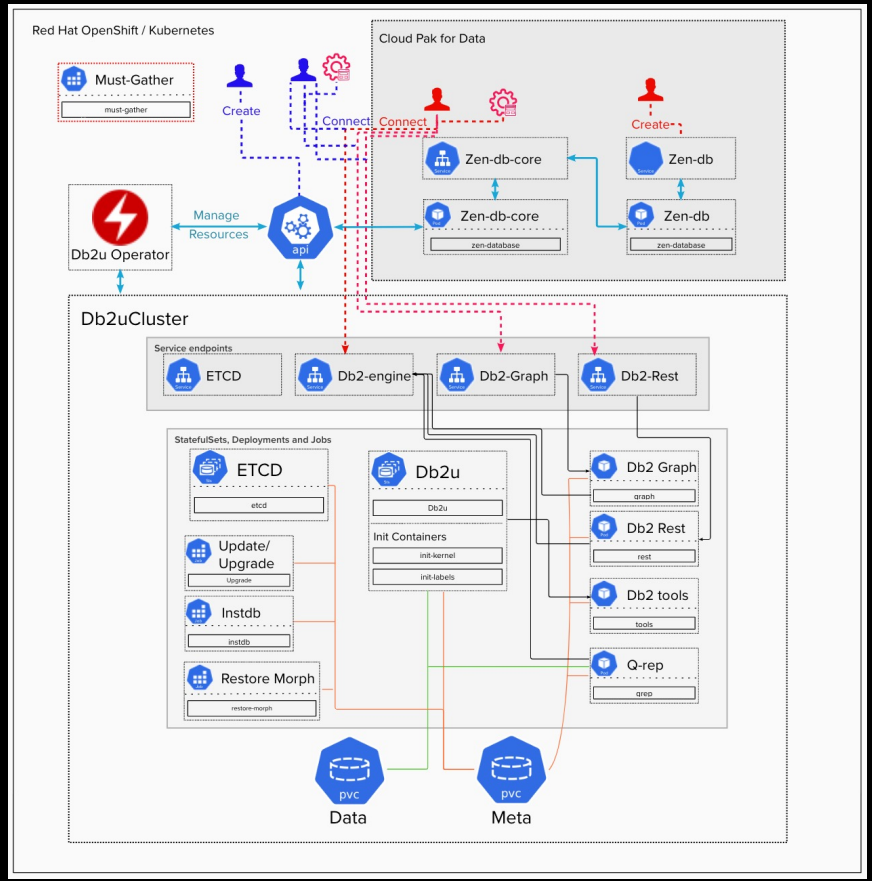
Db2U Architecture

- Db2U Architecture
  - Overview
  - Kubernetes Resource Model
  - Storage

Db2U Architecture: Overview

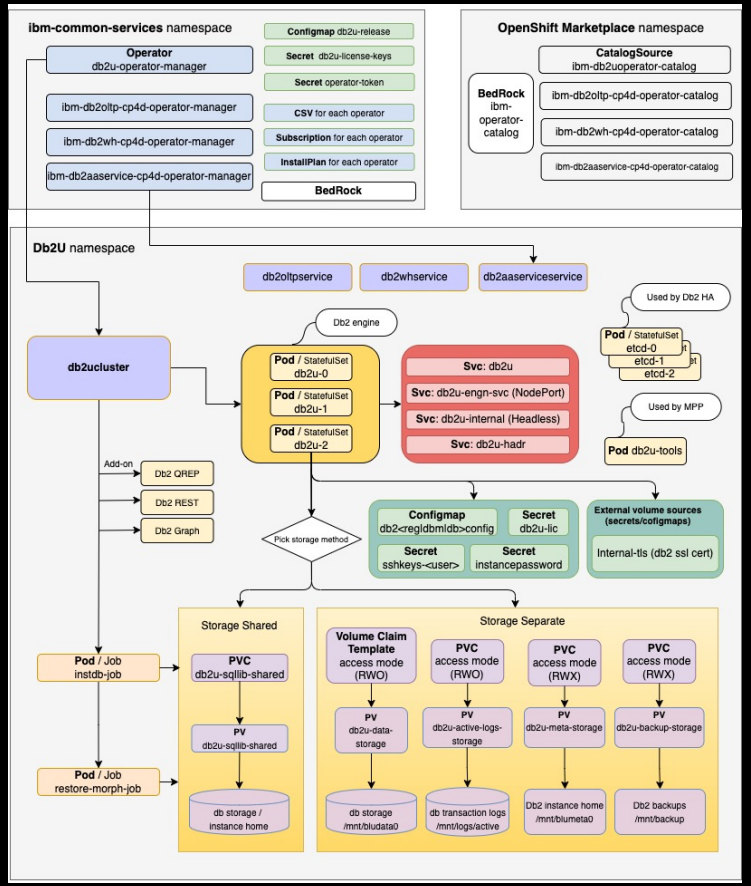
Underlying Kubernetes resource model:

- Db2 Engine Pod lifecycle managed using a `StatefulSet` resource, since Db2 is a stateful application.
- Onetime tasks managed via `Job` resource
- In-pod HA to recover Db2 failures, avoiding a pod lifecycle event. This built-in HA leverages ETCD for state information
- Lifecycle of (stateless) Add-Ons (REST, Graph, Qrep, etc.) managed via `Deployment` resource



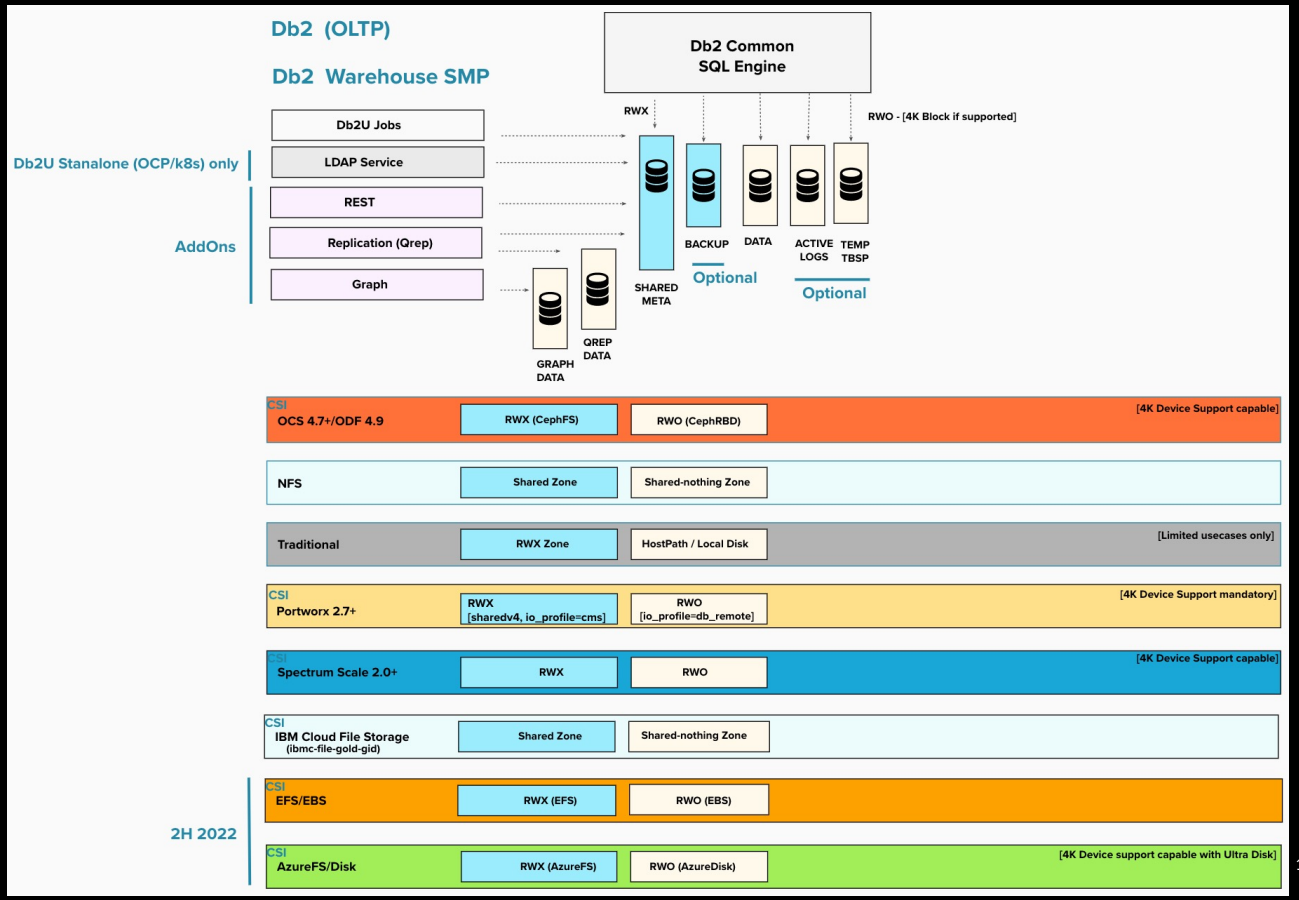
Db2U Architecture: Kubernetes Resource Model

- All Db2 configuration settings (Registry/DBM/DB cfg) injected via CR are transposed into `ConfigMaps` and mounted into Db2U PODs.
- Persistent Volume attachment:
  - Shared Storage volume (Db2 instance home/other shared metadata) via `PersistentVolumeCliaim` (PVC) with `ReadWriteMany` (RWX) access mode.
  - Data Storage (Db2 database paths) via `VolumeClaimTemplates` with `ReadWriteOnce` (RWO) access mode in Db2U `StatefulSet`



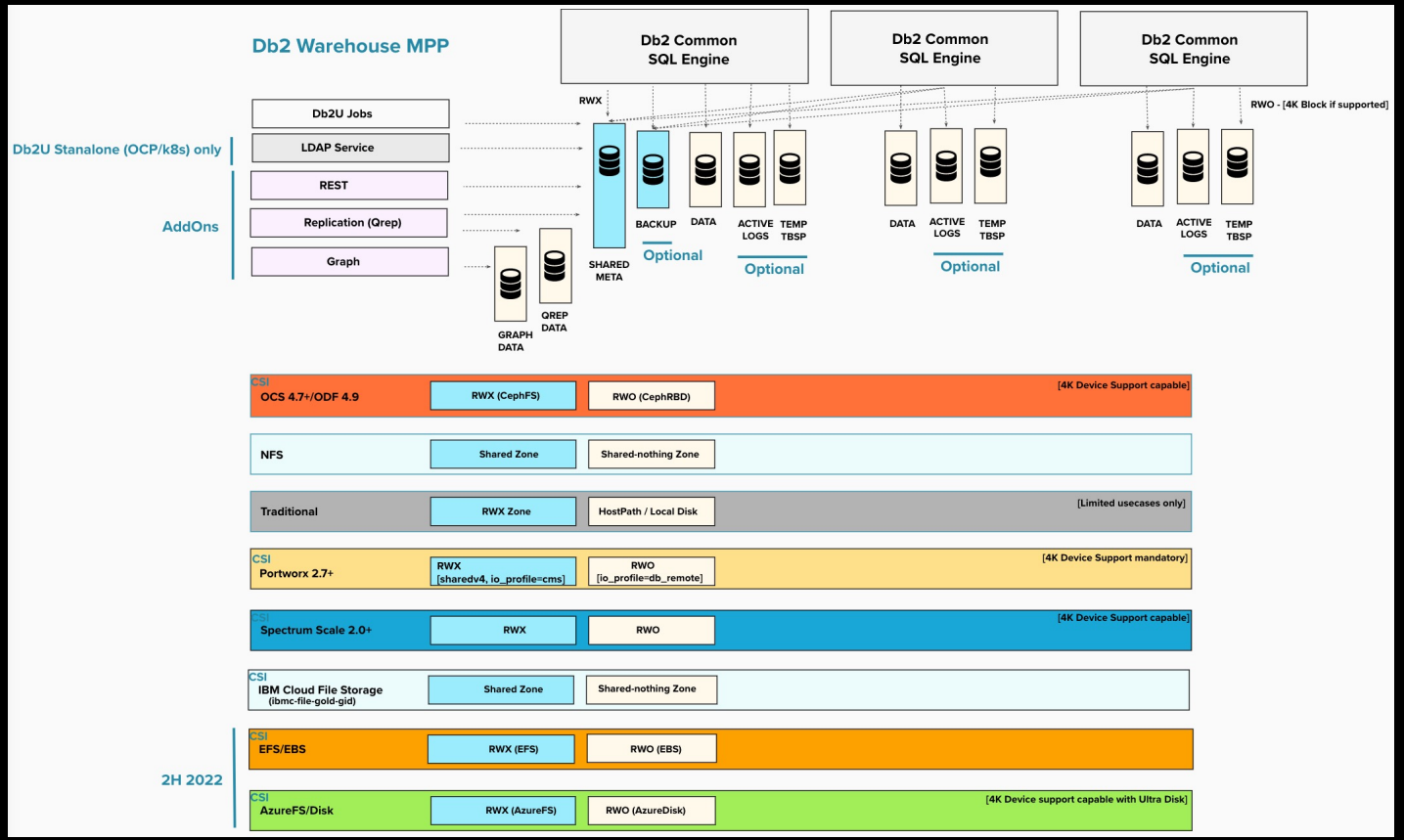
Db2U Architecture: Storage (Db2 OLTP/ WH SMP)

- Container Native Storage Options
  - OpenShift Container Storage (OCS/ODF) 4.7+, 4.9
  - Portworx 2.7+
  - IBM Spectrum Scale CSI 2.0+
  - Public Cloud Provider Native Storage (EKS/AKS)
  - NFS / Host Path (IBM Cloud, Dell EMC Isilon, Local...)
  - NAS (Dell EMC Isilon, NetApp Trident CSI)



Db2U Architecture: Storage (Db2 Warehouse MPP)

- Container Native Storage Options
  - OpenShift Container Storage (OCS/ODF) 4.7+, 4.9
  - Portworx 2.7+
  - IBM Spectrum Scale CSI 2.0+
  - Public Cloud Provider Native Storage (EKS/AKS)
  - NFS / Host Path (IBM Cloud, Dell EMC Isilon, Local...)
  - NAS (Dell EMC Isilon, NetApp Trident CSI)



Db2U Security Posture

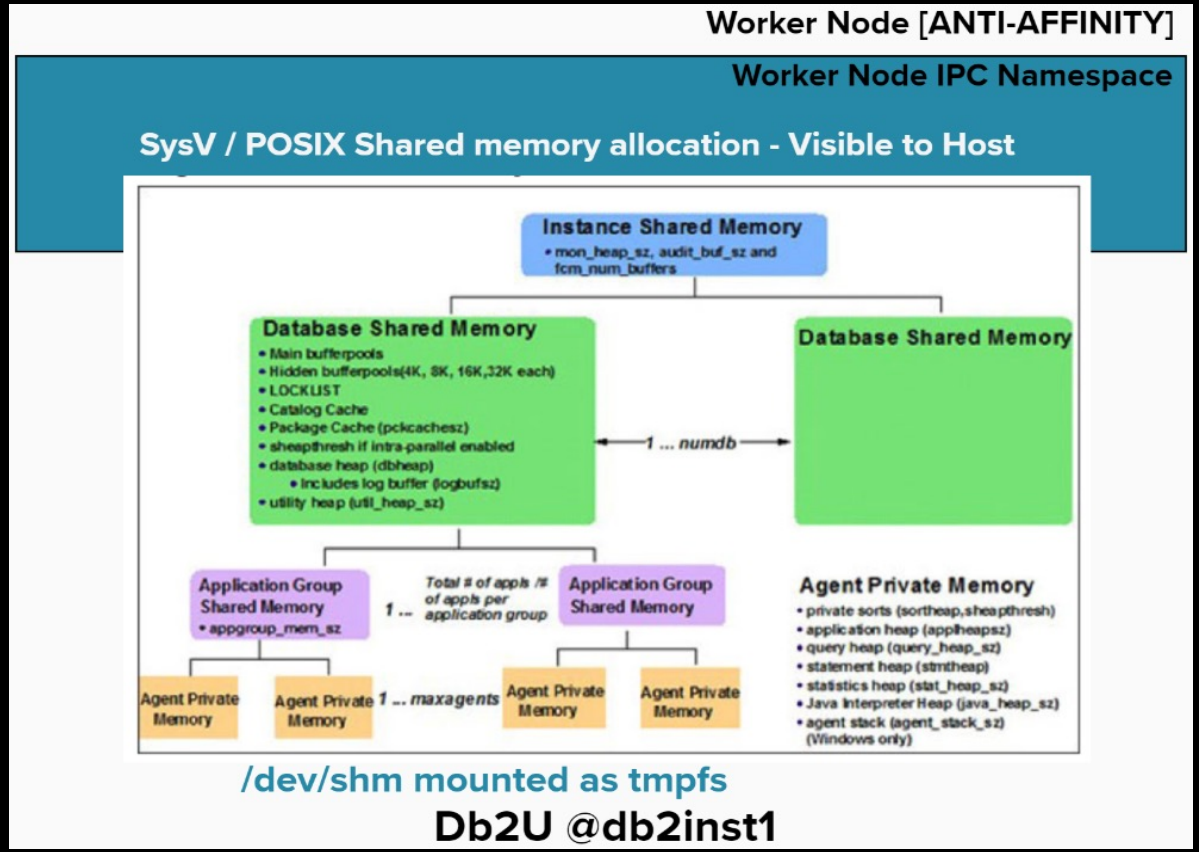
- Db2U Security Posture
  - Non-Privileged
  - Service Account and Security Context Constraint / Pod Security Policy

Db2U Non-Privileged: Overview

- To satisfy security compliance requirements in cloud native environments, Db2U container(s) are run in *non-privileged* mode with a *limited set of capabilities*. This was achieved by the following combination of methodologies:
  - Relaxation of Linux namespace isolation for shared memory inter-process communication (IPC)
  - A targeted minimal set of Linux System Capabilities
  - Cloud-native approach to Linux IPC kernel parameter tuning

Db2U Non-Privileged: IPC Shared Memory Access

- **Db2 Warehouse MPP**: SysV IPC shared memory allocations by Db2 database engine supported via Sharing IPC Namespace with the Host
- **Db2 OLTP/Warehouse SMP**: POSIX IPC shared memory allocations isolated using a private shm device



Db2U Non-Privileged: Linux System Capabilities

- A targeted minimal set of Linux System Capabilities
  - **SYS_RESOURCE**: Db2 engine needs this sys-capability to throttle resource limits
  - **IPC_OWNER**: Db2 engine's dynamic IPC tuning requires bypassing permission checks for operations on IPC objects
  - **SYS_NICE**: Db2 workload management or multi-processing activities in the engine relies on ability to manage process thread prioritization

Db2U Non-Privileged: IPC Kernel Parameter Tuning

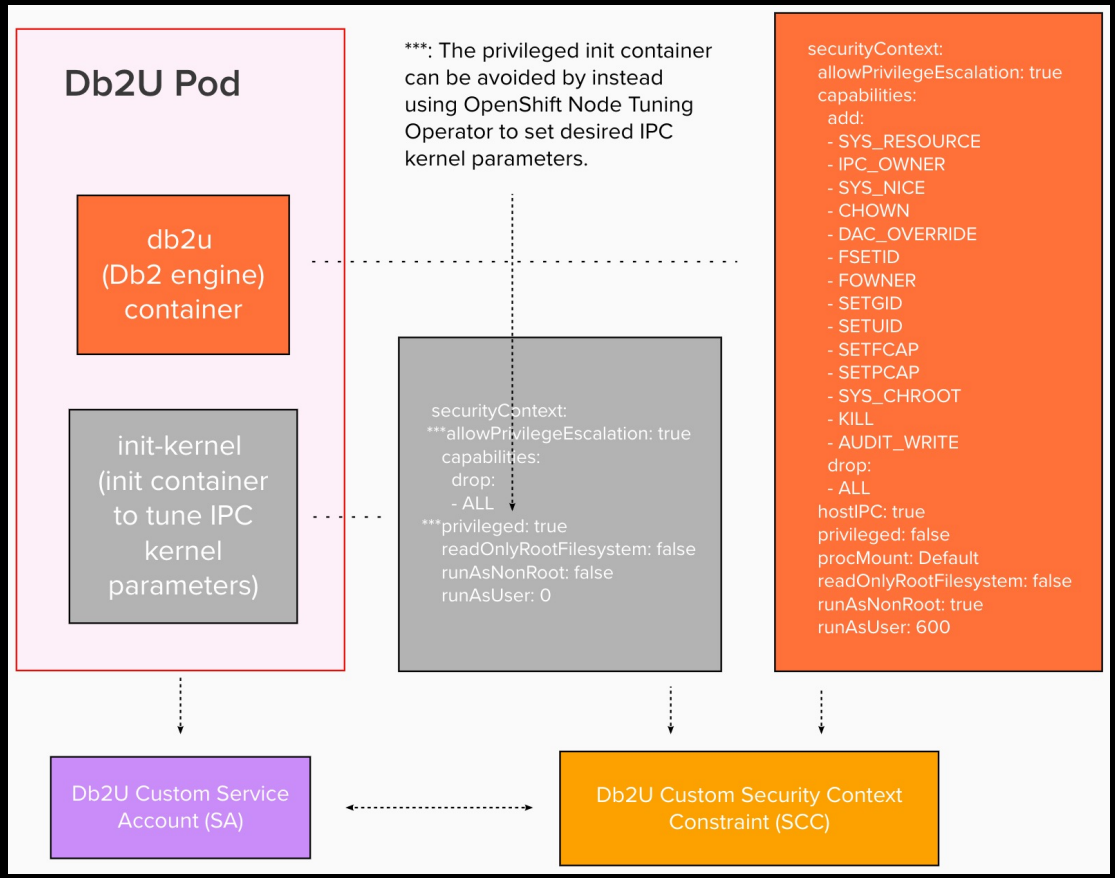
- A container itself cannot modify the upper bounds of IPC kernel parameters without write access to `/proc` and `/proc/sys`.
- Therefore, updating Linux IPC Kernel parameters, is a prerequisite to deploying a Db2 database container in non-privileged mode
- There are three options to tune IPC kernel parameters to meet Db2 engine requirements:
  - **All: Use a privileged, run-as-root *init container*** to mount host proc filesystems with write-access, and online-update IPC proc files. The Db2 container in the same POD inherits via default read-only proc fs mounting
  - **Db2 OLTP/Warehouse SMP**: Inject sysctls into pod spec. Requires *unsafe sysctl* support to be enabled on the cluster
  - **Warehouse MPP**: Apply them at the worker node-level by leveraging OpenShift *Node Tuning Operator* to apply a tuned profile for sysctls

Db2U Non-Privileged: Run as Non-Root User

- **Avoid running db2u (engine) container ENTRYPOINT under UID 0**
- Run under user `db2uadm` (UID 600)
- Define `sudo` rules for that user, such that initial setup requirements, such as creating directories, changing permissions/mod-bits, etc. can be met
- Injected into the pod/container spec via:
  - `runAsNonRoot: true`
  - `runAsUser: 600`

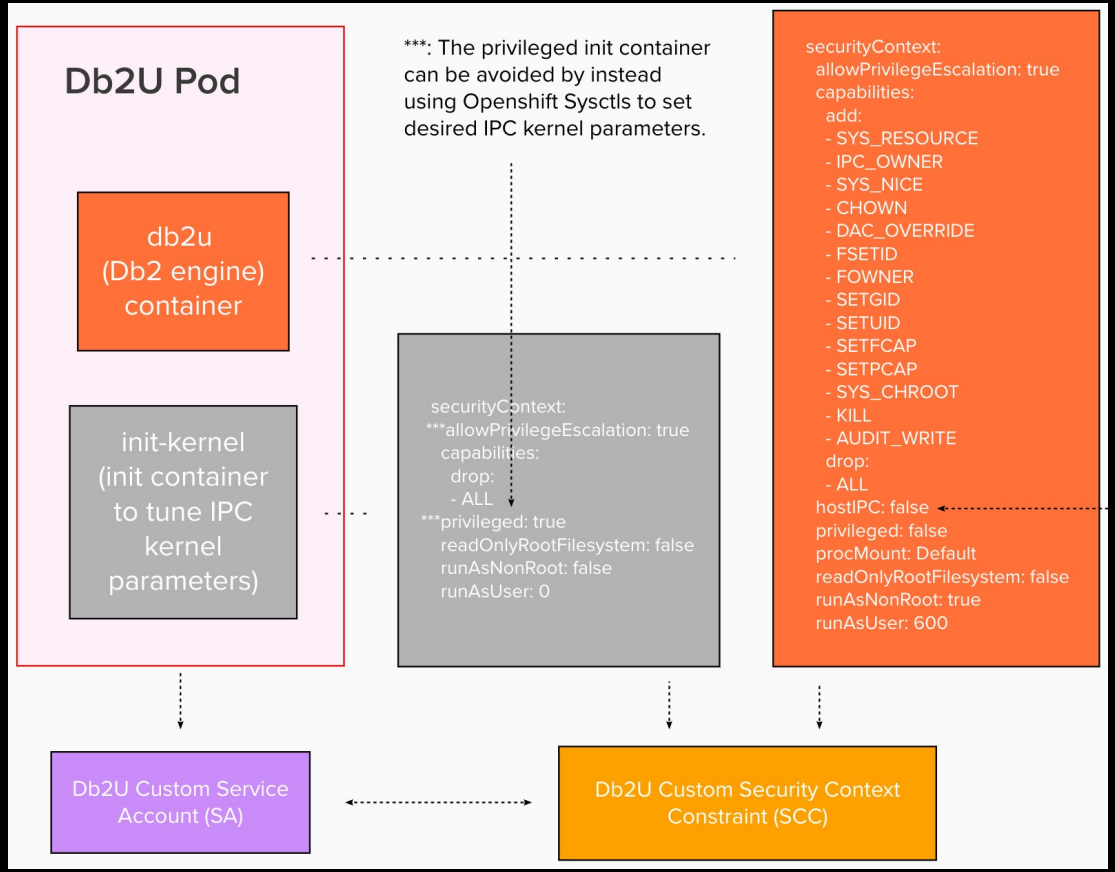
Db2U Security: SA and SCC/PSP (Warehouse MPP)

- Each instance of Db2U will have a unique *Service Account* (SA) and a custom *Security Context Constraint* (OpenShift) or Pod Security Policy (Kubernetes) to go along with it. This SCC/PSP is generated dynamically based on the minimum set of k8s access that is required by Db2U pods/containers.
- User provided SA is supported. However, in that route Db2U operator will not dynamically generate SCC/PSP, Roles and do Role-bindings, etc. Hence, all that must be done by the cluster admin prior to deploying Db2U.



Db2U Security: SA and SCC/PSP (OLTP / Warehouse SMP)

- Single node Db2 configurations (OLTP / Warehouse SMP)
  - Can run without enabling `HostIPC` since there is no inter-partition data sharing over IPC
  - Do IPC Kernel Parameter tuning by setting sysctls at pod level. However , unsafe sysctl support must be enabled on the cluster for this

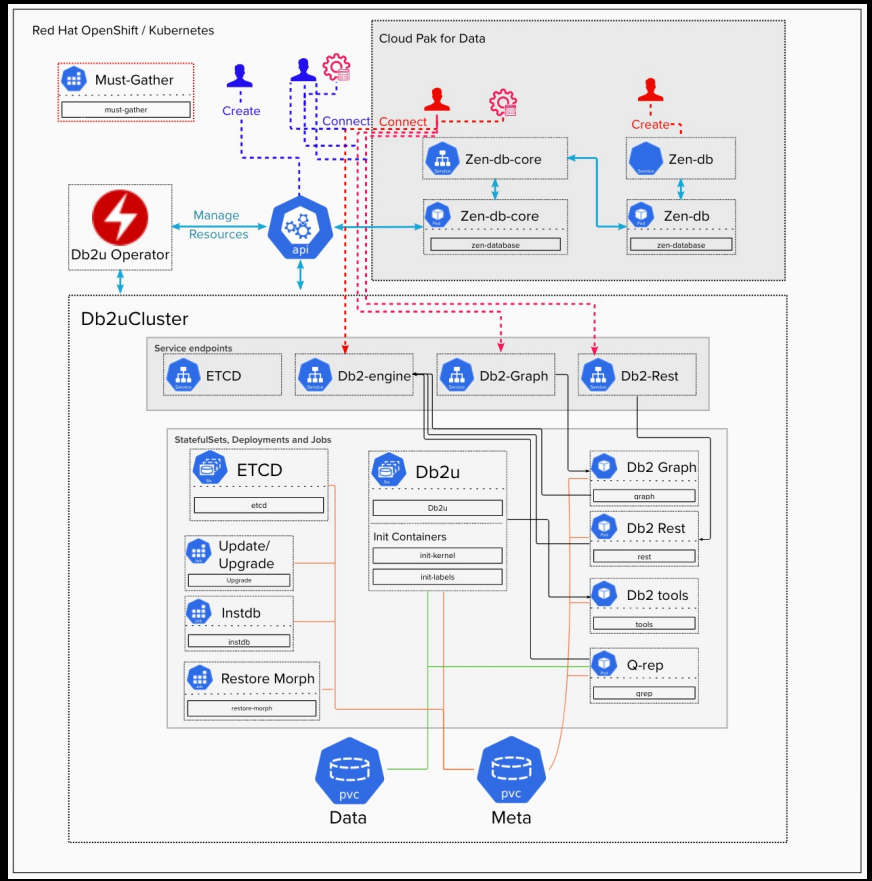


Db2 Operator: 2022/23 Roadmap

- A Survey of Current Capabilities and Future Direction
  - Overview
  - Limitations in Current Implementation
  - Next Gen – Core Capabilities
  - Next Gen – Cloud-native Backup and Restore
  - Next Gen – Cloud-native Audit Facility
  - Next Gen – Cloud-native Log Streaming
  - Next Gen – Public Cloud Provider Alignment

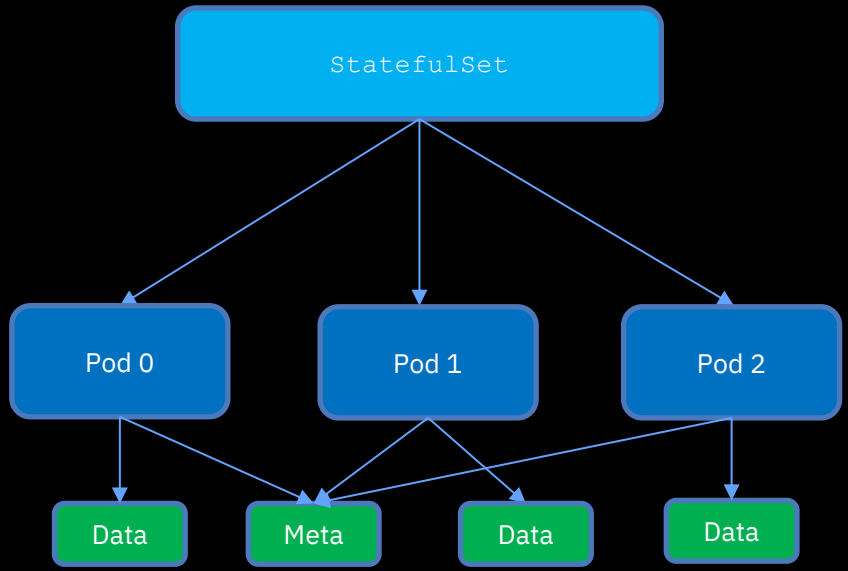
Db2U Current vs Next Generation: Overview

- **Current**
  - Db2 engine pods lifecycle is managed via a Kubernetes `StatefulSet` Object
    - All Pods are rendered from a single Pod Template spec
- **Next Gen** (Q3 2022)
  - Db2 engine pods lifecycle is managed via a NEW Kubernetes Custom Resource `Db2uEngine` Object
    - Spec of each Pod defined independently



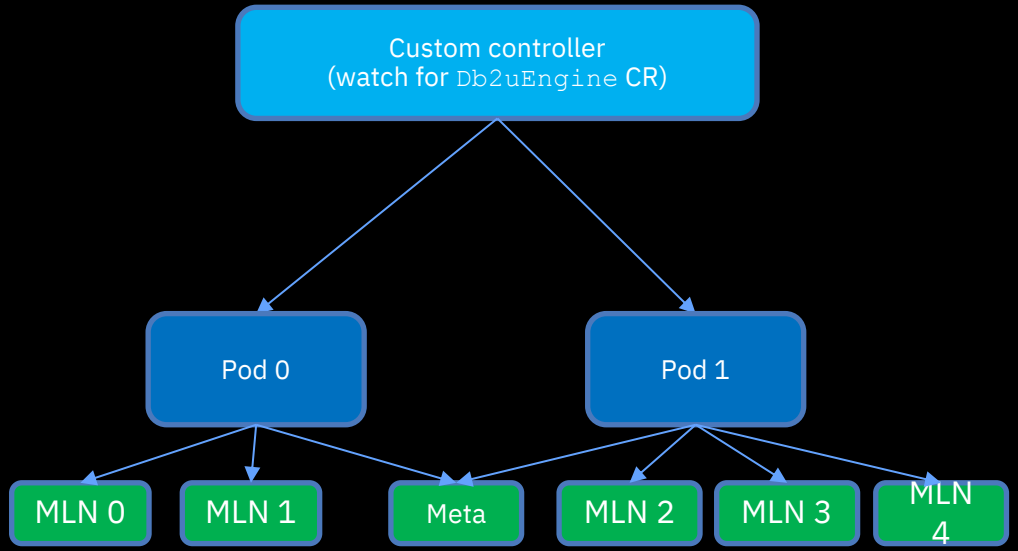
Db2U Current: Key Limitations

- Limitations in Db2 Warehouse MPP Deployments
  - Database storage paths for all Db2 partitions (MLNs) on a given pod, are mapped to a single Kubernetes volume
  - Homogeneity
    - In number of MLNs per pod
    - In Pod resource limits (all Pods must use same resource limits)



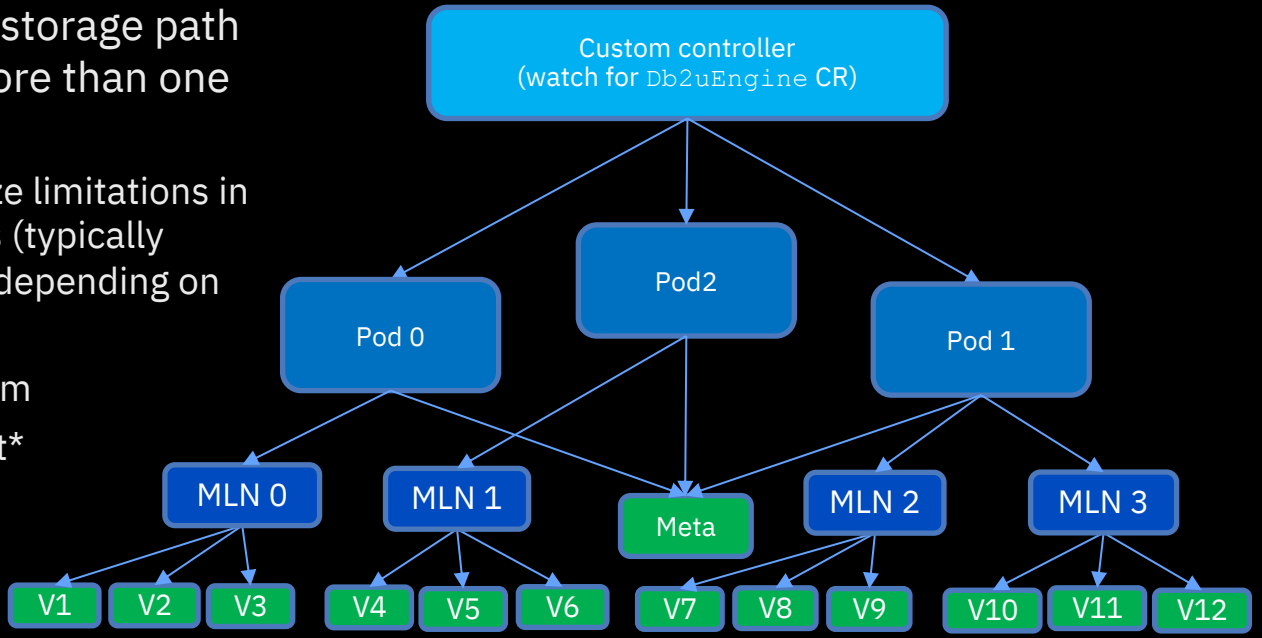
Db2U Next Generation: Core Capabilities

- What's new
  - One-to-One mapping between each database MLN storage path and Kubernetes volumes
    - Better alignment with MPP shared-nothing architecture
    - Leads to better support for horizontal scaling
  - Heterogenous configurations



Db2U Next Generation: Core Capabilities (Contd.)

- What's new
  - Each Database MLN storage path can be backed by more than one volume (1H 2023)
    - Mitigate volume size limitations in cloud deployments (typically 16TB to 64TB cap depending on the vendor)
    - Better IO parallelism
    - Lower storage cost*

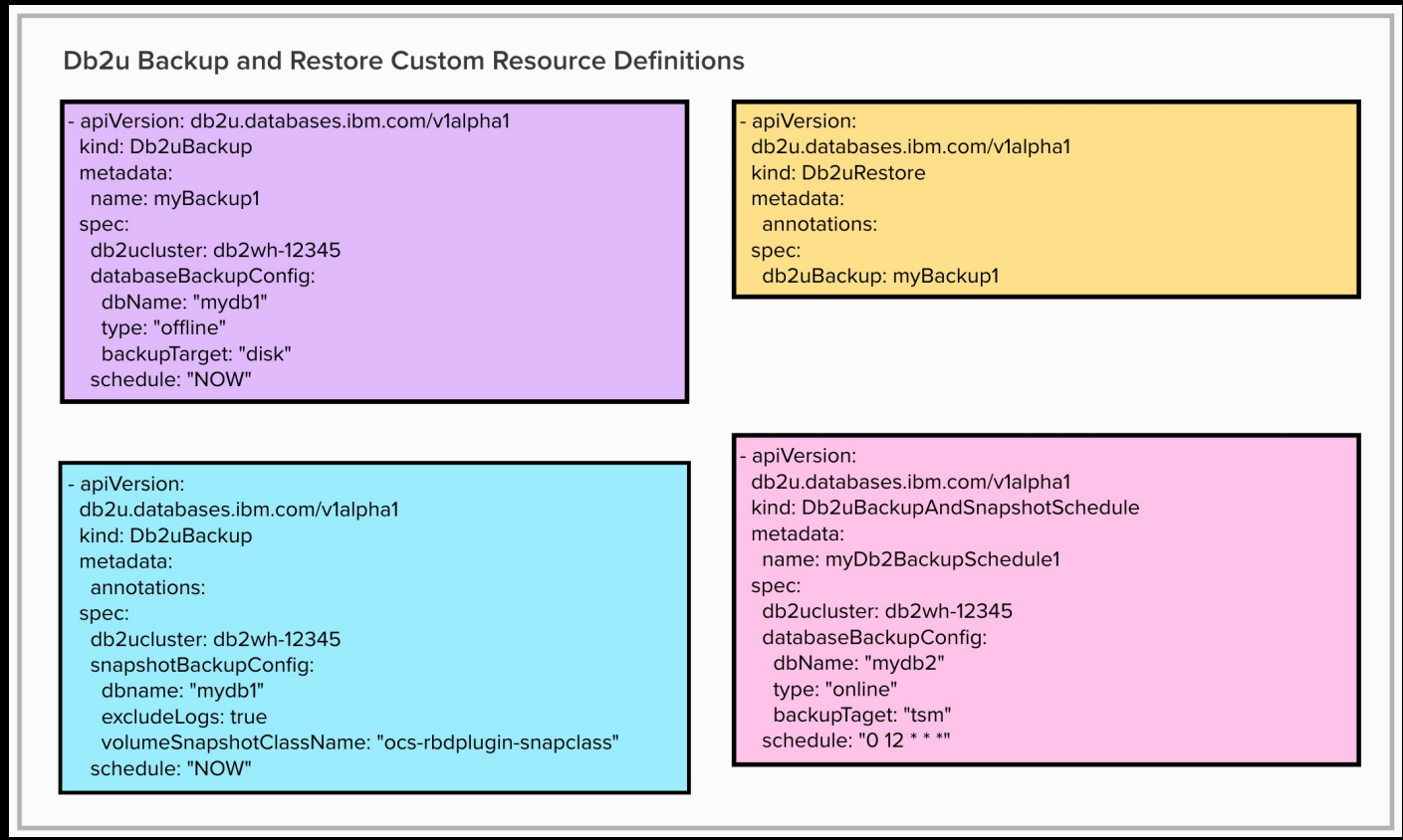


Db2U Next Generation: Demo

- Demo
  - Core Capabilities
  - Key Differences between Current vs Next Generation

Next Gen: A cloud-native Backup & Restore Experience

[2H 2022] A Kubernetes controller driven approach to managing Db2 Backup/Restore, and Snapshot capabilities via Custom Resource Kind `Db2uBackup` and `Db2uRestore`

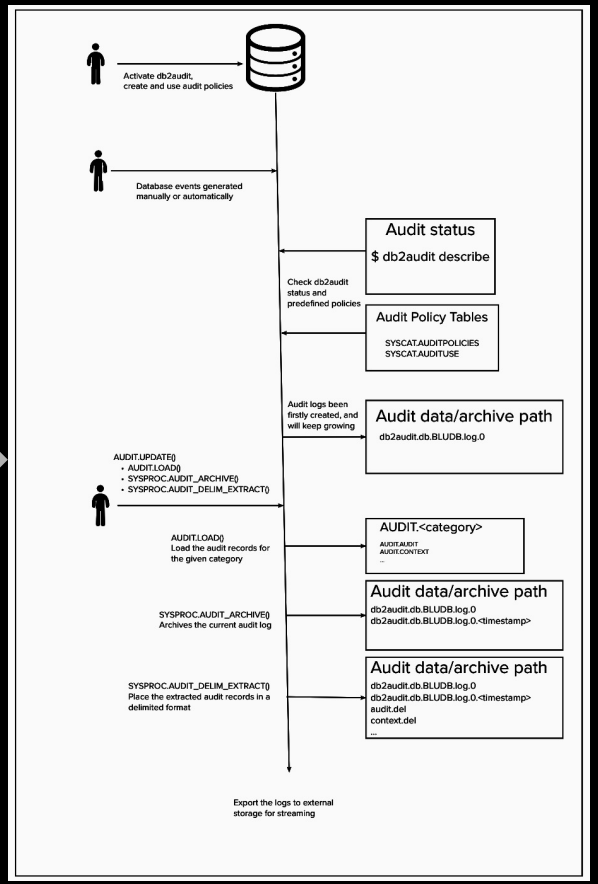


Next Gen: A Cloud-native Db2 Audit Facility

[2H 2022] A Kubernetes controller driven approach to managing Db2 Audit facility via Custom Resource Kind `Db2uAudit`

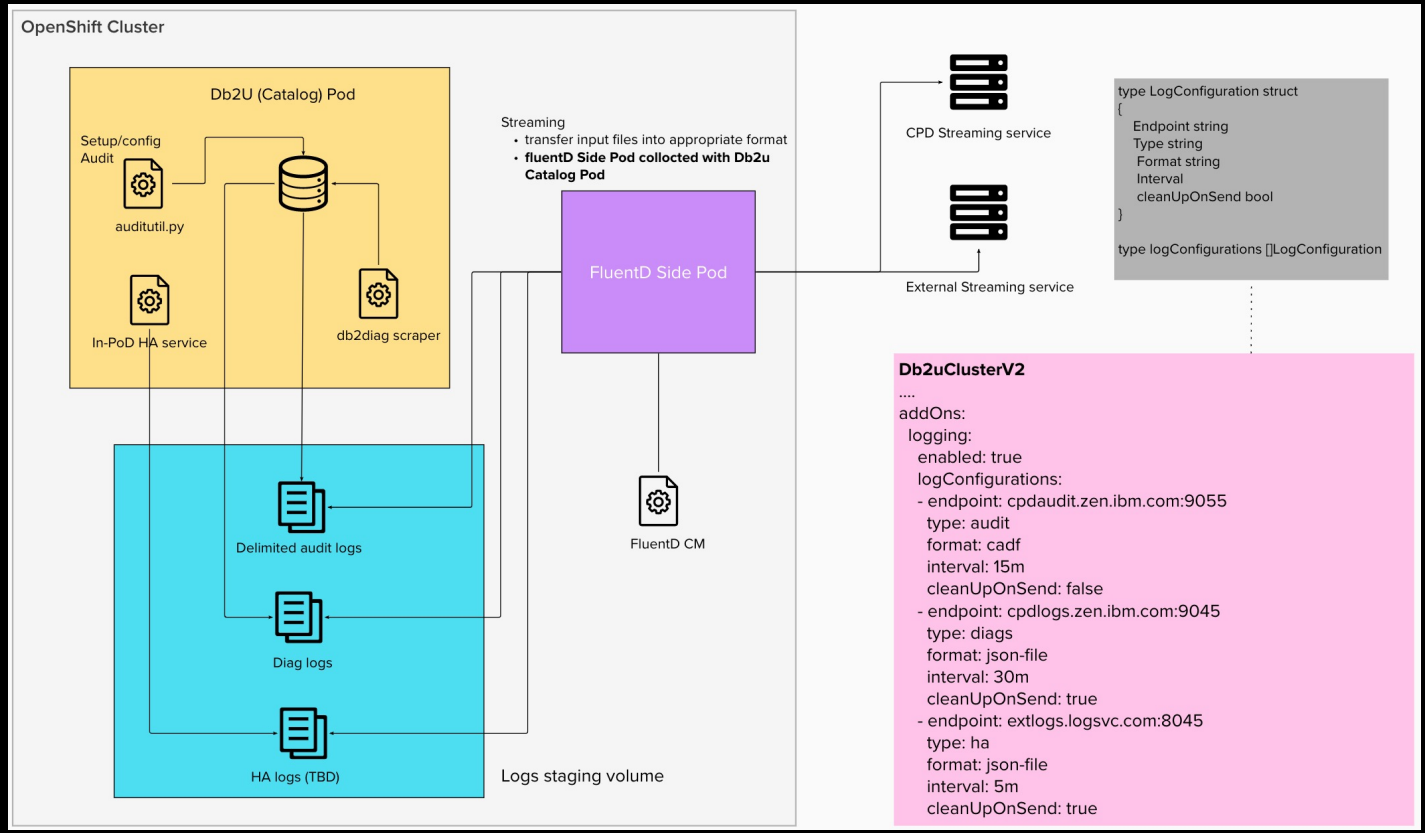
Db2uAudit

```
- apiVersion: db2u.databases.ibm.com/v1alpha1
  kind: Db2uAudit
  metadata:
    name: myAudit1
  spec:
    db2ucluster: db2wh-12345
    auditConfigurations:
    - dbName: BLUDB
      interval: 15m
      applyDefaultPolicy: false
      customPolicy:
        name: myCustAuditPol1
        category: audit
        status: failure
        errorType: audit
        recordObjects:
        - type: schema
          objects:
          - myschema1
          - myschema2
        - type: table
          objects:
          - myschema3.mytab1
```



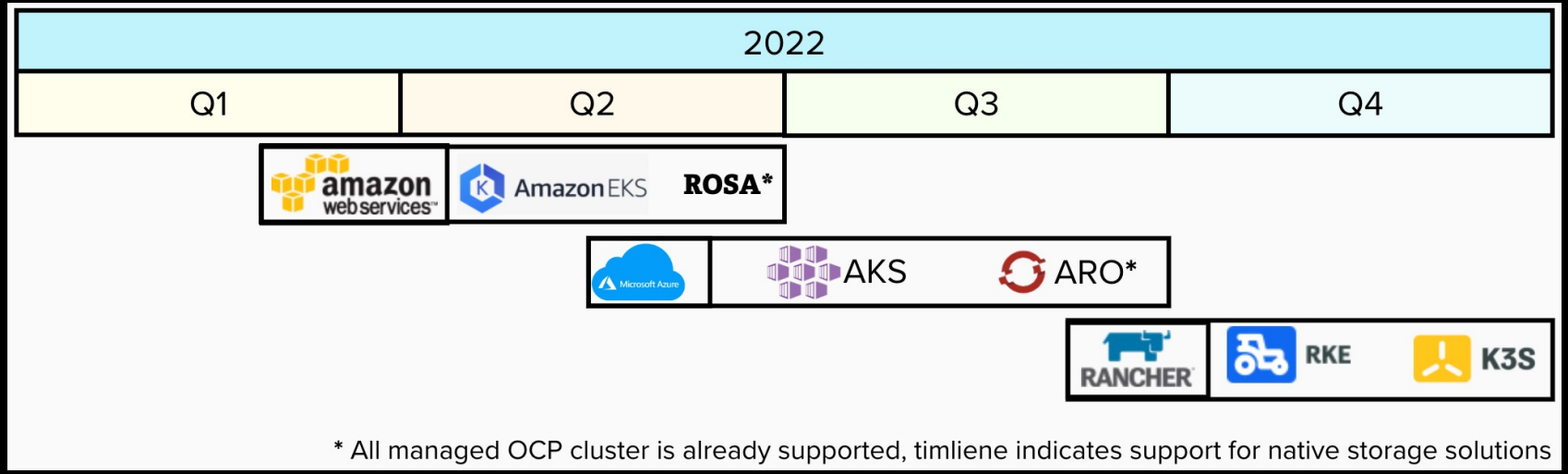
Next Gen: A Cloud-native Log Streaming

[2H 2022 / 1H 2023] Support log streaming (audit, diaglogs, HA logs, etc.) to Cloud Pak for Data or to an external logging service using a *Side Pod*, and enabled via Db2U Custom Resource `addOns` mechanics



Next Gen: Perfectly aligned for Public Cloud

- **Amazon**
  - Elastic Kubernetes Service (EKS) with EFS (Shared RWX) and EBS (Per-MLN RWO) volumes
  - Red Hat OpenShift Service on AWS (ROSA) with OCS/ODF
- **Azure**
  - Azure Kubernetes Service (AKS)
  - Azure Red Hat OpenShift (ARO)
- **Google Cloud** – Google Kubernetes Engine (GKE)
- **Rancher** – Rancher Kubernetes Engine (RKE)



* All managed OCP cluster is already supported, timliene indicates support for native storage solutions

Db2 Operator Next Generation: Summary

- Summary of New features
  - Support large-scale data warehouse and transactional databases
  - Better support for horizontal scaling
  - Cloud-native backup, restore and snapshot capabilities
  - More Day 2 operations will be transformed to provide a cloud-native user experience – I.E., simply interact with Kubernetes API rather than directly with Db2
  - Better aligned with Public Cloud Provider Infrastructure

Db2 on CPD, OpenShift, Kubernetes - “Elevated Value”

The Db2 Operator

Operator Extended VALUE

Time-Saving

Cost-Saving

Modernize

Extended HA

Complete Lifecycle

Managed Lifecycle
-- Db2 Special build
-- Upgradeability
--- **AutoUpgrade Technology** [~20']
OCP/Kubernetes alignment

Modern

Infrastructure Consolidation
-- Better HW/VM utilization
Deploy in the Cloud [AWS/Azure/...]
API-driven Management
Automation

Fast Deployments

Aligned with DevOps productivity requirements
-- Ready in minutes [10-20']

Complete Ecosystem

Add-ons pluggability
-- REST/GRAPH
-- Data Replication (Qrep)
-- DMC
-- ***New*** Containers In progress

Secure

Cloud Pak 2.0 Certification
-- Security and Privacy By design
-- On-going assessement
SSL built-in
OpenShift layered strategy

Highly Available

HADR Intra/Inter cluster
4-tiers Resiliency strategy
-- Self Healing [Process/Pod]
-- Standby Node
-- Standby Db2 (HADR)

Modern Storage

Support for Container-Native Storage (CNS)
-- OCS/ODF, Portworx, Spectrum Scale CSI, IBM Cloud file storage
-- Resiliency with Replication

Hybrid Cloud Ready

Cloud Like Experience
-- "As a Service"
-- OpenShift Catalog