Dataset Lifecycle Framework

Next Generation Systems Team
IBM Research – Europe

Freeing the data scientist from the anxiety of management and performance of accessing data in a multi-cloud environment

Scenario

Data Scientist

Search

Catalog

Publish datasets

Data Lake

Manage

Data Provider

Pain points
- *Locate datasets*
- *Manage credentials*
- *Configure Jobs*
- *Reduce wait*

Launch Job

kubernetes OPENSHIFT

Access data

Pain points
- *Configure access*
- *Enforce governance*

Produce

Insights

Cloud resources

Datasets Lifecycle Framework - Objectives

*Make **Dataset** a first-class citizen in Kubernetes Openshift environments*

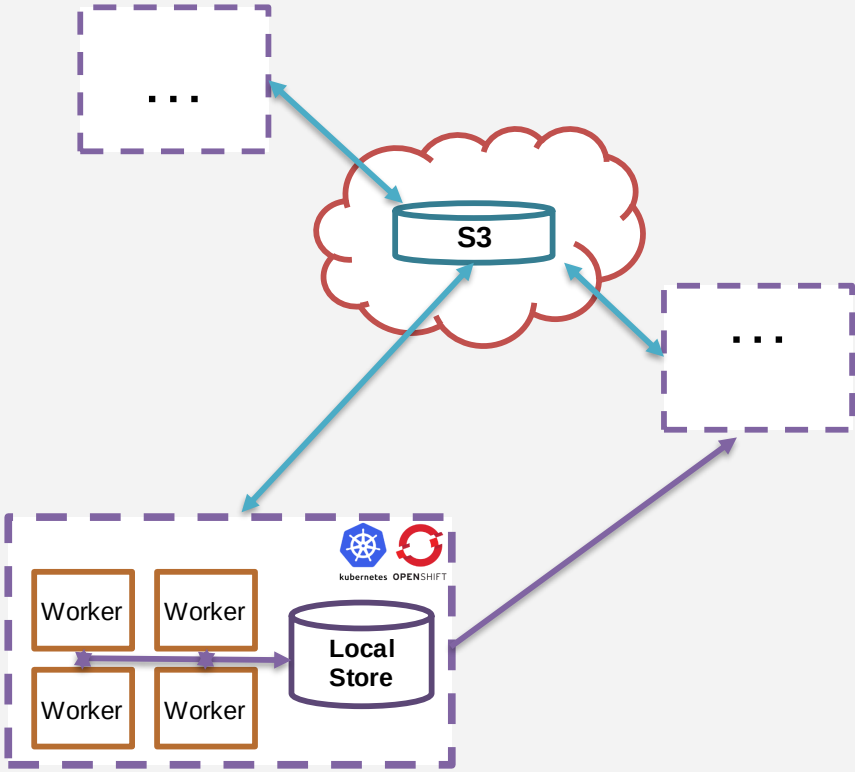
- Implement as **CRD (Custom Resource Definition)** and support **S3, NFS** data sources
- Ability to lookup Datasets from remote catalogs, such as **Hive Metastore**
- Require **minimal changes** to the end-user workflows in order to work with datasets in their workloads

*Transparent **Data Caching***

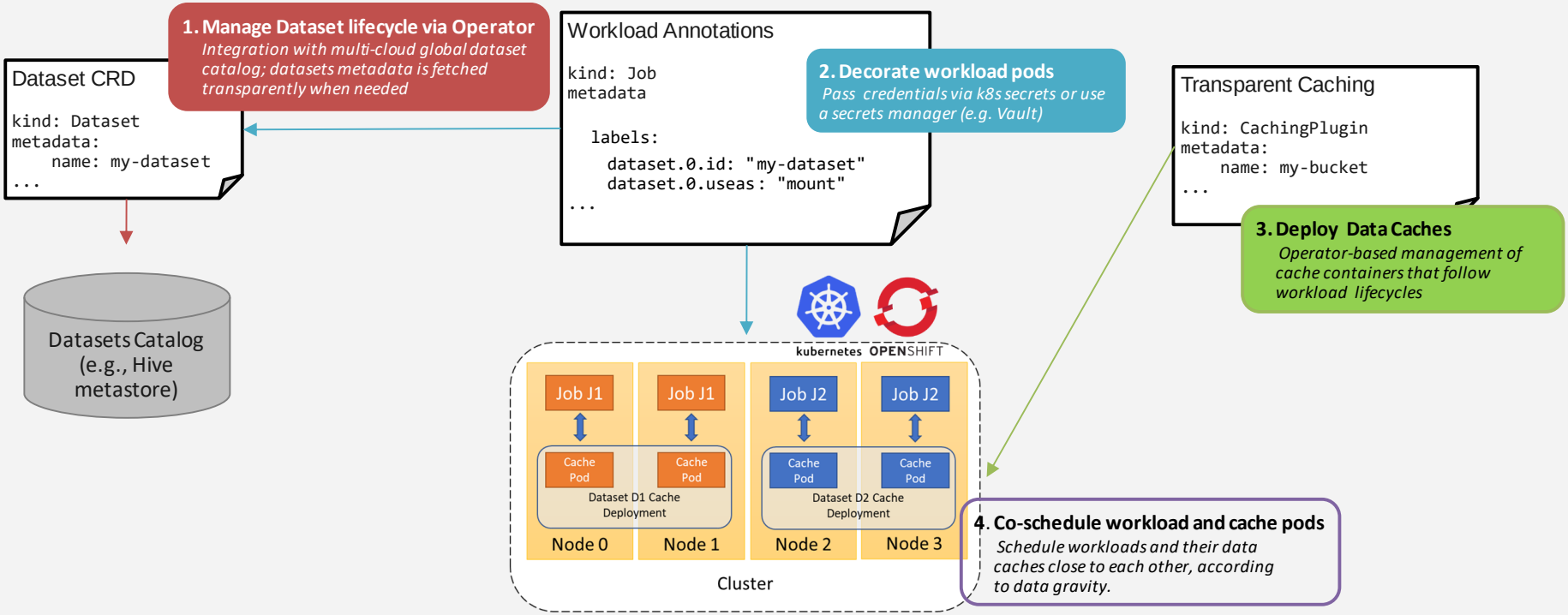
- **Pluggable interface** for caching frameworks to implement
- **On the fly deployment** of cache pods based on dataset usage patterns

***Optimized Workload Scheduling** to bring Pods closer to cached Data*

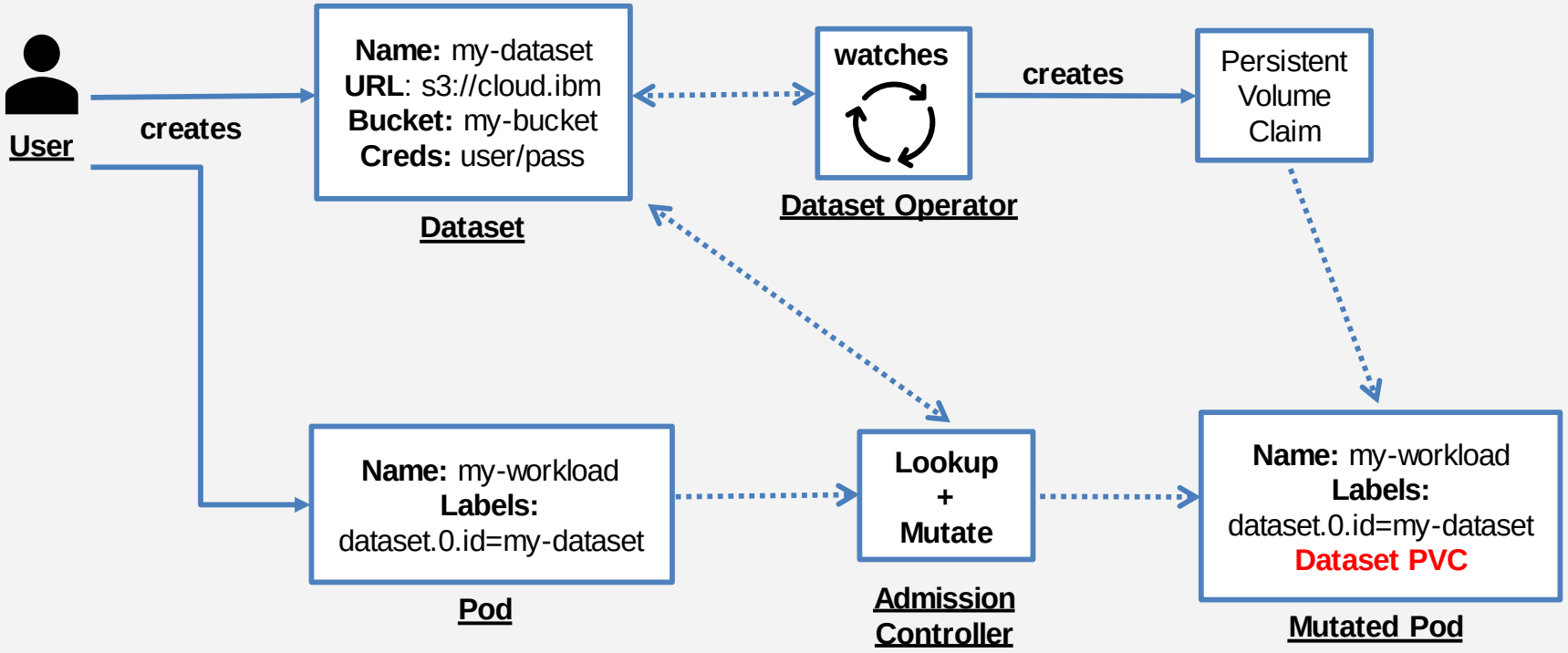
*Integration with popular **DL and ML Frameworks***



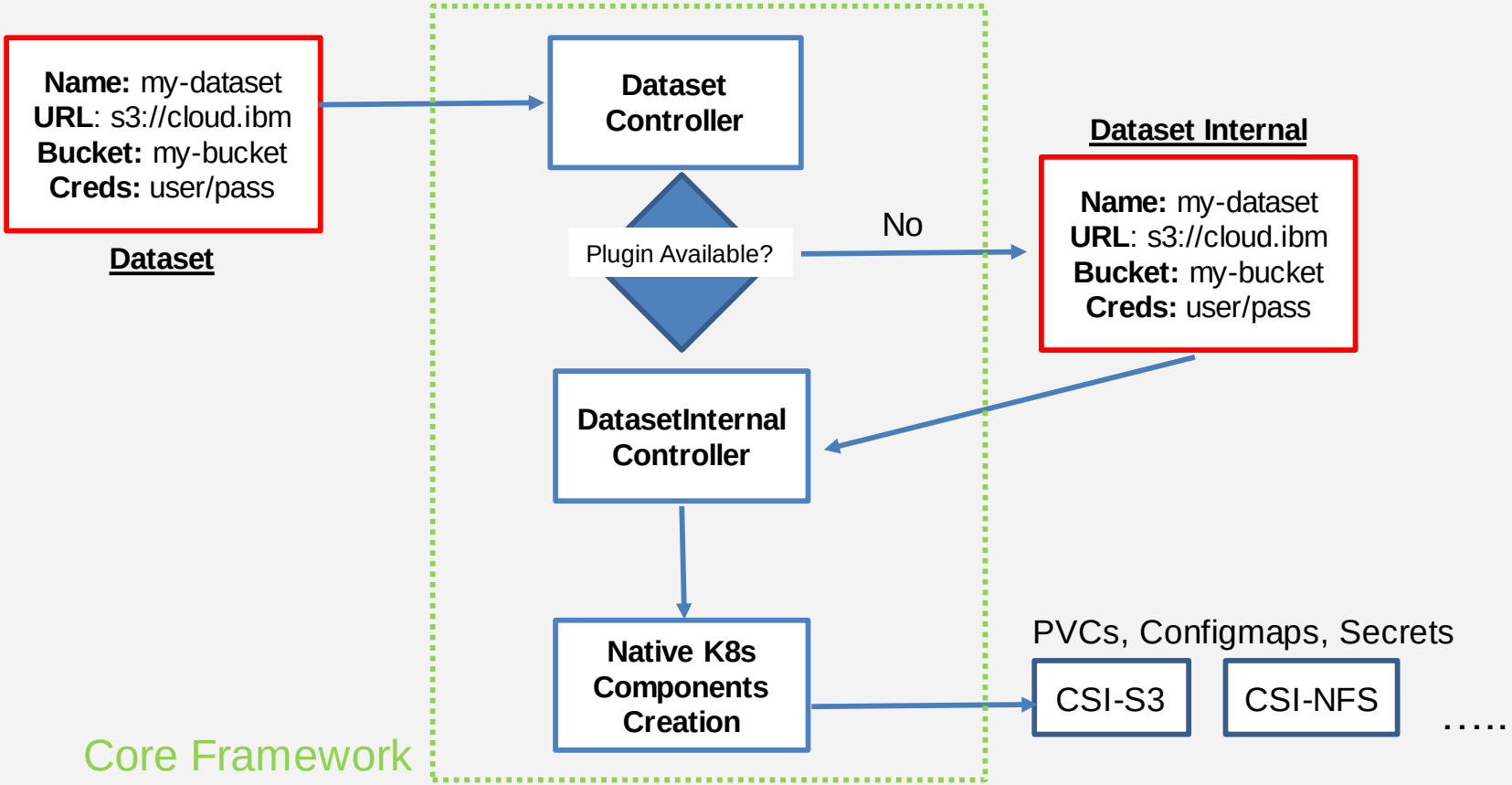
Datasets Lifecycle Framework – Overall Approach



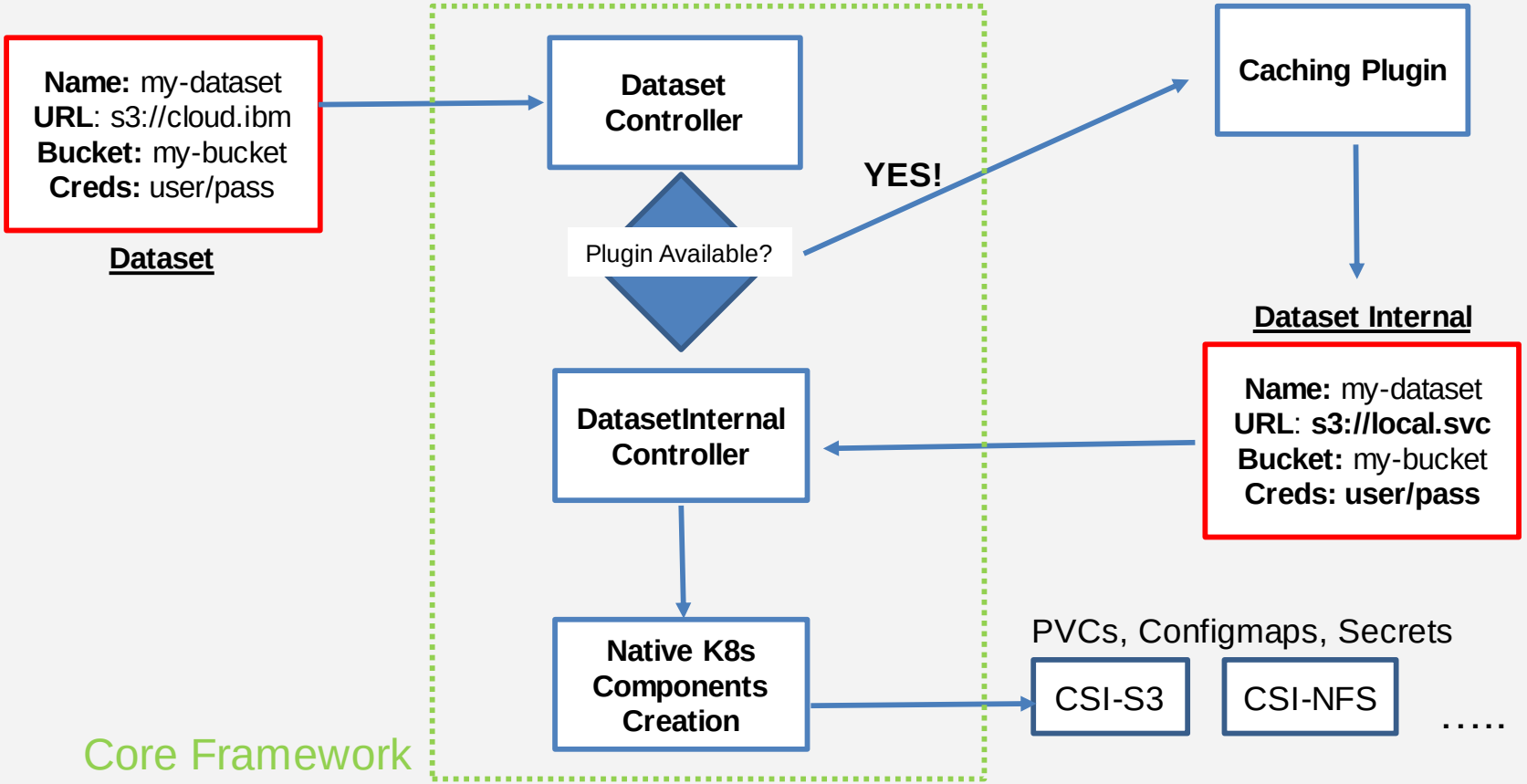
Datasets Lifecycle Framework - Components



Datasets Lifecycle Framework – Transparent Caching



Datasets Lifecycle Framework – Transparent Caching



Datasets Lifecycle Framework - Integrations

(1) Hive Metastore

- Ability to lookup dataset information and metadata from Hive metastore catalog

(2) Noobaa (https://www.noobaa.io/)

- S3 buckets created by Noobaa can be transparently used as Datasets

(3) Trusted Service Identity (https://github.com/IBM/trusted-service-identity)

- Management of credentials to access datasets in Object Stores

Find us on GitHub:

https://github.com/IBM/dataset-lifecycle-framework

DEMO:

https://asciinema.org/a/276331

Thank you

Christian Pinto, christian.pinto@ibm.com
Yiannis Gkoufas, yiannisg@ie.ibm.com
Srikumar Venugopal, srikumarv@ie.ibm.com
Michael Johnston, michaelj@ie.ibm.com

BACKUP SLIDES

Datasets Lifecycle Framework - Components

(1) Dataset CRD and Operator:

- Custom Resource Definition, allows user to create Dataset objects
- Operator monitors namespace for Datasets
- Interacts with the corresponding storage plugins to enable access to data source

(2) Storage Plugins:

- If the dataset created can be mounted, then we use one of the available CSI (Container Storage Interface) implementations
- Available Implementations: S3, NFS

(3) Pods Admission Controller:

- Monitors the creation of pods which use datasets and decorates them on-the-fly with the appropriate volumes and volume mounts

(1) Dataset CRD

```yaml
apiVersion: com.ie.ibm.hpsys/v1alpha1
kind: Dataset
metadata:
  name: test
spec:
  local:
    type: "COS"
    accessKeyID: "testKeyId"
    secretAccessKey: "testKey"
    endpoint: "https://s3.eu.cloud-object-storage.appdomain.cloud"
    bucket: "test-yiannis"
    region: "" #it can be empty
```

<my-dataset.yaml>

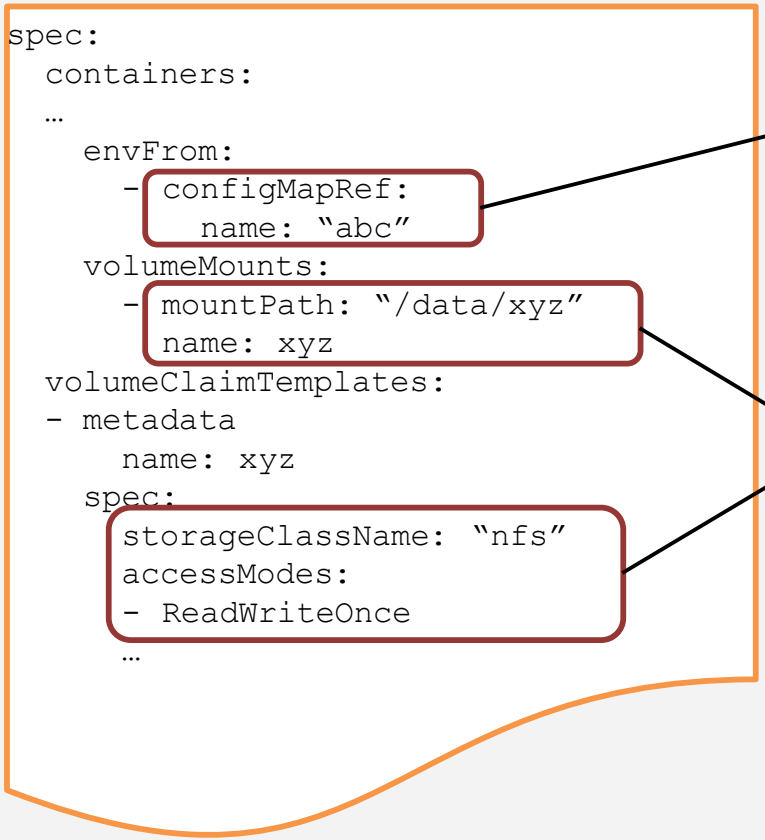
(1) How do PODs use datasets

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    dataset.0.id: "test"
    dataset.0.useas: "mount"
spec:
  containers:
    - name: nginx
      image: nginx
```

<my-pod.yaml>

- Workloads (e.g., Jobs/ Deployments) **specify the datasets that they will use as** kubernetes labels
- Custom k8s admission hooks intercept workload creation request and augment the creation process by:
  - Collecting monitoring information on the workload using the dataset
  - Passing necessary information into the Pods environments (e.g., **ConfigMap**, **secret**)
  - Optionally, creating and mounting **PVCs** for the dataset inside the containers (e.g., S3FS mounts)
  - Post creation hooks are also invoked to:
    - Call dataset lifecycle events listeners

Data Access in Kubernetes



Configure Secrets:

- Create secrets
- Refer in spec or maybe not

Configure PVC/OBC:

- Needs knowledge of storage backend
- Not portable

Configure Pod:

- Easiest step
- Maybe the app does not need mount points

Datasets Lifecycle Framework - Roadmap

Q2 2020

Object Bucket Claims using Rook

Plugin Caching example with Ceph

Initial integration with Apache Spark

Q3 2020

Object Storage Caching via Ceph + Rook

Co-location of cache and task pods for Apache Spark (data-aware scheduling)

Q4 2020

Dynamic deployment of object storage cache

Extended Kubernetes scheduler to enforce constraints while scheduling application/COS pods