Kubernetes operator for XRootD cluster

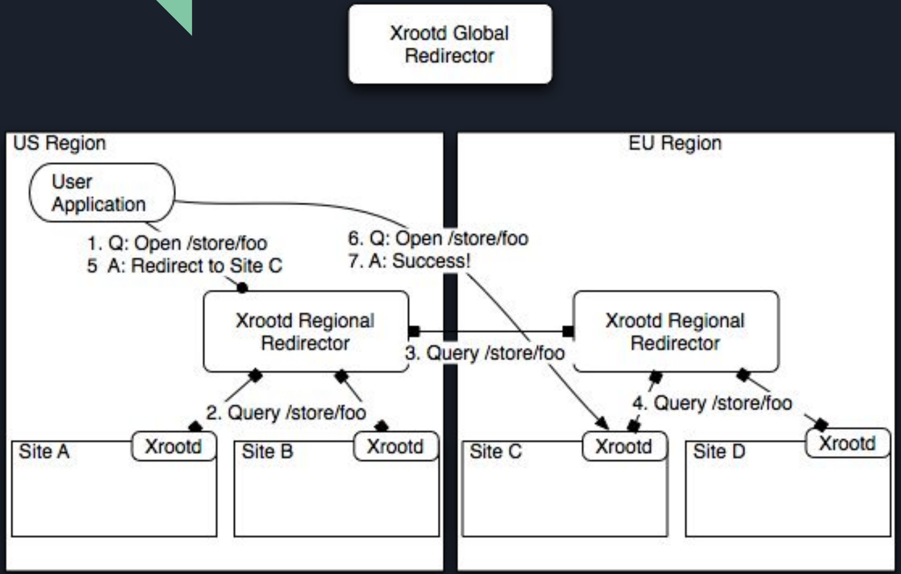
Hosted at xrootd/xrootd-k8s-operator

Report at my blog

Project Goals

- To develop a operator that:
    - eases and fully automate deployment and management of XRootD clusters
    - targeted for all clusters compliant with Kubernetes API
    - is intended for use by the XRootD community in order to scale-up worldwide XRootD clusters management
    - is easy-to-install and has seamless upgrades
    - provides deep insights to the cluster state and alerts on failure
- Write well-written documentation for the operator that:
    - describes the installation and update process
    - explains configuration options for XRootD cluster
    - describes how to extend the cluster
    - documents the contribution guidelines and development process

XRootD Protocol



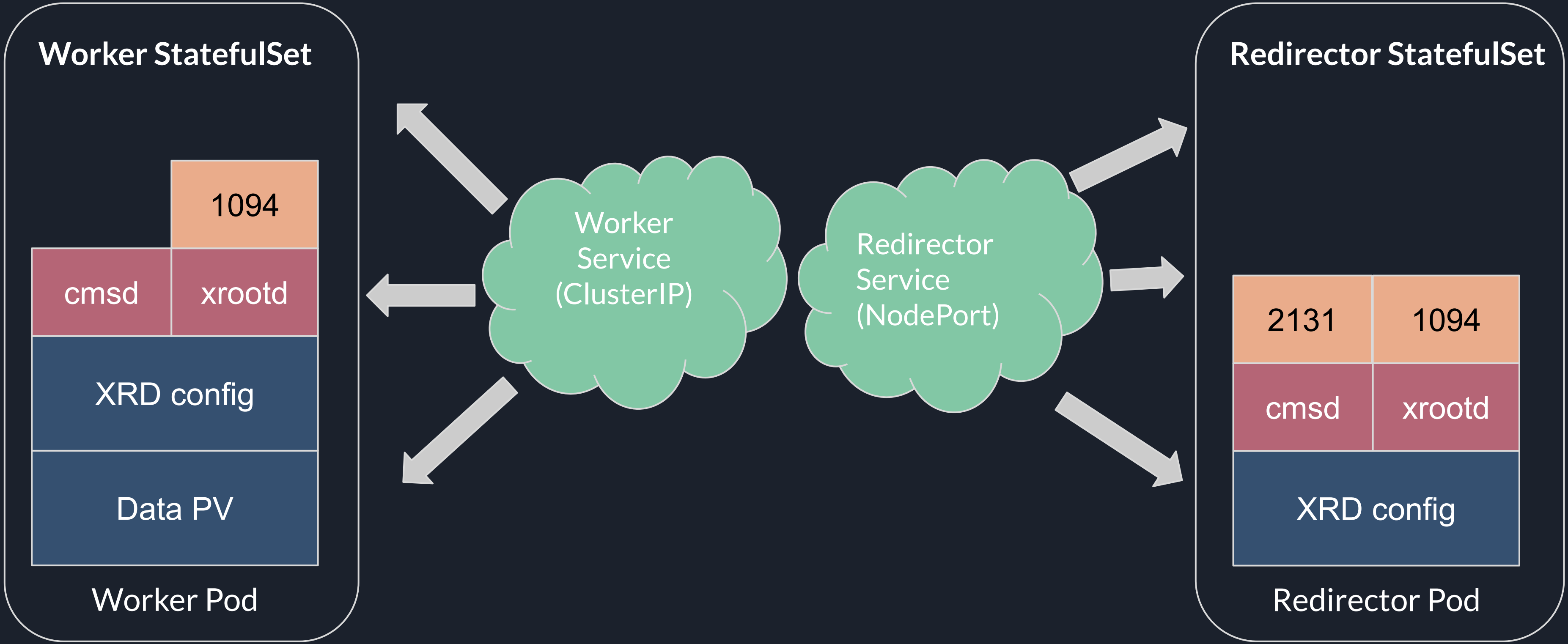
XRootD protocol enables high performance, scalable fault-tolerant access to data repositories of various kinds, including EOS.

It is meant to solve the **Any Data, Anytime, Anywhere (AAA)** requirement to access the remote files regardless if they are present in your region or halfway around the world!

It's possible by abstracting two types of nodes in any XRootD cluster:

1. **Redirectors -** These nodes coordinates the function of the cluster and enable communication via Intra-region and Cross-region redirection
2. **Workers -** These nodes are actually the ones storing and providing the data to the client

XRootD Cluster Architecture



Installation

OLM via OperatorHub

- Install OLM in your cluster
- Install **Subscription CR** for Xrootd operator
- OLM will now fetch the latest operator bundle image, belonging to the specified channel
- OLM will install the required CRDs, permissions, role and operator deployment
- Updating operator is seamlessly handled by OLM

Manually via script

- Deploy the operator using installation script
- Updating operator version requires manual re-installation.

Cluster Configuration via CRDs

XRootD CRD

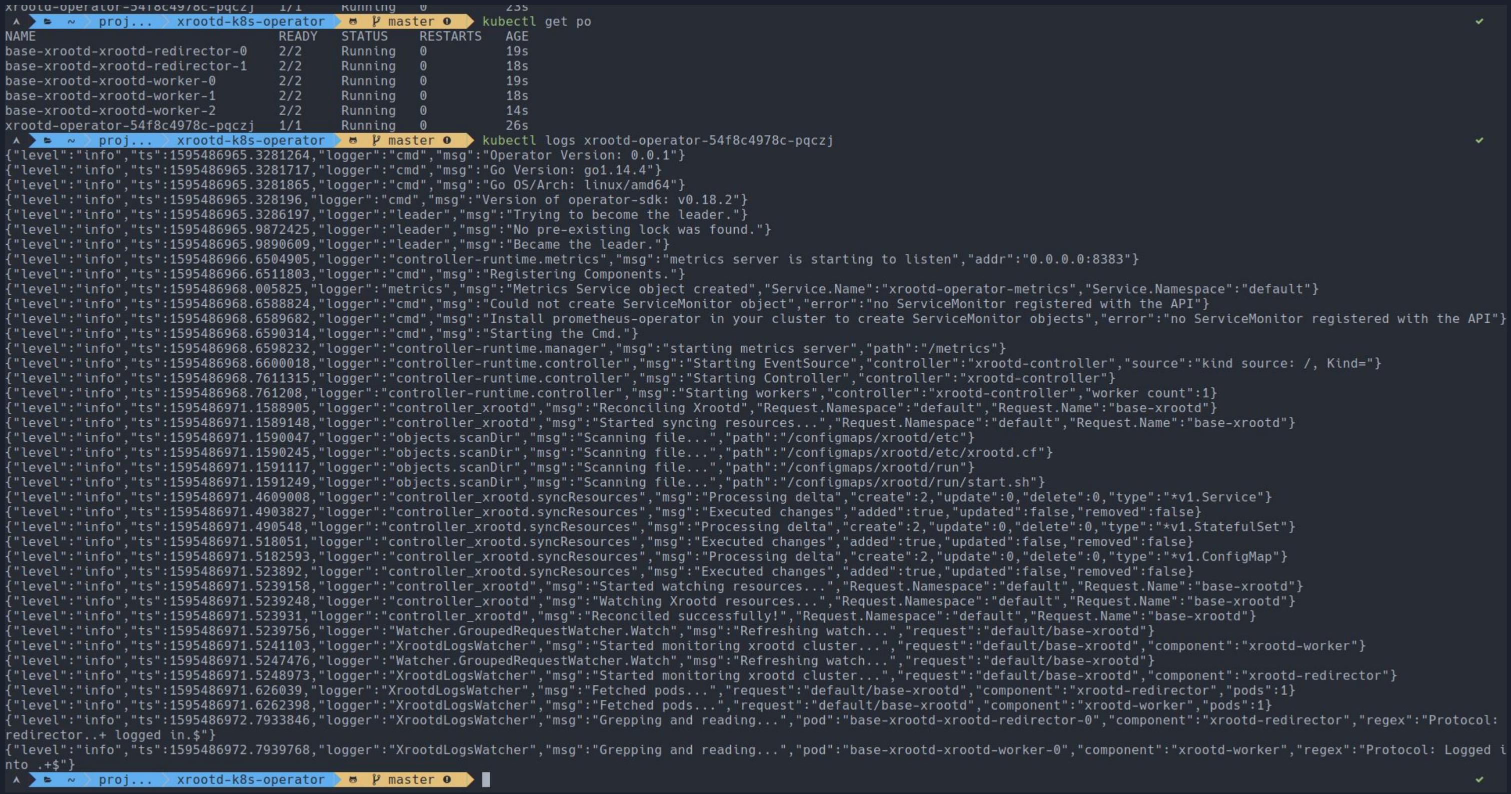
```yaml
apiVersion: xrootd.xrootd.org/v1alpha1
kind: XrootdCluster
metadata:
  name: sample-cluster
spec:
  version: 4.11.2
  redirector:
    replicas: 2
  worker:
    replicas: 3
    storage:
      capacity: "1Gi"
      class: "default"
```

XRootD Version Catalog CRD

```yaml
apiVersion: catalog.xrootd.org/v1alpha1
kind: XrootdVersion
metadata:
  name: 4.11.2
spec:
  version: 4.11.2
  deprecated: false
  image: "qserv/xrootd:v4.11.2"
```

Example

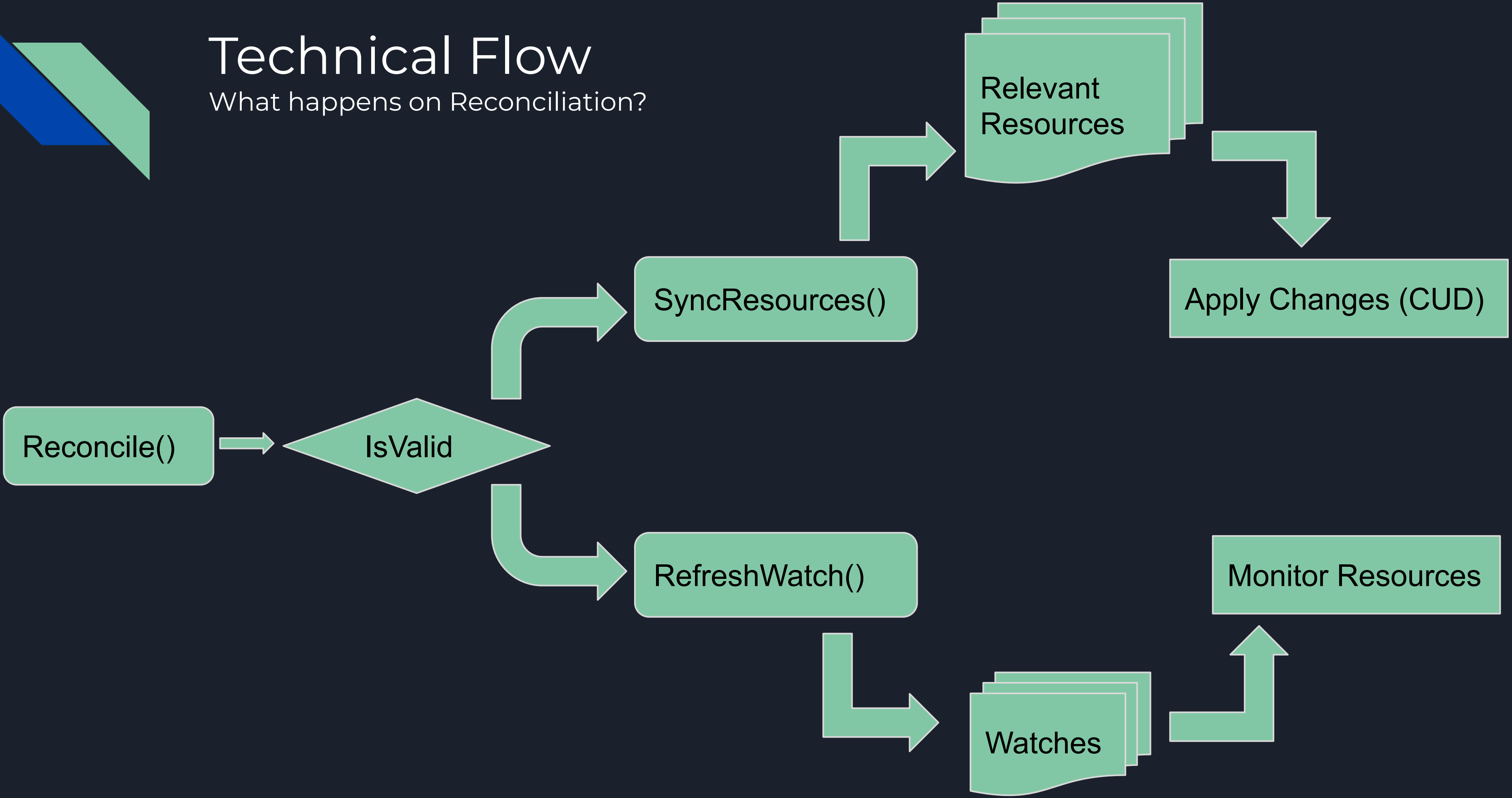
```
kubectl apply -k manifests/base
```



```
xrootd-operator-54f8c4978c-pqczj   1/1     Running   0          23s
 ~  proj...  xrootd-k8s-operator   master  kubectl get po
NAME                               READY   STATUS    RESTARTS   AGE
base-xrootd-xrootd-redirector-0    2/2     Running   0          19s
base-xrootd-xrootd-redirector-1    2/2     Running   0          18s
base-xrootd-xrootd-worker-0        2/2     Running   0          19s
base-xrootd-xrootd-worker-1        2/2     Running   0          18s
base-xrootd-xrootd-worker-2        2/2     Running   0          14s
xrootd-operator-54f8c4978c-pqczj   1/1     Running   0          26s
 ~  proj...  xrootd-k8s-operator   master  kubectl logs xrootd-operator-54f8c4978c-pqczj
{"level":"info","ts":1595486965.3281264,"logger":"cmd","msg":"Operator Version: 0.0.1"}
{"level":"info","ts":1595486965.3281717,"logger":"cmd","msg":"Go Version: go1.14.4"}
{"level":"info","ts":1595486965.3281865,"logger":"cmd","msg":"Go OS/Arch: linux/amd64"}
{"level":"info","ts":1595486965.328196,"logger":"cmd","msg":"Version of operator-sdk: v0.18.2"}
{"level":"info","ts":1595486965.3286197,"logger":"leader","msg":"Trying to become the leader."}
{"level":"info","ts":1595486965.9872425,"logger":"leader","msg":"No pre-existing lock was found."}
{"level":"info","ts":1595486965.9890609,"logger":"leader","msg":"Became the leader."}
{"level":"info","ts":1595486966.6504905,"logger":"controller-runtime.metrics","msg":"metrics server is starting to listen","addr":"0.0.0.0:8383"}
{"level":"info","ts":1595486966.6511803,"logger":"cmd","msg":"Registering Components."}
{"level":"info","ts":1595486968.005825,"logger":"metrics","msg":"Metrics Service object created","Service.Name":"xrootd-operator-metrics","Service.Namespace":"default"}
{"level":"info","ts":1595486968.6588824,"logger":"cmd","msg":"Could not create ServiceMonitor object","error":"no ServiceMonitor registered with the API"}
{"level":"info","ts":1595486968.6589682,"logger":"cmd","msg":"Install prometheus-operator in your cluster to create ServiceMonitor objects","error":"no ServiceMonitor registered with the API"}
{"level":"info","ts":1595486968.6590314,"logger":"cmd","msg":"Starting the Cmd."}
{"level":"info","ts":1595486968.6598232,"logger":"controller-runtime.manager","msg":"starting metrics server","path":"/metrics"}
{"level":"info","ts":1595486968.6600018,"logger":"controller-runtime.controller","msg":"Starting EventSource","controller":"xrootd-controller","source":"kind source: /, Kind="}
{"level":"info","ts":1595486968.7611315,"logger":"controller-runtime.controller","msg":"Starting Controller","controller":"xrootd-controller"}
{"level":"info","ts":1595486968.761208,"logger":"controller-runtime.controller","msg":"Starting workers","controller":"xrootd-controller","worker count":1}
{"level":"info","ts":1595486971.1588905,"logger":"controller_xrootd","msg":"Reconciling Xrootd","Request.Namespace":"default","Request.Name":"base-xrootd"}
{"level":"info","ts":1595486971.1589148,"logger":"controller_xrootd","msg":"Started syncing resources...","Request.Namespace":"default","Request.Name":"base-xrootd"}
{"level":"info","ts":1595486971.1590047,"logger":"objects.scanDir","msg":"Scanning file...","path":"/configmaps/xrootd/etc"}
{"level":"info","ts":1595486971.1590245,"logger":"objects.scanDir","msg":"Scanning file...","path":"/configmaps/xrootd/etc/xrootd.cf"}
{"level":"info","ts":1595486971.1591117,"logger":"objects.scanDir","msg":"Scanning file...","path":"/configmaps/xrootd/run"}
{"level":"info","ts":1595486971.1591249,"logger":"objects.scanDir","msg":"Scanning file...","path":"/configmaps/xrootd/run/start.sh"}
{"level":"info","ts":1595486971.4609008,"logger":"controller_xrootd.syncResources","msg":"Processing delta","create":2,"update":0,"delete":0,"type":"*v1.Service"}
{"level":"info","ts":1595486971.4903827,"logger":"controller_xrootd.syncResources","msg":"Executed changes","added":true,"updated":false,"removed":false}
{"level":"info","ts":1595486971.490548,"logger":"controller_xrootd.syncResources","msg":"Processing delta","create":2,"update":0,"delete":0,"type":"*v1.StatefulSet"}
{"level":"info","ts":1595486971.518051,"logger":"controller_xrootd.syncResources","msg":"Executed changes","added":true,"updated":false,"removed":false}
{"level":"info","ts":1595486971.5182593,"logger":"controller_xrootd.syncResources","msg":"Processing delta","create":2,"update":0,"delete":0,"type":"*v1.ConfigMap"}
{"level":"info","ts":1595486971.523892,"logger":"controller_xrootd.syncResources","msg":"Executed changes","added":true,"updated":false,"removed":false}
{"level":"info","ts":1595486971.5239158,"logger":"controller_xrootd","msg":"Started watching resources...","Request.Namespace":"default","Request.Name":"base-xrootd"}
{"level":"info","ts":1595486971.5239248,"logger":"controller_xrootd","msg":"Watching Xrootd resources...","Request.Namespace":"default","Request.Name":"base-xrootd"}
{"level":"info","ts":1595486971.523931,"logger":"controller_xrootd","msg":"Reconciled successfully!","Request.Namespace":"default","Request.Name":"base-xrootd"}
{"level":"info","ts":1595486971.5239756,"logger":"Watcher.GroupedRequestWatcher.Watch","msg":"Refreshing watch...","request":"default/base-xrootd"}
{"level":"info","ts":1595486971.5241103,"logger":"XrootdLogsWatcher","msg":"Started monitoring xrootd cluster...","request":"default/base-xrootd","component":"xrootd-worker"}
{"level":"info","ts":1595486971.5247476,"logger":"Watcher.GroupedRequestWatcher.Watch","msg":"Refreshing watch...","request":"default/base-xrootd"}
{"level":"info","ts":1595486971.5248973,"logger":"XrootdLogsWatcher","msg":"Started monitoring xrootd cluster...","request":"default/base-xrootd","component":"xrootd-redirector"}
{"level":"info","ts":1595486971.626039,"logger":"XrootdLogsWatcher","msg":"Fetched pods...","request":"default/base-xrootd","component":"xrootd-redirector","pods":1}
{"level":"info","ts":1595486971.6262398,"logger":"XrootdLogsWatcher","msg":"Fetched pods...","request":"default/base-xrootd","component":"xrootd-worker","pods":1}
{"level":"info","ts":1595486972.7933846,"logger":"XrootdLogsWatcher","msg":"Grepping and reading...","pod":"base-xrootd-xrootd-redirector-0","component":"xrootd-redirector","regex":"Protocol:
redirector..+ logged in.$"}
{"level":"info","ts":1595486972.7939768,"logger":"XrootdLogsWatcher","msg":"Grepping and reading...","pod":"base-xrootd-xrootd-worker-0","component":"xrootd-worker","regex":"Protocol: Logged i
nto .+$"}
 ~  proj...  xrootd-k8s-operator   master 
```

Technical Flow

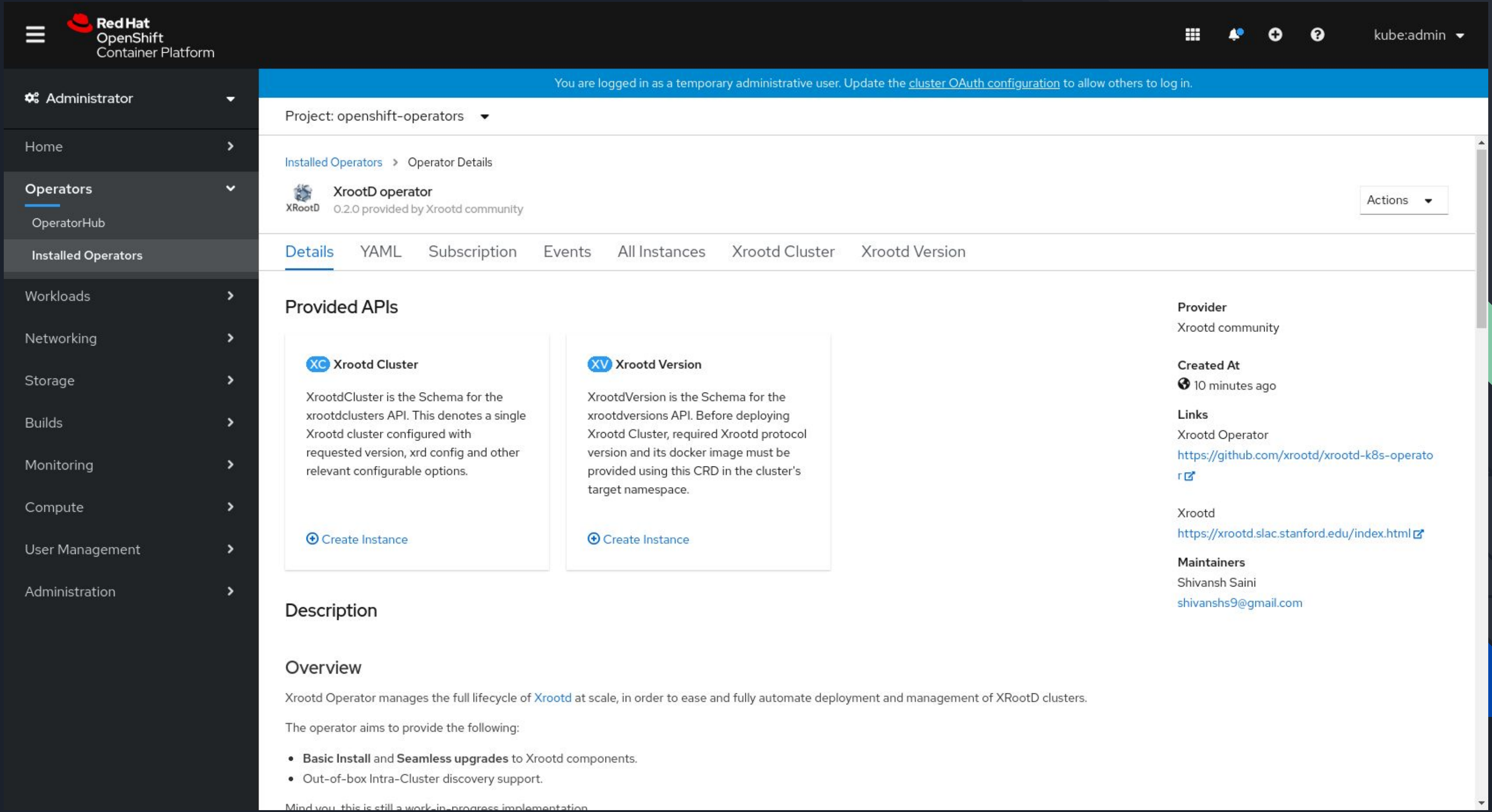
What happens on Reconciliation?



Achieved Goals

- Developed a k8s operator, demonstrating how to deploy and manage an XRootD service at scale using Kubernetes.
- Wrote documentation for the CRDs and configurations of the operator.
- Implemented Kubernetes operator’s advanced features like seamless upgrades and deep insights.
- Added support for OLM for easy installation and seamless upgrades of the operator.
- Made the operator compliant and tested with both upstream Kubernetes and Openshift 4. Used OLM descriptors to add UI controls for cluster status and creation.
- Added unit and integration tests increasing the test coverage to **~72%**.
- Wrote E2E test scenarios to ensure the operator works as intended in real-world scenarios.
- Followed best CI/CD practices with custom Github Workflows.
- Developed a new Github Action to setup Operator-SDK and used it for xrootd github workflows.
- Migrated the operator to Operator-SDK v1.
- Promoted best practices in the operator by ensuring **A+** in go report card and OLM scorecard tests.
- Published the operator to operatorhub.io
- Explained to the XRootD community how to leverage this operator in order to ease and scale-up worldwide XRootD clusters management.

Screenshots



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: openshift-operators

Installed Operators > Operator Details

XrootD operator
0.2.0 provided by Xrootd community

Provided APIs

XC Xrootd Cluster

XrootdCluster is the Schema for the xrootdclusters API. This denotes a single Xrootd cluster configured with requested version, xrd config and other relevant configurable options.

Create Instance

XV Xrootd Version

XrootdVersion is the Schema for the xrootdversions API. Before deploying Xrootd Cluster, required Xrootd protocol version and its docker image must be provided using this CRD in the cluster's target namespace.

Create Instance

Provider
Xrootd community

Created At
10 minutes ago

Links
Xrootd Operator
https://github.com/xrootd/xrootd-k8s-operator

Xrootd
https://xrootd.slac.stanford.edu/index.html

Maintainers
Shivansh Saini
shivanshs9@gmail.com

Description

Overview

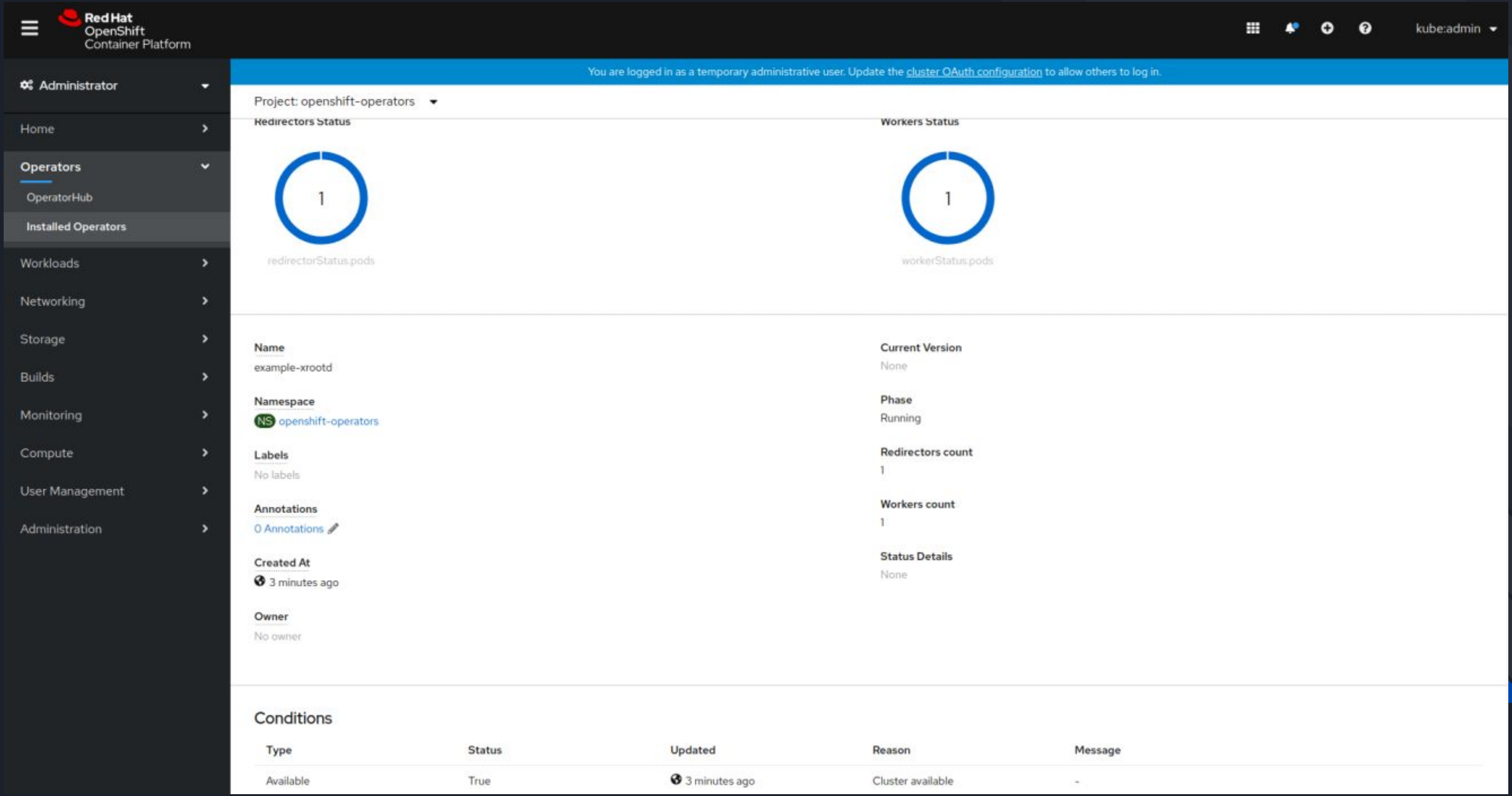
Xrootd Operator manages the full lifecycle of Xrootd at scale, in order to ease and fully automate deployment and management of XRootD clusters.

The operator aims to provide the following:

- **Basic Install** and **Seamless upgrades** to Xrootd components.
- Out-of-box Intra-Cluster discovery support.

Mind you, this is still a work-in-progress implementation

Screenshots



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: openshift-operators

Redirectors Status

1

redirectorStatus.pods

Workers Status

1

workerStatus.pods

Name	example-xrootd
Namespace	NS openshift-operators
Labels	No labels
Annotations	0 Annotations
Created At	3 minutes ago
Owner	No owner
Current Version	None
Phase	Running
Redirectors count	1
Workers count	1
Status Details	None

Conditions

Type	Status	Updated	Reason	Message
Available	True	3 minutes ago	Cluster available	-