State of the OPERATOR FRAMEWORK

Making Operators easier to develop and consume

Daniel Messer
Product Management

Jason Dobies
Technical Marketing

OPERATOR SDK

Build, package and test an Operators without the toil of integrating with Kubernetes APIs

OPERATOR LIFECYCLE MANAGER

Central point on cluster to deploy, and update, and generally manage the availability of Operators

OPERATORHUB.io

Community Catalog to publish to and install Operators from

https://github.com/operator-framework

What you can do today

Operator SDK

Enabling everybody to write Operators

HELM

Install & Update

Install & Update + Day 2 Operations

GO

Install & Update + Day 2 Operations +
Metrics/Alert analysis & workload tuning

Packaging

Testing & Validation

Operator SDK

Enabling everybody to write Operators

Support for Helm 3
Build Operators from Helm v2 and v3 charts

Ansible collection
Ansible Operator supports k8s module collection

Custom metrics
Every Operator supports custom metric endpoints

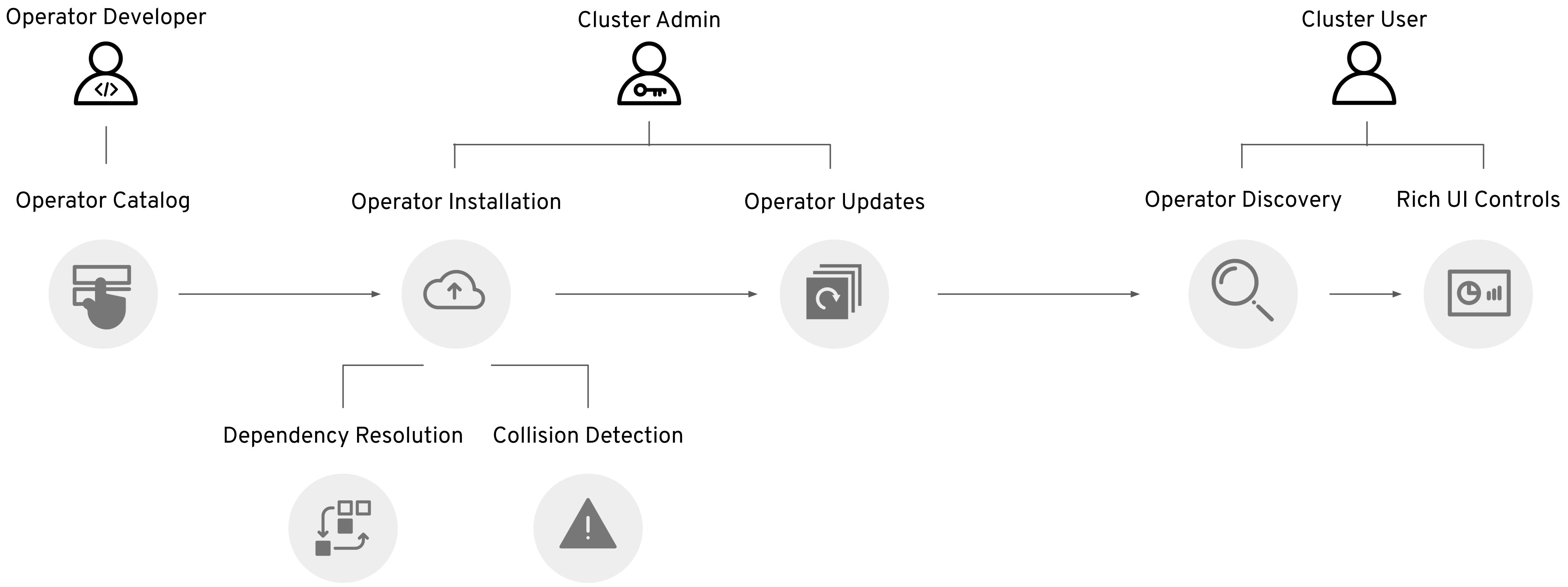
Generate Packaging
Operator Metadata for OLM gets generated

Kubernetes Compatibility
Keep in sync with new Kubernetes releases

Scorecard v2
Enable testing your Operator in a pipeline

Operator Lifecycle Manager

The missing control panel for Operators



Operator Lifecycle Manager

Making it safe to extend your cluster

Operator Configurability
Operator-level config that persists across updates

Proxy and Disconnected Support
Pass down Proxy config and enable catalog mirroring

User-defined Update path
Granularly define the supported update path

Usability Improvements
Better health data and error messages

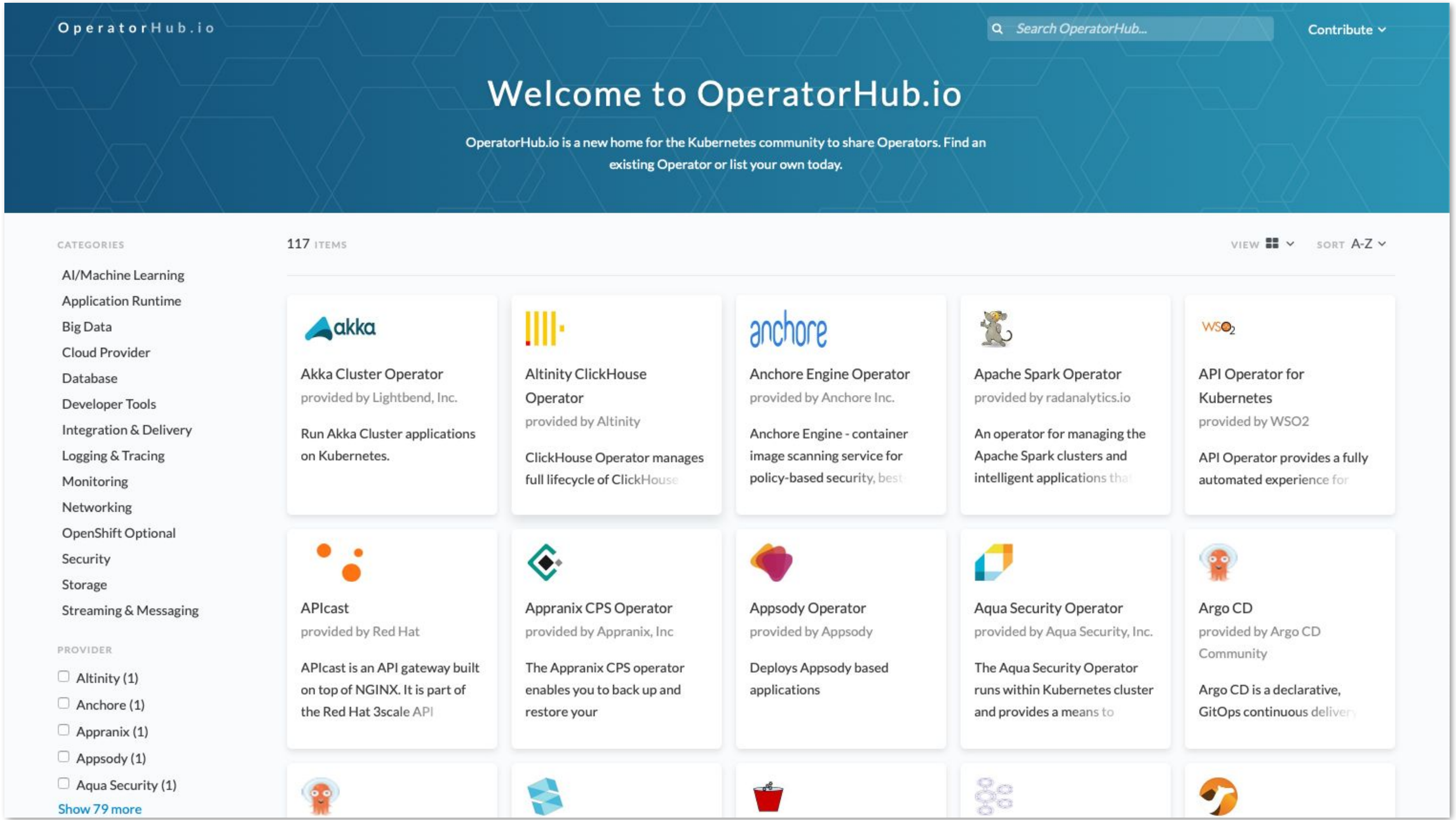
Operator Tenancy
Install Operator for certain namespaces only

Catalogs in container images
Ship catalogs in container images from registries

100+ Operators published today

OperatorHub.io

The community registry for Kubernetes Operators



Welcome to OperatorHub.io

OperatorHub.io is a new home for the Kubernetes community to share Operators. Find an existing Operator or list your own today.

CATEGORIES

AI/Machine Learning
Application Runtime
Big Data
Cloud Provider
Database
Developer Tools
Integration & Delivery
Logging & Tracing
Monitoring
Networking
OpenShift Optional
Security
Storage
Streaming & Messaging

PROVIDER

- Altinity (1)
- Anchore (1)
- Appranix (1)
- Appsody (1)
- Aqua Security (1)

Show 79 more

117 ITEMS

VIEW SORT A-Z

Akka Cluster Operator
provided by Lightbend, Inc.
Run Akka Cluster applications on Kubernetes.

Altinity ClickHouse Operator
provided by Altinity
ClickHouse Operator manages full lifecycle of ClickHouse

Anchore Engine Operator
provided by Anchore Inc.
Anchore Engine - container image scanning service for policy-based security, best

Apache Spark Operator
provided by radanalytics.io
An operator for managing the Apache Spark clusters and intelligent applications that

API Operator for Kubernetes
provided by WSO2
API Operator provides a fully automated experience for

APIcast
provided by Red Hat
APIcast is an API gateway built on top of NGINX. It is part of the Red Hat 3scale API

Appranix CPS Operator
provided by Appranix, Inc
The Appranix CPS operator enables you to back up and restore your

Appsody Operator
provided by Appsody
Deploys Appsody based applications

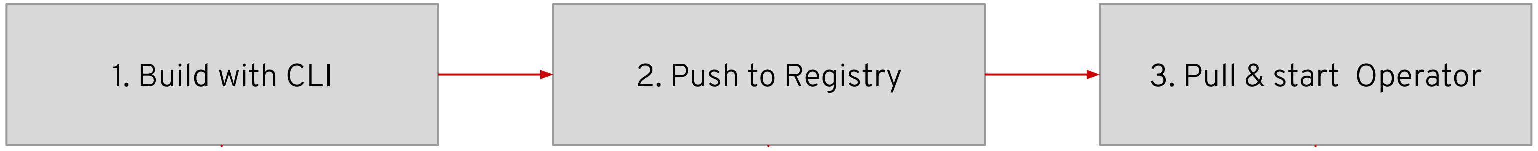
Aqua Security Operator
provided by Aqua Security, Inc.
The Aqua Security Operator runs within Kubernetes cluster and provides a means to

Argo CD
provided by Argo CD Community
Argo CD is a declarative, GitOps continuous delivery

The road ahead

New Operator Bundle Format

Ship your Operator Metadata in Container Images!



```
$ operator-sdk bundle create --directory=0.1.0

$ tree test
test
├── my-manifests
│   ├── cluster.crd.yaml
│   └── test-operator.v0.1.0.csv.yaml
├── metadata
│   └── annotations.yaml
└── Dockerfile


$ podman build .
$ podman push quay.io/test/test-operator:v0.1.0
```

```
$ kubectl apply -f -

apiVersion:
operators.operatorframework.io/v2alpha1
kind: Operator
metadata:
  name: test-operator
spec:
  bundle:
    image:
      quay.io/test/test-operator:v0.1.0
```

OperatorHub.io will accept this new format among the current one.

New Operator Package Manager

Build and distribute catalogs in `yum/dnf` style

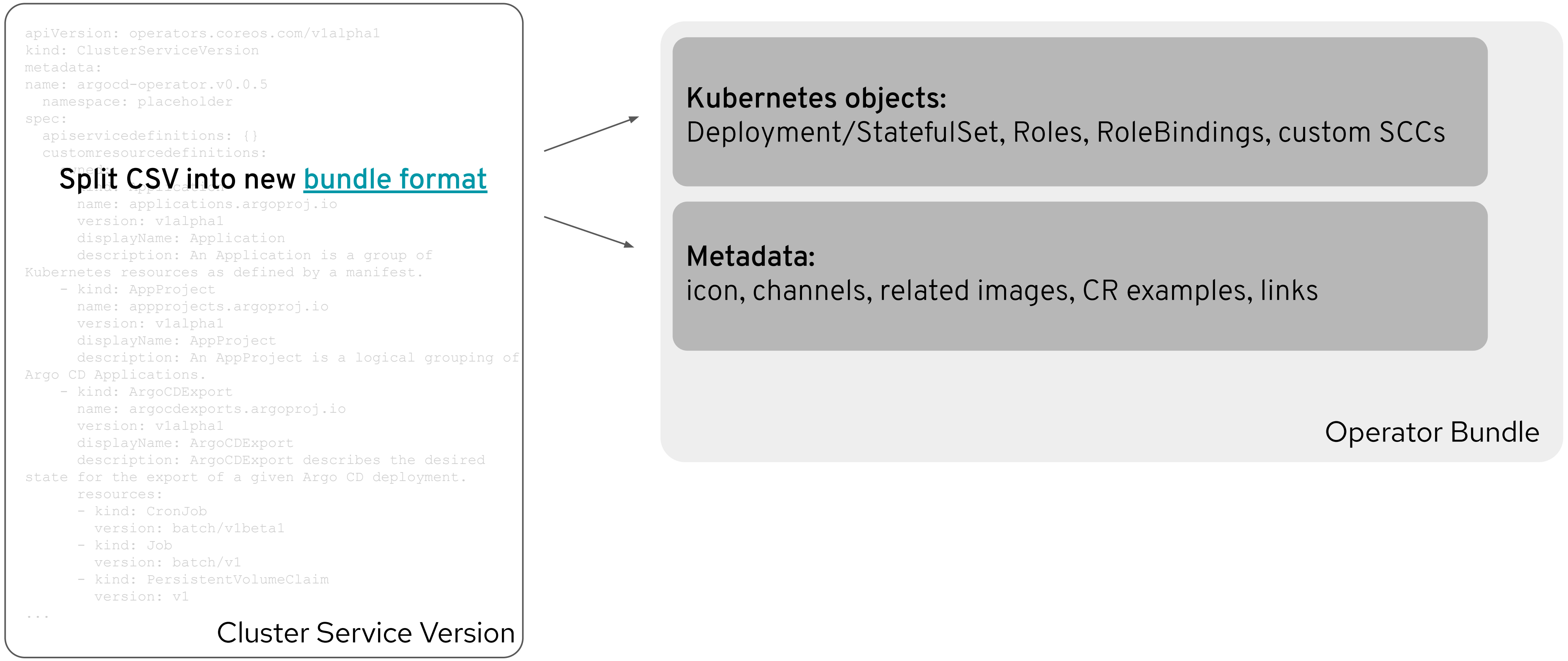
```
$ podman push quay.io/test/test-operator:v0.1.0
```

```
# start a new index
$ opm index add \
     --bundle quay.io/test/test-operator:v0.1.0 \
     --tag quay.io/catalog/my-catalog:latest


# add to existing index
$ opm index add \
     --bundle quay.io/test/test-operator:v0.1.1 \
     --from-index quay.io/catalog/my-catalog:latest
     --tag quay.io/catalog/my-catalog:latest
```

```
$ kubectl apply -f -

  apiVersion: operators.coreos.com/v1alpha1
  kind: CatalogSource
  metadata:
    name: my-catalog
    namespace: default
  spec:
    sourceType: grpc
    image: quay.io/catalog/my-catalog:latest
    updateStrategy:
      registryPoll:
        interval: 30m
```

CSV-less bundles

Give us your Kubernetes manifest and we'll do the rest (almost)



```
apiVersion: operators.coreos.com/v1alpha1
kind: ClusterServiceVersion
metadata:
name: argocd-operator.v0.0.5
  namespace: placeholder
spec:
  apiservicedefinitions: {}
  customresourcedefinitions:
    owned:
    - kind: Application
      name: applications.argoproj.io
      version: v1alpha1
      displayName: Application
      description: An Application is a group of
Kubernetes resources as defined by a manifest.
    - kind: AppProject
      name: appprojects.argoproj.io
      version: v1alpha1
      displayName: AppProject
      description: An AppProject is a logical grouping of
Argo CD Applications.
    - kind: ArgoCDExport
      name: argocdexports.argoproj.io
      version: v1alpha1
      displayName: ArgoCDExport
      description: ArgoCDExport describes the desired
state for the export of a given Argo CD deployment.
      resources:
      - kind: CronJob
        version: batch/v1beta1
      - kind: Job
        version: batch/v1
      - kind: PersistentVolumeClaim
        version: v1
...
```

Split CSV into new bundle format

Cluster Service Version

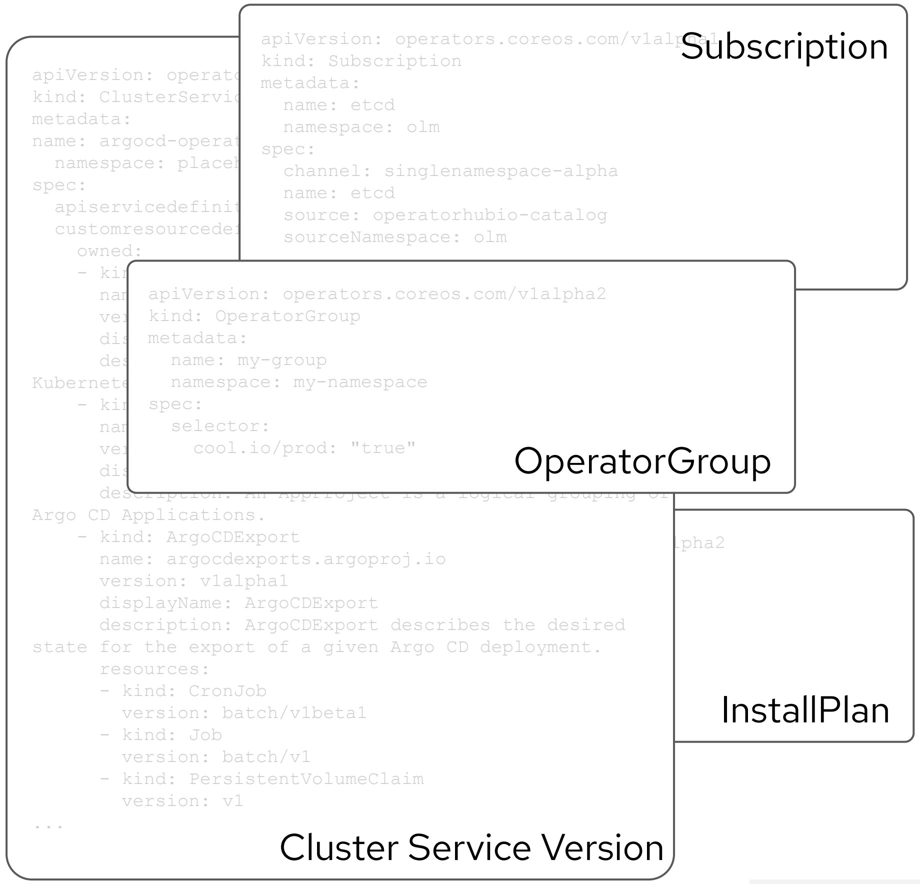
Kubernetes objects:
Deployment/StatefulSet, Roles, RoleBindings, custom SCCs

Metadata:
icon, channels, related images, CR examples, links

Operator Bundle

New Operator API

One Kubernetes object to rule them all



```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: etcd
  namespace: olm
spec:
  channel: singlenamespace-alpha
  name: etcd
  source: operatorhubio-catalog
  sourceNamespace: olm
```

```
apiVersion: operators.coreos.com/v1alpha2
kind: OperatorGroup
metadata:
  name: my-group
  namespace: my-namespace
spec:
  selector:
    cool.io/prod: "true"
```

```
apiVersion: operato
kind: ClusterServic
metadata:
name: argocd-operat
  namespace: placeh
spec:
  apiservicedefinit
  customresourcede
    owned:
    - kin
      nam
      ver
      dis
      des
Kubernete
    - kin
      nam
      ver
      dis
      description: An AppProject is a logical grouping of
Argo CD Applications.
    - kind: ArgoCDExport
      name: argocdexports.argoproj.io
      version: v1alpha1
      displayName: ArgoCDExport
      description: ArgoCDExport describes the desired
state for the export of a given Argo CD deployment.
      resources:
      - kind: CronJob
        version: batch/v1beta1
      - kind: Job
        version: batch/v1
      - kind: PersistentVolumeClaim
        version: v1
...
```

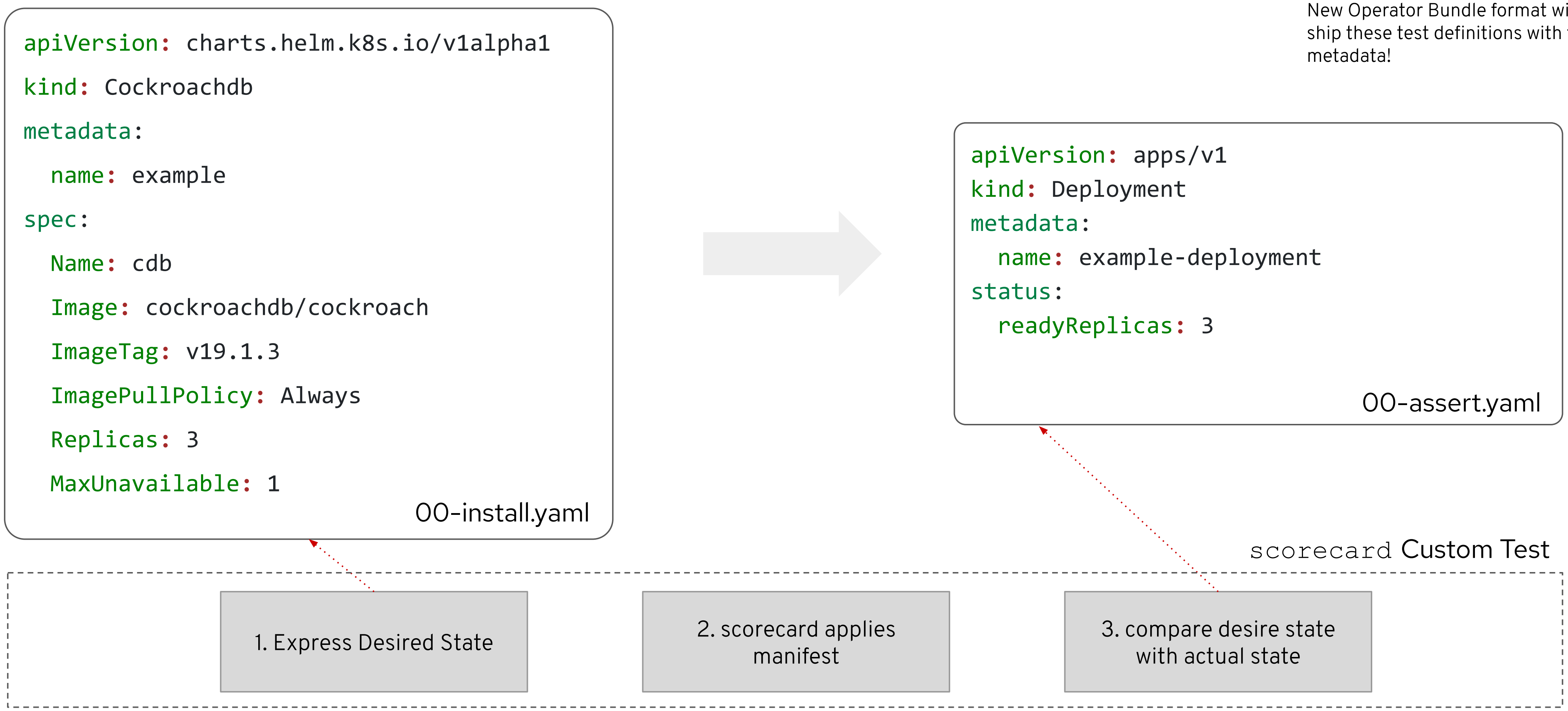
```
apiVersion: operators.coreos.com/v2alpha1
kind: Operator
metadata:
  name: plumbus
spec:
  updates:
    type: CatalogSource
    catalogSource:
      package: plumbus
      channel: stable
      entrypoint: plumbus.v2.0.0
      approval: Automatic
      ref:
        name: community
        namespace: my-ns

status:
  updates:
    available:
    - name: community
      channel: beta

  metadata:
    displayName: Plumbus
    description: Welcome ...
    version: 2.0.0-alpha
    apis:
      provides:
      - group: how.theydoit.com
        version: v2alpha1
```

```
# Discover Operators!
$ kubectl get operators
```

SDK's scorecard test tool supports kuttl

Use YAML to declare assertion based tests

New Operator Bundle format will allow you to ship these test definitions with the rest of the metadata!



```yaml
apiVersion: charts.helm.k8s.io/v1alpha1
kind: Cockroachdb
metadata:
  name: example
spec:
  Name: cdb
  Image: cockroachdb/cockroach
  ImageTag: v19.1.3
  ImagePullPolicy: Always
  Replicas: 3
  MaxUnavailable: 1
```

00-install.yaml

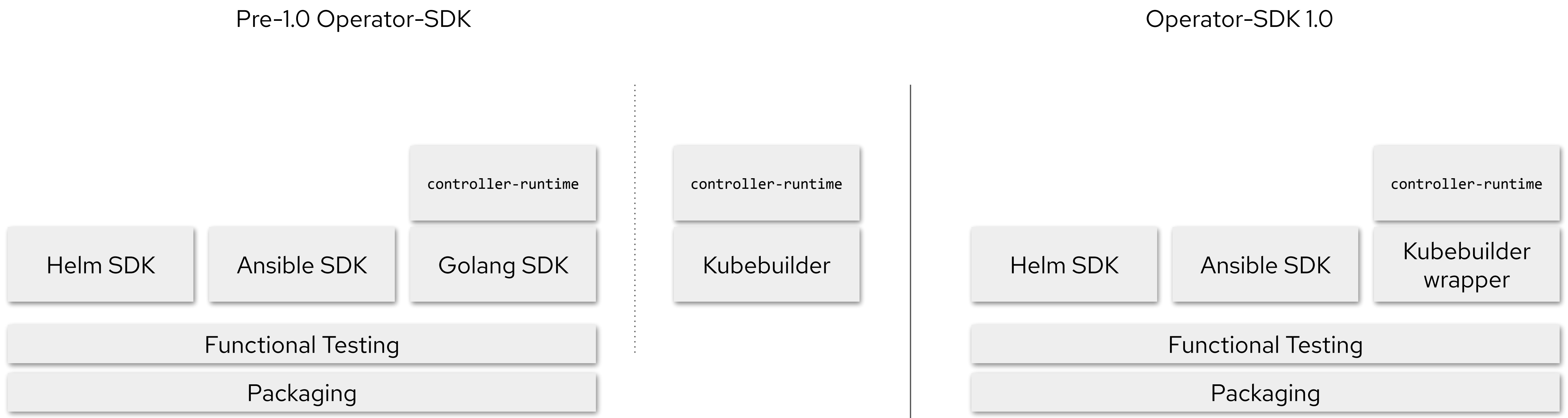
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: example-deployment
status:
  readyReplicas: 3
```

00-assert.yaml

scorecard Custom Test
1. Express Desired State
2. scorecard applies manifest
3. compare desire state with actual state

Operator-SDK goes Kubebuilder

Seamlessly import Kubebuilder Golang-Operator projects



Incompatible scaffolding, CLI and project directory layout. Two separate Golang scaffolding implementations.

Import Kubebuilder projects and re-using existing CLI switches. One common Golang scaffolding implementation

Summary

Operator Developer Enhancements

Improved workflow for offline catalogs and upgrade testing

CSV-less bundles
Define your Operator with (mostly) just Kubernetes manifests

Build catalogs with Kubernetes tools
Use tools that you are familiar with to package up your Operator for consumption.

Semver-based upgrade logic
Add a simpler mechanism to track upgrade logic to sit alongside the options for a complex graph

Integrated packaging
SDK has can package, validate and run Operators using OLM directly

Bundle custom functional tests
Provide developers a tool to package up tests that cover important use-cases for their software.

Write Operator in Kubebuilder-style
SDK will adopt Kubebuilder for Golang Operators

Cluster Admin Enhancements

Easier interaction with OLM

Soft dependencies / disable resolution

OLM supports optional dependencies or force-overriding dependency resolution

OPM for offline mirroring of images

Easily mirror all of the content required to run Operators behind a firewalled network using `opm`

New Operator object

Simplify registering/running Operators for all users, but is very helpful during testing and development.

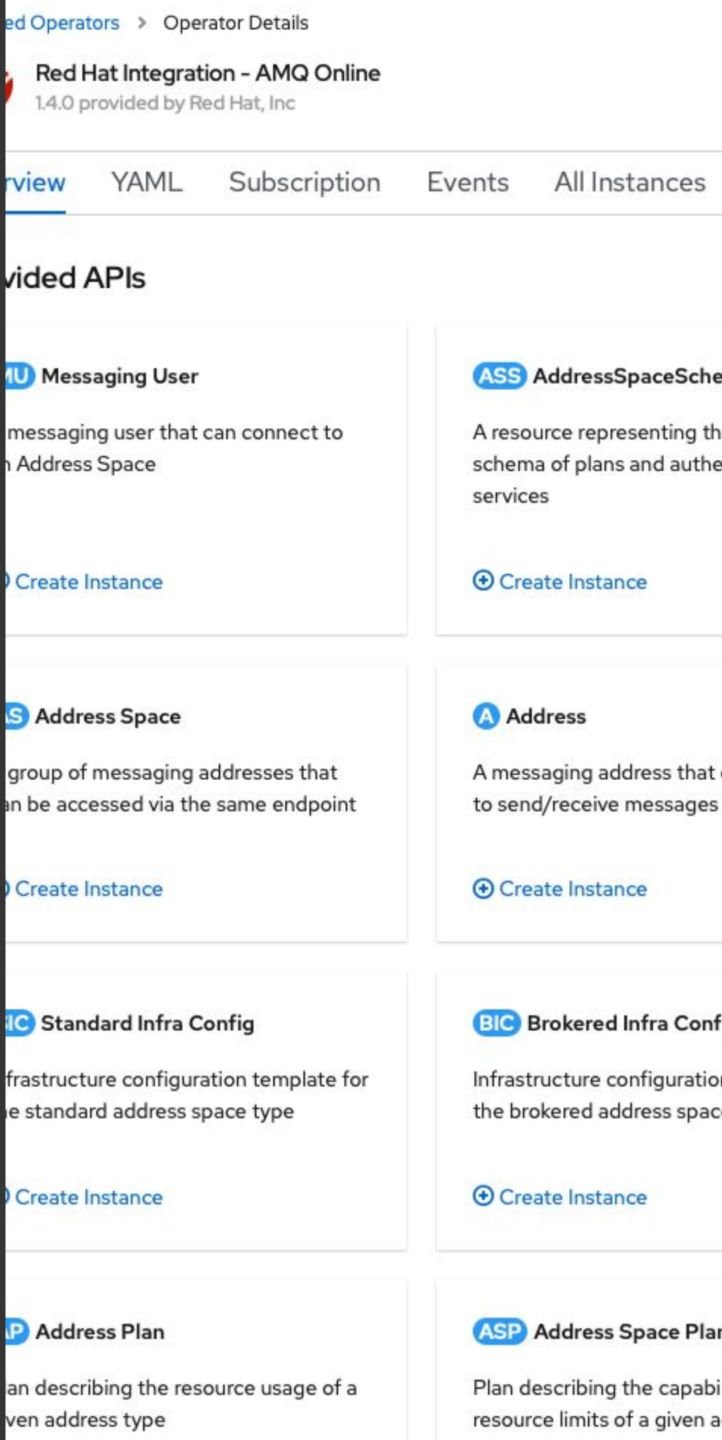
OLM manages webhooks

OLM supports Operators with webhooks and manages certification creation/rotation

Select Operator version to install

Pick an older version to install and get alerted on updates being available

Demo!


ed Operators > Operator Details

Red Hat Integration - AMQ Online
1.4.0 provided by Red Hat, Inc

vided APIs

MU Messaging User
messaging user that can connect to
Address Space
Create Instance

ASS AddressSpaceSche
A resource representing th
schema of plans and authe
services
Create Instance

S Address Space
group of messaging addresses that
n be accessed via the same endpoint
Create Instance

A Address
A messaging address that
to send/receive messages
Create Instance

IC Standard Infra Config
frastructure configuration template for
e standard address space type
Create Instance

BIC Brokered Infra Conf
Infrastructure configuratio
the brokered address spac
Create Instance

P Address Plan
an describing the resource usage of a
ven address type

ASP Address Space Plar
Plan describing the capabi
resource limits of a given a

Thank you

If you like the Operator Framework please add a GitHub star. If you want to contribute we are looking forward to code, documentation or any other contribution and ideas!

https://github.com/operator-framework

#kubernetes-operators on the k8s slack

https://groups.google.com/forum/#!forum/operator-framework

https://operatorhub.io