State of the Operators

Framework, SDKs, and beyond

Daniel Messer
Product Manager, OpenShift
OPERATORS ACROSS THE INDUSTRY

etcd

Prometheus

kafka

STRIMIZI

Couchbase

TensorFlow

Oracle WebLogic

Spark

mongoDB

confluent

redis

dynatrace

cassandra

CRUNCHY
Enterprise PostgreSQL

JAEGER

...and many more

https://github.com/operator-framework/awesome-operators
Why do we need Operators?
Why do we need Operators?

Containers brought simplicity to the development world

```bash
$ docker pull postgres
$ docker pull redis

$ docker run --name some-postgres -e POSTGRES_PASSWORD=foo -d postgres
$ docker run --name some-redis -d redis
```
Existing operational logic in Kubernetes

- Deployment
- ReplicaSet
- StatefulSet
- DaemonSet
- CronJob
- ...

[Image of the document page]
- Resize/Upgrade
- Reconfigure
- Backup
- Healing
OK, but what *is* an Operator?
What is an Operator?

- An operator is a Kubernetes native application
  - Leverages the Kubernetes API (usable with kubectl)
  - Runs on Kubernetes as containers
  - Resembles a custom controller
- Purposely built for an application
  - Operational knowledge baked in and automated
    - handling upgrades from one version to another
    - handling complex failure recovery scenarios
    - scaling a stateful application up and down
  - Best suited for complex and stateful applications (but not only!)
  - Example: a Prometheus operator specifically designed for it
• **Operator SDK** - Allows developers to build, package and test an Operator based on your expertise without requiring all the knowledge of Kubernetes API complexities

• **Operator Lifecycle Manager** - Helps you to install, and update, and generally manage the lifecycle of all of the Operators (and their associated services) running across your clusters

• **Operator Metering** - Enable usage reporting for Operators and resources within Kubernetes
OPERATOR FRAMEWORK IN ACTION

DEVELOPER

OPERATOR SDK

“create new operator”

scaffolding

custom logic

KUBERNETES OPERATOR

=
TYPES OF OPERATORS

Phase I: Basic Install
Automated application provisioning and configuration management

Phase II: Seamless Upgrades
Patch and minor version upgrades supported

Phase III: Full Lifecycle
App lifecycle, storage lifecycle (backup, failure recovery)

Phase IV: Deep Insights
Metrics, alerts, log processing and workload analysis

Phase V: Auto Pilot
Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning

HELM
ANSIBLE
GO
CREATE AN OPERATOR WITHOUT CODING

$ operator-sdk new cockroachdb-operator --type=helm --helm-chart stable/cockroachdb
OPERATOR FRAMEWORK IN ACTION

**Developer**
- SDK
- + { | | } = **Operator SDK**
- “create new operator”

**Administrator**
- Operator lifecycle manager
- “list packages”
- “make available to”
- package
- metadata
- subscription
- namespace
- “subscribe to channel”

**User**
- “create new operator”
- “collect metrics”
- “subscribe to channel”

**Operator Metering**
- managed application
- “Create application”
- operator instance
- “list packages”
- “subscribe to channel”

**Operator**
- scaffold
- custom logic
- Kubernetes
- +
- Operator lifecycle manager
Operator Hub - Allows administrators to selectively make operators available from curated sources to users in the cluster.
COMMUNITY OPERATOR CONTRIBUTION

DEVELOPER

OPERATOR SDK

>-

"create new operator"

scaffolding

custom logic

KUBERNETES OPERATOR

COMMUNITY REPO

QUAY

OPERATOR HUB

USER

"browse catalog"

"create new operator"

"create application"

managed application

operator instance

"install"
Demo
What’s ahead
OPERATOR SDK & LIFECYCLE

**Cross-Platform Support**

Objectives:
- OCP, OKD and k8s consistency

Features:
- Universal Base Image Support
- installation for non-OCP clusters

Stage: Development

**Operator Testability**

Objectives:
- aid developers with e2e testing
- validate operator maturity

Features:
- scorecard utility

Stage: Prototype

**Continuous Testing**

Objectives:
- automated scorecard testing
- validate operator maturity

Stage: Discovery

**Partner Enablement**

Objectives:
- UBI support for partners
- scorecard for partners

Stage: Discovery

**No-Fuzz Operator Install**

Features:
- OperatorGroups

Stage: Prototype

**Stage: Prototype**

**Stage: Discovery**

**Stage: Prototype**

**TBD, ideas around over-the-air updates, operator status aggregated from operands...**
https://github.com/operator-framework/community-operators
METERING/CHARGEBACK

- Testing a developer preview now
  - Install from OperatorHub
- Base functionality on all providers
- Tie into cloud providers for $$
- Included reports for 80% use-case
  - Customers can write custom reports and time periods
- Popular use-case: shame teams over requesting RAM
Where to get started?

- [https://github.com/operator-framework/getting-started](https://github.com/operator-framework/getting-started)
- [https://github.com/operator-framework/community-operators](https://github.com/operator-framework/community-operators)
- [https://commons.openshift.org/sig/operators.html](https://commons.openshift.org/sig/operators.html)
- [#kubernetes-operators](https://groups.google.com/forum/#!forum/operator-framework) on the kubernetes slack
- [https://groups.google.com/forum/#!forum/operator-framework](https://groups.google.com/forum/#!forum/operator-framework)
Thank you!