redhat.

State of the Operators

Framework and SDKs

Openshift and Cloud Native Rome meetup, March 4th, 2019

Tero Ahonen
Principal Specialist Solution Architect, Openshift

OPERATORS ACROSS THE INDUSTRY

...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

```
$ 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
- …

- 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 FRAMEWORK

redhat.

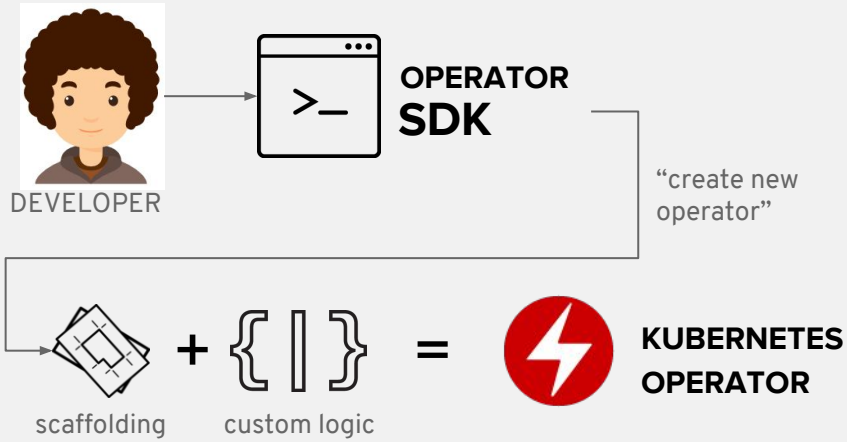
OPERATOR FRAMEWORK

- **Operator SDK** - Allows developers to build, package and test an Operator based on your expertise without requiring all the knowledge of Kubernetes API complexities

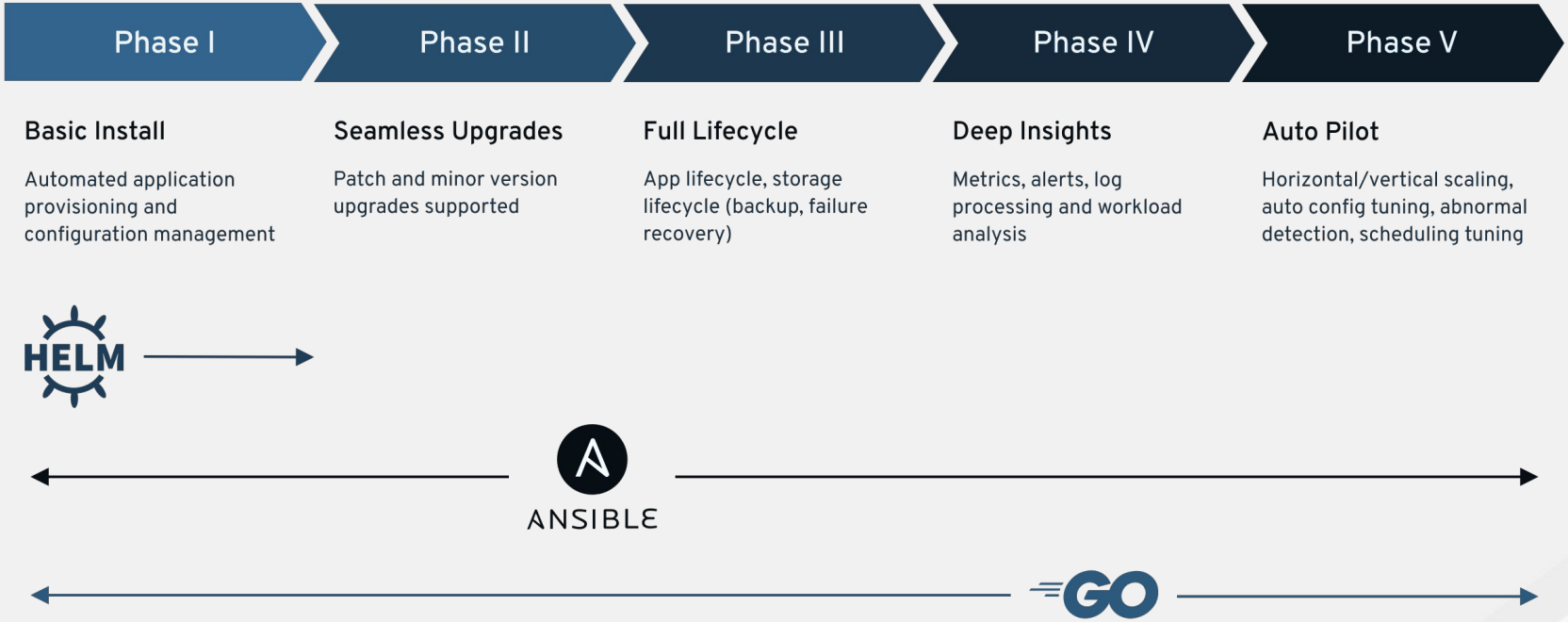
- **Operator Lifecycle Manager** - Helps you to install, and update, and generally manage the lifecycle of all of the Operators (and their associated services) running across your clusters

- **Operator Metering** - Enable usage reporting for Operators and resources within Kubernetes

OPERATOR FRAMEWORK IN ACTION



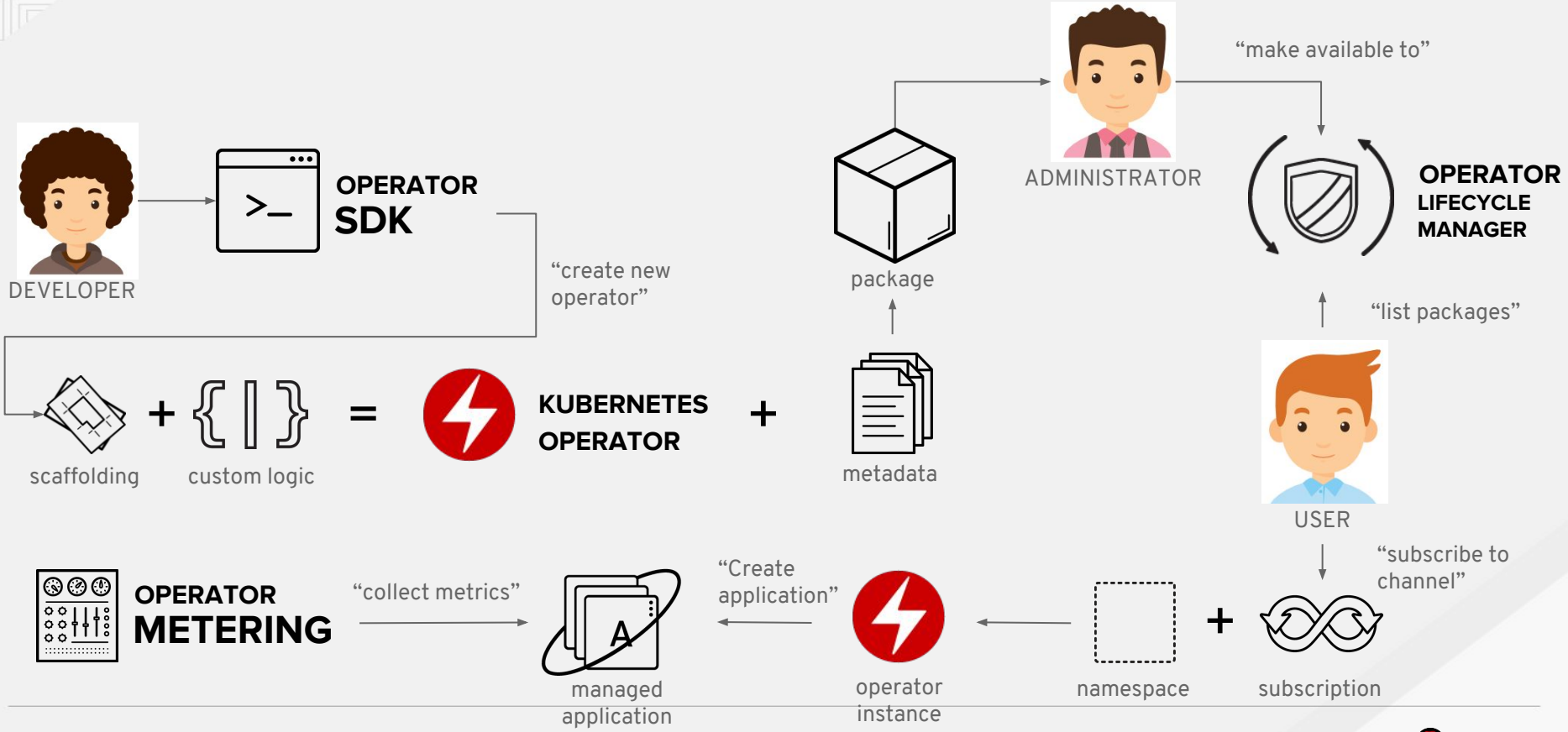
TYPES OF OPERATORS



CREATE AN OPERATOR WITHOUT CODING

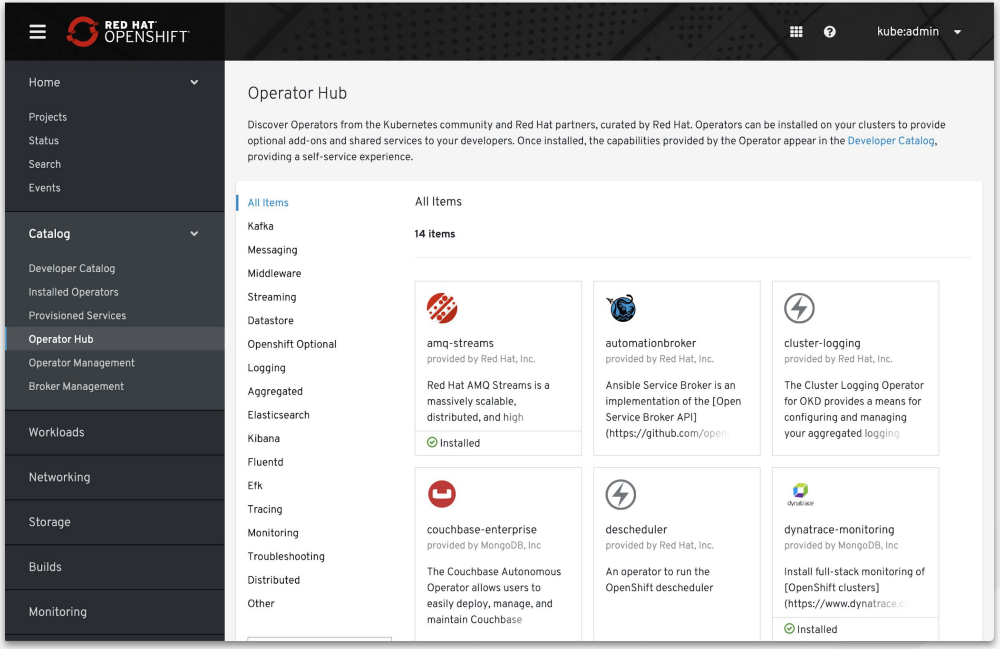
```
$ operator-sdk new cockroachdb-operator --type=helm --helm-chart stable/cockroachdb
```

OPERATOR FRAMEWORK IN ACTION



OPERATORS IN OPENSHIFT

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

pull request

COMMUNITY REPO

QUAY

OPERATOR
HUB

"browse catalog"

USER

"install"

"Create application"

operator instance

"collect metrics"

managed application

OPERATOR
METERING

redhat.

OPERATORS IN OPENSHIFT

cluster-kube-apiserver-operator
cluster-kube-controller-manager-operator
cluster-kube-scheduler-operator
cluster-openshift-api-server-operator
cluster-openshift-controller-manager-operator
cluster-monitoring-operator
cluster-network-operator
cluster-dns-operator
cluster-ingress-operator
machine-api-operator
auth-config-operator

cluster-autoscaler-operator
cluster-node-tuning-operator
cluster-storage-operator
cluster-samples-operator
cluster-image-registry-operator
cluster-logging-operator
cluster-service-catalog-operator
console-operator
machine-config-operator
service-serving-cert-signer
olm-operator

Demo

What’s ahead

OPERATOR SDK & LIFECYCLE

Now

Operator Testability

SDK

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

Features:
- `scorecard` utility

Stage: Prototype

Cross-Platform Support

OLM

SDK

Objectives:
- OCP, OKD and k8s consistency

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

Stage: Development

Next

Continuous Testing

SDK

Objectives:
- automated scorecard testing
- validate operator maturity

Stage: Discovery

Partner Enablement

SDK

Objectives:
- UBI support for partners
- `scorecard` for partners

Stage: Discovery

No-Fuzz Operator Install

OLM

Features:
- `OperatorGroups`

Stage: Prototype

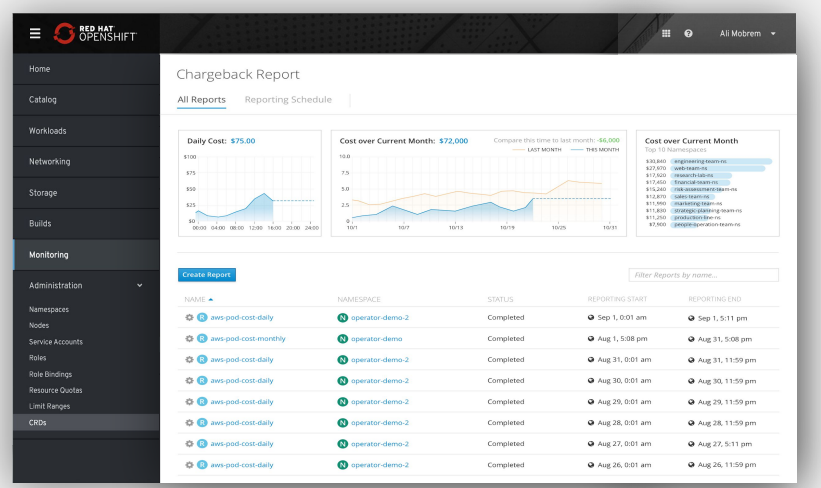
Future

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



CPU
Memory
Storage

✖

Request
Actual Usage
Utilization %

✖

Pod
Namespace
Node
Cluster

redhat.

Where to get started?

- https://github.com/operator-framework/getting-started
- https://github.com/operator-framework/community-operators
- https://commons.openshift.org/sig/operators.html
- #kubernetes-operators on the kubernetes slack
- https://groups.google.com/forum/#!forum/operator-framework
- Openshift Commons Briefing my Daniel Messer https://youtu.be/GgEKEYH9MMM

Thank you!