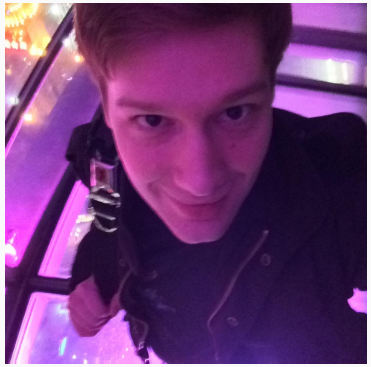
What’s New in Operator Framework

Jonathan Berkhahn, IBM
Jesús M. Rodríguez, Red Hat

Who are we?



Jonathan Berkhahn, IBM

@jberkhahn

Jesús M. Rodríguez, Red Hat

@jmrodri

Audience Poll

What is Operator Framework?

- Operator SDK
  - Simplifies building Kubernetes Operators
  - Has an extensible CLI & scaffolding
  - Uses upstream versions of `kubebuilder`, `controller-runtime` & `controller-tools`
- Operator Lifecycle Manager (OLM)
  - Helps install, manage, & upgrade Operators on a cluster
  - Provides over-the-air updates & catalogs
  - Offers a dependency model

What’s New?

Java Operators

- SDK can now scaffold out Java-based Operators

What are Java Operators?

- Operators written entirely in Java
- `java-operator-plugins` scaffold your Operator project
- Uses `java-operator-sdk` (a controller-runtime like library)
- Uses Quarkus for fast, native builds

Why Java Operators?

- Reach developers using most popular enterprise language
- Expand the language support of `operator-sdk`
  - Go, Ansible, Helm, & Java
- Native Java developer feel

Ooh what else?

Phase 2 Plug-ins

- What is a plug-in?
  - “ a software component that adds a specific feature to an existing computer program” - Wikipedia
  - Extends the scaffolding of `operator-sdk` commands
    - `init`: project initialization
    - `create api`: scaffolds Kubernetes API
    - `create webhooks`: scaffolds Kubernetes webhooks

Phases

- Phase 1
  - Made `operator-sdk` extensible
  - Supported only Go-based projects
  - Compiled as part of `operator-sdk`
- Phase 1.5
  - Introduces plug-in bundle
  - Allows chaining of plug-ins
  - Supports only Go-based projects (still)

Why Phase 2?

- Phase 2
  - Out-of-tree plug-ins do not have to be compiled into `operator-sdk`
  - Discover & run external executables as plug-ins
  - No longer restricted to Go
    - Python
    - Go
    - Java
    - Insert your favorite language here

But wait …

there’s more

Hybrid Helm

- Helm Operators
  - Easy to set up & running quickly
  - Limited functionality because everything is baked-in
- Hybrid Helm allows you to write Golang controller code in addition to standard Helm reconciliation.
- Helm plug-in repository contains:
  - A library with helper logic for interacting with Helm & the `controller-runtime`.
  - Helper functions to add Helm-abstractions like Helm pre/post hooks, overrides, etc.

Future Work

Future Work - SDK

- External bundle validators
  - Allow for calling validations from external binaries
  - Custom validation logic for your specific Operator
  - Allow validators to be version bumped independent of SDK version
- Support for external language plug-ins (like Python)
  - Phase 2 plug-ins will open up support for SDK plug-ins written in other languages
  - No longer have to be compiled in to SDK, can be installed separately

Future Work - OLM

- File-based catalogs
  - Plain text catalog (JSON or YAML) evolution
  - Fully backwards compatible with the previous `sqlite` catalog format
  - Supports declarative configuration
- Rukpak - OLM “2.0”
  - A pluggable, packaging and distribution solution
  - Small set of CRD APIs and controllers, based on the abstraction of a Bundle
    - Support for generic Bundles, i.e. Bundle backed by a Helm chart, etc.

Demo

Q&A

???

How to contact us

- Slack: #operator-sdk-dev, #kubernetes-operators, #olm-dev
- Mailing list: operator-framework Google Group
- Discord: #java-operator-sdk
- Community page: https://github.com/operator-framework/community/