Cluster-Logging Operator Review

Greetings,

This email is to touch base with you as an operator product team within Red Hat and pass along various guidelines and standards that are being defined for use within Red Hat. We (the Portfolio Enablement (PE) Team) spent some time reviewing the Cluster-Logging Operator operator and have the following findings.

Cluster-Logging Operator is one of the first teams the PE team is reaching out to, so we can schedule a conference call to go over this if it would be helpful.

Deprecated APIs (Priority)

We could check that the Cluster-Logging Operator is upgraded to use the latest api versions for CRDs (v1) and by looking into the image (registry.redhat.io/redhat/redhat-operator-index:v4.7) we can also check that your bundles are configured to be distributed from 4.6 by using OCP label index.

Actions: See the Upgrade Deprecated k8s APIs for CRD/Webhooks guide and ensure that your project is following its recommendations and you have no doubts about it. Also, we checked that this project has not been distributed prior 4.6. Could you please confirm that?

Disconnected Annotation

This operator is defined as configurable to work as disconnected in https://access.redhat.com/articles/4740011. We could identify that the biggest part of your operator bundles have the annotation already. However, see in the bundle's report generated for its package from 4.7 image that a few of them are not setting it:

```
operators.openshift.io/infrastructure-features: '["Disconnected"]'
```

The list will eventually be drawn in an automated way from your annotations and if you do not ensure that the annotation is set, your operator may not be included in this list anymore. For further information check the doc.

Actions: Ensure that all operator bundles which support this feature are configured with the disconnect annotation such as follows.

Channel Naming

We found channel names such as 4.6 to distribute your project and then, according to the channel naming doc it would be recommended to use the prefixes preview, fast and stable and not the OCP versions on it.

Actions: See the channel naming doc and check if you are able to start to work with the channels as described in this doc.

SDK Usage

It seems like Cluster-Logging Operator is using and was scaffolded with SDK 0.18.1 and its layout has deviated as well. Note that layout deviations might reduce the understandability and manuntendabilitty of your project.

Actions: Please review the SDK upgrade documentation, how to keep your project updatable, we feel your operator could benefit from the latest layout which uses kustomize as the helpers provided by the SDK tool to integrate your project with OLM and the latest scorecard features. Also, see that 0.18.1 is no longer supported. To check an example of the latest layout see the memcached sample in http://operator-sdk/testdata/go/v3/memcached-operator/

Scorecard Default Images

This project was checked against the default Scorecard image configuration which is added by default in the projects which are generated using SDK.

Actions: See the columns Scorecard Suggestions, Scorecard Errors, Scorecard Failing Tests in the bundle's report. By meeting these criteria you can bring a better representation of the operator on the OCP catalog.

These checks can be added manually by creating the test/scorecard directory and adding the default config scaffold by SDK. See an example here. Also, you will win that for free by adopting the SDK's latest layout.

Scorecard Custom Images

CVP will be enabling an ability to run functional tests for operators using the operator-sdk scorecard utility. Development teams can write their own functional tests as scorecard custom images and have CVP execute these tests. Both DPTP and CVP plan on being able to support custom scorecard test execution. To demonstrate this capability, we have developed an example scorecard custom image found here: https://github.com/operator-framework/tekton-scorecard-image

Actions: We did not find any scorecard tests within the Cluster-Logging Operator repo, but that is expected since most people are unaware they can write custom functional tests using Scorecard. It might be worth the Cluster-Logging Operator team reviewing the usefulness of implementing a custom scorecard test image given its new adoption by DPTP and CVP.

Has Multiple Architectures and Disconnected support

We checked that your project is supporting multiple architectures such as amd64 Ppc64le, s390x and also is supporting Disconnected Mode.

Actions: Could you please let us know if your users have been facing performance issues in the Disconnected Mode?

Operator De-Scoping

We checked that your bundles are not supporting install mode for all namespaces. Could you please let us know why you do not allow it? Could you share the impacted scenarios if the user would be able to install the solution for all namespaces?

Discussion Item: We'd like to discuss whether the work going on within OLM https://issues.redhat.com/browse/OLM-2159, to define descoped operators, can support Quay's requirements.

Best Practices

The common guidance can be found in the Best Practices sections in SDK and OLM public documentation.

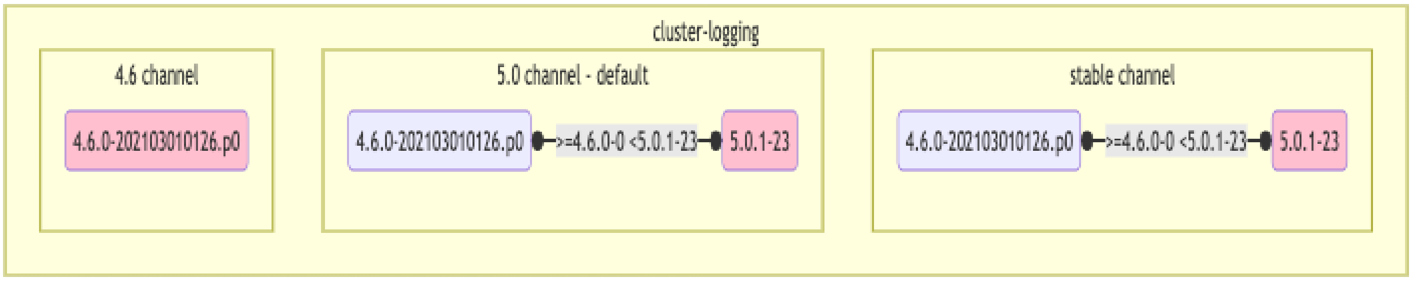
Note that the public documents do not address what might be specific and valid only for OpenShift. In this case, you can use the OCP docs (e.g.here, the partner docs (e.g.here) and the internal ones like this page.

Actions: Check out mainly Operator Best Practices and also the Common recommendations and suggestions for operators/SDK and OLM to ensure that your project has been following them. If you have a specific reason for not following these guidelines please don't hesitate to reach out.

We could check that your project has been using ginkgo and gomega which are the frameworks adopted by the k8s community for the tests and that is great. However, it seems that you have not been covering your controllers with tests using EnvTest. Then, we would like to suggest you check it since it could bring advantages and might reduce the need of other tests which can bring more cost to be executed.

Upgrade Graph

The upgrade graph for the Cluster-Logging Operator package in the Red Hat OpenShift 4.7 catalog looks like:



NOTE: *This graph was generated using the Graph Visualization Utility. To understand its legend see Channel Upgrade Graphs Legends.*

Action: The team might want to review the graph above to make sure it matches your expectation.

Next Steps

If you would find it useful, we would like to discuss the action items listed in this review with you at a convenient time and answer any questions you may have.

The PE team can help operator developer teams address issues or gaps found in OLM, SDK, and Operator Framework. We can perform reviews or answer questions as needed regarding Red Hat operators being developed.

Contact us via Slack at #forum-portfolio-enablement or via email in response to this email.

Thanks,

Camila Macedo and the Portfolio Enablement TeamAdditional Materials
These may just provide extra information or didn’t apply based on our review:

CFC Website

https://docs.engineering.redhat.com/display/CFC/Introduction

Hotfix Example

This is an example of building and distributing a hotfix for an Operator using OLM’s skipRange
https://docs.google.com/document/d/18UCaka2mGcRzvdhktvZi5cPLe29ssQ1WzUc4faxvDBA/edit#heading=h.88v8zffb5wvy

OLM Upgrade Graphing

This document defines how upgrade graphs can be depicted using the Mermaid graphing tooling
https://olm.operatorframework.io/docs/contribution-guidelines/upgrade-graphs/

https://docs.google.com/document/d/1N29w764eZroOywvLK1tb00XqX40A9yo4ghmSLE9PaMI/edit?usp=sharing

There are instructions in this repo to re-create the graph both from a copy of the index db, and also instructions on how to create it on-the-fly locally from your current local copy of the Quay repo.

Graph Visualization Utility

This tool lets you generate an upgrade graph based on your operator definitions, you can run this utility as you develop so that you can verify your upgrade graphs before you push a release.
https://github.com/operator-framework/index-mermaid-graph

Channel Versioning

This document discusses best practices on versioning of your operator images
https://docs.google.com/document/d/1X4xwBK4ECIXjaA_0DuVNuWieY9QjbI42BMRAyIYc_HM/edit#heading=h.4o2snu2vbv1v

Audit Tool

The PE team has written an audit tool that checks your operator against various best practices.

https://github.com/operator-framework/audit
Operator teams can run that tool themselves to check against the audit criteria.
Here is an example of running the audit command:

```
$ audit-tool bundles
--index-image=registry.redhat.io/redhat/redhat-operator-index:v4.7
--filter=cluster-logging
```