Google Summer of Code

PROPOSAL

Rewrite NFS Operator to use controller-runtime

By

Ahmad Nurus Shobah

CLOUD NATIVE
COMPUTING FOUNDATION

ROOK

Table of Contents

Abstract **3**

Background **3**

Deliverables **3**

Timeline 4

Issues 5

General Notes **5**

About me **5**

Personal Information And Contact Detail 5

Why me 6

1. Abstract

This project aims to implement controller-runtime for Rook nfs-operator and addresses several issues along with it. Currently, Rook nfs-operator only simply watches an event of CustomResource from an informer. Controller-runtime is a good library for building an operator which incorporates the best-practice controller principles.

2. Background

Rook is cloud-native storage orchestration that leverages Kubernetes operator pattern to provide automation tasks including deployment, bootstrapping, configuration, provisioning, scaling, upgrading and more. Before, Rook used operator-kit for implementing operator pattern. But due to operator-kit limitations in some areas such as controller watchers, it began to be abandoned.

Currently, Rook nfs-operator only simply watches an event of CustomResource from an informer. Controller-runtime is widely used for writing Kubernetes operators. It is also leveraged by Kubebuilder and Operator SDK. Controller-runtime has a lot of features that are responsible for an operator and its controllers. By providing a higher level controller function called `Reconcile()` made it also much simpler and easy to understand.

This project proposal aims to implement controller-runtime for nfs-operator so it can improve the reliability of the operator. Besides that, it will also fix several issues related to the NFS Rook.

3. Deliverables

There are five milestones for successfully completing this project:

1. Implementing controller-runtime in nfs-operator
2. Implementing for NFS CRDs configuration validation
3. Implementing persistent volume deletion in nfs-provisioner
4. Addressing NFS server memory leak
5. Updating unit-test and integration-test

Point 4 above (addressing NFS server memory leak) is optional and will be completed if time permits.

During the community bonding period I’m planning to investigate best practice to implement controller-runtime for Rook nfs-operator controller and creating draft for NFS CRDs configuration validation and feature for persistent volume deletion in nfs-provisioner.

First coding phase I'll implement controller-runtime for nfs-operator controllers and investigate other issues along with Rook NFS. So in the first evaluation, we already implement controller-runtime for nfs-operator and have solutions for those issues.

For the second coding phase I'll implement the solutions to fix those issues. So in the second evaluation the other issues along with Rook NFS are already addressed.

In the last coding phase, the goal is to have controller-runtime in nfs-operator and the other issues already fixed. Also updating unit-test and integration-test to make sure this project works as expected.

3.1. Timeline

Date	Task
May 5	Community Bonding period begins
May 5 - June 2	Investigate best practice to implement controller-runtime for Rook nfs-operator controller
May 5 - June 2	Draft NFS CRDs configuration validation
May 5 - June 2	Draft persistent volume deletion in nfs-provisioner feature
June 2	Community Bonding period ends
June 2 - June 25	Implement controller-runtime for nfs-operator
June 10 - June 29	Investigate and discuss several issue related Rook NFS
June 29 - July 3	First Evaluation
July 3 - July 14	Implement NFS CRDs configuration validation
July 15 - July 24	Implement persistent volume deletion in nfs-provisioner
July 27 - July 31	Second Evaluation
July 31 - August 14	Updating unit-test and integration-test
August 15 - August 24	Submit final code and project summaries
August 24 - August 31	Final Evaluation

3.2. Issues

Main issue the project

- https://github.com/rook/rook/issues/4950

Other open issues along with it

- https://github.com/rook/rook/issues/3073
- https://github.com/rook/rook/issues/3074
- https://github.com/rook/rook/issues/2721
- https://github.com/rook/rook/issues/4259

4. General Notes

- I will be available to work full time (a minimum of 40 hours per week) during the GSoC period.
- I will attend Rook community meeting
- I will create a GitHub repository that contains notes and information about the progress.
- I will create a google doc for tracking the progress daily. And keep informing my mentors about the progress

5. About me

5.1. Personal Information And Contact Detail

Name: Ahmad Nurus Shobah
Email: ahmadnurus.sh@gmail.com
Slack nick: @prksu
Github username: @prksu
Twitter: @ahmadnurus
University: Universitas AMIKOM Yogyakarta
Timezone: GMT+7 (Indonesia)

5.2. Why me

I have been working with Kubernetes for the past 1 year. I have experience with developing and deploying applications in Kubernetes. I also have experience provisioning Kubernetes clusters to some cloud providers, including AWS, GCP and DigitalOcean.

I am also intrigued with the Kubernetes *Operator pattern.* I've done writing my own simple operator https://github.com/fossildev/ghost-operator that using Operator-SDK. At this time, I'm focused on helping cluster-api projects, especially on DigitalOcean provider. I've done migrating cluster-api-provider-digitalocean api types from v1alpha1 to v1alpha2.