

Proposal

**Add Network File System
(NFS) as a Rook storage
backend**

By Rohan Raj Gupta

Table of contents:

1. Abstract
2. Why NFS
3. Why this project?
4. Related work
 - 4.1. nfs-provisioner
 - 4.2. Quartermaster NFS Driver
5. Proposed Deliverables
 - 5.1. Community bonding
 - 5.2. Coding Phase
 - 5.2.1. Phase 1
 - 5.2.2. Phase 2
 - 5.2.3. Phase 3
6. Commitment
7. About Me
 - 7.1. About Me
 - 7.2. Personal Information and contact details

1. Abstract

Rook is an open source orchestrator for distributed storage systems running in kubernetes, currently in alpha state and has focused initially on orchestrating Ceph on top of Kubernetes. There is no option for Network File System (NFS) yet.

This project aims to add NFS as another storage backend.

2. Why NFS

NFS is widely used for persistent storage in kubernetes cluster. Using NFS storage is a convenient and easy way to provision storage for applications.

- An NFS volume allows an existing NFS (Network File System) share to be mounted into the pod.
- The contents of an NFS volume are preserved and the volume is merely unmounted if the pod is stopped/destroyed. This means that an NFS volume can be pre-populated with data, and that data can be “handed off” between pods.
- NFS supports multiple read/write simultaneously so a single share can be attached to multiple pods .

3. Why this project?

Rook brings storage platforms to Kubernetes with a fully Kubernetes-native design. It integrates deeply into cloud native environments leveraging extension points and providing a seamless experience for scheduling, lifecycle management, resource management, security, monitoring, and user experience.

With this project Rook is providing another widely adopted storage option for admins and users of cloud-native environments. This project will automate NFS starting from its configuration (such as allowed hosts, read/write permissions etc.) to deployment and provisioning. The operations on NFS which cannot be done natively by Kubernetes will be automated.

NFS doesn't provide an internal provisioner for kubernetes, so Rook is needed as an external provisioner (Other external provisioners are compared with rook in the Related works section below).

4. Related Works

4.1. nfs-provisioner

(<https://github.com/kubernetes-incubator/external-storage/tree/master/nfs>)

nfs-provisioner is an out-of-tree dynamic provisioner for Kubernetes 1.4, which can be used to quickly & easily deploy shared storage that works almost anywhere.

But this lacks advanced host management and configuration like different access permissions for different hosts etc. Rook will overcome this to enhance the security.

4.2. Quartermaster NFS Driver

(<https://github.com/coreos/quartermaster/blob/master/pkg/storage/nfs/README.md>)

Quartermaster is a framework for managing *containerized storage systems* like Ceph, GlusterFS, NFS-Ganesha, Rook and others on top of Kubernetes. Quartermaster enables the deployment, installation, and integration of these type of storage systems onto a Kubernetes cluster. Quartermaster abstracts this complexity and presents the client with a simple storage deployment model which is fully integrated with Kubernetes. By simplifying the deployment of storage systems, Quartermaster makes it possible to easily and reliably deploy, upgrade, and get status of a desired storage system in a Kubernetes cluster. Quartermaster does not provision storage. Instead, Quartermaster deploys and initializes the storage so that it then can be provisioned.

Quartermaster's NFS Driver is in experimental stage and not for production.

As mentioned in section 3, rook is also providing provisioning and much more.

5. Proposed Deliverables

5.1 Community bonding (April 23 - May 14)

- Join the Slack community
- Attend one of the community meetings
- Get development environment and tools set up
- Build local changes to produce runnable artifacts

- Run locally built code in Minikube environment
- Run and pass unit and integration tests
- Submit a pull request, integrate feedback and have it approved and merged to master

5.2 Coding Phase (May 14 - August 6)

5.2.1 Phase 1 (May 14 - June 15):

Design document 1-pager (2-4 weeks)

- Investigate related technology such as Kubernetes, NFS servers, volume provisioning, etc.
- Identify full scope and requirements of feature
- Prototype as needed to explore potential implementation choices and trade-offs
- Write detailed design document that specifies how feature will be implemented and tested
- Submit a pull request for the design, integrate feedback and have it approved and merged to master

Coding implementation (8-10 weeks)

User facing experience (1 week)

- New types, custom resources, and config options to capture the full experience for a user to setup and configure NFS storage with Rook

5.2.2 Phase 2 (June 16 - July 13)

Dynamic provisioner and/or orchestration for running NFS server (2-4 weeks)

- Implement the designed components for dynamic provision of NFS volumes for pod consumption, potentially an addition to the existing Rook dynamic provisioner and/or a custom controller to orchestrate deployment of NFS server components.
- Service creation for external access of storage.

5.2.3 Phase 3 (July 14 - August 6)

NFS server component (2-3 weeks)

- Implement an NFS server that will use the provided backing storage (host path, persistent volumes, CephFS) and handle NFS client requests for storage operations via the NFS protocol

Integration tests (2 weeks)

- Author an end to end integration test that deploys and consumes NFS storage from Rook to verify applications can consume this storage and prevent regressions to the this functionality in the future

6. Commitment

- I will be available to work full time (a minimum of 40 hours per week) during GSoC period.
- Will create a weekly report and publish it in blog.
- Will create a google doc and try to update it daily to keep track of daily tasks completed and problems faced during if any. This will help mentors to keep track of progress.
- Will attend community meetings.
- My time zone is IST (Indian Standard Time GMT +5:30) but i am flexible with timings which suits my mentor the best.

7. About Me

7.1 About Me

I am Rohan Raj Gupta a 4th year computer science engineering undergrad at University of Engineering and Management, Jaipur. I love technology and am constantly learning new things.

I have knowledge of linux system administration and i know C, Golang and Python. I have learned the concepts of Docker(namespaces, other types of containers like LXC, rocket, docker file, networking etc) and kubernetes(pods, deployments, PV, PVC, storage classes etc) by going through official documentation and other resources and have also tried out the examples to learn how to use.

This is my first time applying for GSoC and Rook is going to be the first open source project that i will contribute to. So i am very much excited and motivated from the fact that my contributions through this project is going to benefit a lots of users.

I am in touch with the mentors of this project on slack since last few days and they are very much supportive. I have set up the development environment for Rook and have run and tested the examples on Minikube environment to get familiar with Rook. I have also started working on an issue which is helping me to know more about Rook and kubernetes.

7.2 Personal Information and contact details

Name: Rohan Raj Gupta

Email: rohanrgupta1996@gmail.com

Slack nick: rohan47

Github: <https://github.com/rohan47>

University: University of Engineering and Management, Jaipur

Twitter: <https://twitter.com/rohanrajgupta>

gpg : 49C9 7EE3 A35C 2292 CF5D 63A0 0B35 B736 DD03 F4FA
(<https://keybase.io/rohan47>)