PolyFS - The Easiest to Use Distributed Filesystem

Katharos Technology

ABSTRACT: In the software community today, there is a shocking lack of easy-to-use distributed filesystems. Shared storage is crucial to any business beond the scale of single server, and a distributed filesystem is the most universal way to enable shared storage. PolyFS seeks to close the gap in shared filesystems and provide a solution that is easy to use, manage, and deploy while at the same time providing top-of-the-line performance. PolyFS, while not being limited to any particular application or usage, will be designed to include a Docker deployment and volume plugin and will provide an end-to-end solution for distributed volumes hosted by and for Docker Swarm.

NOTE: This is a work-in-progress design document outlining the plan for the design of PolyFS. This document will be updated as we discover more and change the design. The architecture in outlined in this paper has not been tested as of yet.

Introduction

As a DevOps engineer with experience running stateful containers in the Cloud with Docker Swarm, I am well acquainted with the need for a distributed Docker storage solution. When running applications on a Docker cluster, you have to have a way to make sure that each application can get to its persisted data. Unlike running an app on a single server, you cannot use the local disc for storage because you never know which server a particular container may start up on. You could constrain your app to run only on a specific server, but that eliminates much of the advantages of having the Docker cluster in the first place. You want to be able to view your cluster as a pool of resources that you can run your applications in, without having to worry about where those resources are comming from.

The need for shared storage is not limited to Docker. Some applications *require* that you provide some form of a shared filesystem in order to cluster the application. The easiest solution to this is NFS, a protocol by which you can host network share for a filesystem. You simply run the NFS server on the host that will provide the storage and you can mount that filesystem over the network on any number of other hosts. This is not without its problems, though, the largest of which being that there is only a single point of failure. If the server hosting the NFS share goes down, you lose access to your data. This makes NFS completely unsuitable for supporting clustered applications. There is no point to supporting your clustered application with a fileystem that is itself not clustered.

Figure 1: This is a demonstration figure.