Topic:

Boosting CephFS Security in Kubernetes: Rook's Intelligent Network Fencing for Uninterrupted Data Flow and Workload Harmony!

Presenter:
Niels de Vos

The CephFS Overview

CephFS (Ceph File System) is a distributed file system that is part of the Ceph storage system.

It is an open-source software-defined storage platform designed to provide scalable and high-performance storage for modern data needs.

CephFS allows you to organize and store files in a distributed manner across multiple nodes, providing a unified storage system accessible by clients through standard file system interfaces such as POSIX.

Challenge

A node can go down!

Imagine this: when one node takes an unexpected turn, kubernetes elegantly guides the pods to a new, stable platform and the pods get rescheduled on another node BUT what about the containers using the volume running on malfunctioned node?

Here's the twist – If a node goes down where a pod is running, and a cephfs RWX volume is mounted, the volume can automatically be mounted on another node.

CephFS remains undisturbed, and the container using this may happily continue, leading to multiple writers and causing data corruption!

Resolution

NetworkFence

NetworkFence is a cluster-scoped custom resource (CR) that allows kubernetes to invoke “Network Fence” operation on a storage provider.

The user needs to specify the list of CIDR blocks on which network fencing operation will be applied, along with the CSI driver name.

The creation of NetworkFence CR will add a Network Fence, and it’s deletion will undo the operation.

Sample

```yaml
apiVersion: csiaddons.openshift.io/v1alpha1
kind: NetworkFence
metadata:
 name: network-fence-sample
spec:
 driver: rook-ceph.cephfs.csi.ceph.com
 cidrs:
   - 10.90.89.66/32
   - 11.67.12.42/24
 secret:
   name: rook-csi-cephfs-provisioner
   namespace: rook-ceph
 parameters:
   key: value
```

Rook's Network Fencing : Guardian of your data

How it works?

Step 1: If a node is confirmed to be down, a taint is added to the node.

$ kubectl taint nodes <node-name> node.kubernetes.io/out-of-service=nodeshutdown:NoExecute

$ kubectl taint nodes <node-name> node.kubernetes.io/out-of-service=nodeshutdown:NoSchedule

Continued..

Step 2 -

Rook will automatically blocklist the node then to prevent connections to Ceph from the CephFS volume on that node by creating a Network Fence CR.

This can be verified by:

$ kubectl get networkfences.csiaddons.openshift.io

NAME DRIVER CIDRS FENCESTATE AGE RESULT

minikube-m02 rook-ceph.cephfs.csi.ceph.com ["192.168.39.187:0/32"] Fenced 20s Succeeded

The node is blocklisted if the state is *Fenced* and the result is *Succeeded*.

Ceph-CSI's Approach to Network Fencing: Evicting Clients for Enhanced Data Consistency

Evicting a CephFS client prevents it from communicating further with MDS daemons and OSD daemons. If a client was doing buffered IO to the file system, any un-flushed data will be lost.

The client eviction process applies to clients of all kinds, which includes FUSE mounts, kernel mounts, and any process using libcephfs.

Client Eviction

The Network Fence CR created by Rook, contains the CIDRs that needs to be blocklisted.

Based on those CIDRs, CephCSI evicts the active clients of that IP Range by using a ceph command -

```
bash-4.4$ ceph tell mds.0 client evict id=4305
```

This can be verified by :

```
bash-4.4$ ceph osd blocklist ls
100.64.0.5:0/2463791145 2023-12-18T10:27:19.567161+0000
```

Rook and CephCSI: A Unified Shield for Automated Network Fencing.

This is how Rook's Network Fencing becomes the guardian of your data, ensuring a smooth transition from one node to another. It's not just about preventing chaos; it's about preserving the integrity of CephFS, the backbone of your containerized world.

Thank you..!!

Let's open the floor for questions and discussions.

-Your interest in our presentation is greatly appreciated.