Open Infrastructure - Task #9487

Migrate our ceph clusters into rook

07/08/2021 11:14 PM - Nico Schottelius

Status:	Rejected	**Start date:**	07/08/2021
Priority:	Normal	**Due date:**	
Assignee:	Nico Schottelius	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
PM Check date:			

Description

Situation

- At ungleich we are running multiple ceph cluster, usually one or more per location. Currently our ceph clusters run on Devuan (w/o systemd).
- Alpine Linux, which we use in combination with Devuan, does not have proper ceph packages.
- Ceph is moving away from packages in favor for containers managed by cephadm, which requires systemd on the host OS.
- We are not inclined to switch to systemd, due to long term maintenance issues we have experienced
- The alternative to cephadm is using ceph with https://rook.io
- We have successfully tested Alpine Linux + crio or docker + rook on test clusters

Objectives

- Have proper operational understanding of rook (vs. "plain ceph") by running multiple test clusters, upgrade, fail, destroy and rebuild them (handled in internal ticket #4608)
- Develop a strategy to migrate our native Ceph clusters into rook without any downtime

High level Strategy per cluster

- Create a k8s cluster
- Adjust the rook settings to match the current cluster (fsid, monitors, networking, etc.)
- Controlled phase in services in rook
- Retire / replace native running services
- Repeat for each cluster

Operational steps / detailed plan

- Setup k8s cluster
  - According to https://code.ungleich.ch/ungleich-public/ungleich-k8s
  - Missing bits: prometheus/monitoring/alerting without ingress
- Rook settings:
  - Networking comes from the k8s cluster itself
    - Need to add BGP peering support for storage cluster networks
  - Tunings/adoptions according to https://rook.io/docs/rook/master/ceph-disaster-recovery.html
- Rook deployment
  - Via Flux/Helmrelease (?)

(TBC)

Rook layout / purpose

- Rook usually uses CSI/is responsible for k8s PV/PVC deployment
- In the cases described in this ticket the primary consumer are external VMs
  - No PVCs need to be managed/created

(TBC)

Staged phase in

It might be possible to phase in rook as follows:

- 1) additional managers (you can't have too many of them)
- 2) remove outside managers (shutdown, delete monit jobs)
- 3) additional monitors
  - Use an EVEN number of additional monitors (f.i. 2 or 4) to stay odd in total
- 4) When stable: remove 1 outside monitor, raise number to odd number in rook (likely to 5)
- 5) When stable: switch to rook only monitors, remove all outside monitors

At this point all monitors and managers are controlled by rook. Then we will migrate OSDs:

- Add some new nodes to the cluster with additional storage, as a k8s worker
  - specify these nodes in rook for device discovery
  - Ensure device classes are properly set in a test cluster before
    - We use "hdd-big" for 3.5" HDD and "ssd" for ssds
      - Depending on the controller, the rotational kernel flag might not be exposed, we need to tune / plan ahead on how to tell rook which host serves which classes

If everything is good so far, we have additional storage nodes running inside rook. Finally, migrating the native nodes to rook:

- Set cluster to noout, preventing rebalance
- Shutdown/delete a native node (w/o purging the OSDs)
  - Verify disk format is compatible with rook -> we use plain disks, non LVM
    - If not compatible, maybe replace all OSDs with LVM based beforehand
      - Maybe 1-2 OSDs/day
- Reboot with Alpine + k8s worker
  - rook should detect existing osds and re-add them
    - Need to test this behaviour on test clusters
- Unset noout
- Repeat for all native nodes

History

#1 - 07/08/2021 11:15 PM - Nico Schottelius

- Description updated

#2 - 07/08/2021 11:34 PM - Nico Schottelius

- Description updated

#3 - 07/08/2021 11:35 PM - Nico Schottelius

- Description updated

#4 - 01/02/2024 02:08 PM - Nico Schottelius

- Status changed from In Progress to Rejected

We phase in new clusters and use the workload inside k8s, mixing is non-trivial due to changing monitor names and addresses.