hswaw - Bugless #6

k0: move ceph-waw3 to static Ceph deployment

02/07/2021 10:31 PM - q3k

Status:	Accepted	
Priority:	Urgent	
Assignee:	q3k	
Category:	hscloud	

Description

Currently we deploy ceph-waw3 via Rook. This caused us a bunch of runaway automation outages. We should investigate moving over the configuration of the cluster (mons, osds, mgr) to be managed with plain NixOS instead.

Moving over ceph-waw3 might be difficult, so this could end up becoming a ceph-waw4... This is to be figured out.

Related issues:

Blocks hswaw - Bugless #10: k0: productionize and make people on call for it	**New**

History

#1 - 02/08/2021 10:12 AM - q3k

- Blocks Bugless #10: k0: productionize and make people on call for it added

#2 - 03/23/2021 12:07 PM - q3k

- Status changed from New to Accepted

- Assignee set to q3k

- Priority changed from Normal to Urgent

We just had yet another outage caused by this dumpster fire of a software.

#3 - 03/23/2021 12:22 PM - q3k

outage tl;dr:

- 10:43ish: q3k woke up to rook having deleted all mons, again, and a bunch of secrets (realized this because i wanted to restart valheim, which then complained about a missing configmap/secret in the rook agent)
- 10:48ish: q3k writes on #hackerspace-pl-staff that rook is fucked again
- recovery: q3k scales down operator, new mon (mon-a), copies all mon data over to workstation, rebuilds a new monmap, applies it to a fairly recent mon data dir from one of the deleted mons, copies it over to mon-a
- recovery: q3k rewrites secrets/configmaps in ceph-waw3 to have old credentials, fsid and single mon (admin credentials recovered from toolbox, new mon credentials created with ceph auth)
- recovery: q3k restarts mon-a with new data, mon starts up, but has new address - rolling restart required to re-point kernel rbd maps into new ip
- recovery: q3k restarts all osds so they talk to new mon ip, mgr recovery, ceph says HEALTH_OK
- recovery: q3k rolling-restarts all nodes so that they mount rbds against new mon svc ip
- recovery: q3k attempts to scale mons back up to three from one, rook fails to bring up consensus for second mon, scaled back down to one mon for now, we want to get rid of rook anyway
- 12:30ish: most k0 services back up, some stragglers, eg. missing s3 secrets (did rook delete them???), typical kubelet data mount timeouts on matrix/synapse-media-0, some pods stuck in unknown after node restart without drain, etc
- 13:10: full recovery

#4 - 09/10/2021 11:00 PM - q3k

Started work on this.

First step, deployed a Ceph cluster on k0 via NixOS: https://gerrit.hackerspace.pl/c/hscloud/+/1084

This has a mon on bc01n02, and OSDs on dcr01s{22,24} (running on new disks, also with dmcrypt!).

Mons will be moved to bc01n{05,06,07} once these are up - waiting for SSDs and initial provisioning.

In the meantime, I'll look into possible migration paths from ceph-waw3, and what exactly is needed to let Rook provision PVs and RGW users for this cluster.

#5 - 09/11/2021 11:15 PM - q3k

I've upgraded Rook to v1.6 so that it can work wit hour new NixOS Ceph. https://gerrit.hackerspace.pl/c/hscloud/+/1090

I'm now considering Bumping ceph-waw3 to Ceph 16 too, and then using RGW multi-site support to migrate over all ceph-waw3 data into ceph-k0. This would allow us to move over S3 data without downtime, maintaining all the old user/bucket/metadata, I think. Looking into it.

#6 - 09/12/2021 12:51 AM - q3k

Started upgrading ceph-waw3 to Ceph 15 first (from Ceph 14), hit some BlueFS/Bluestore/RocksDB corruption...

```
$ kubectl -n ceph-waw3 get deployment -l rook_cluster=ceph-waw3 -o jsonpath='{range .items[*]}{"ceph-version="
}{.metadata.name}: {.metadata.labels.ceph-version}{"\n"}{end}'
ceph-version=rook-ceph-crashcollector-bc01n01.hswaw.net: 15.2.13-0
ceph-version=rook-ceph-crashcollector-dcr01s22.hswaw.net: 15.2.13-0
ceph-version=rook-ceph-crashcollector-dcr01s24.hswaw.net: 15.2.13-0
ceph-version=rook-ceph-mgr-a: 15.2.13-0
ceph-version=rook-ceph-mon-a: 15.2.13-0
ceph-version=rook-ceph-osd-0: 14.2.16-0
ceph-version=rook-ceph-osd-1: 15.2.13-0
ceph-version=rook-ceph-osd-2: 15.2.13-0
ceph-version=rook-ceph-osd-3: 14.2.16-0
ceph-version=rook-ceph-osd-4: 14.2.16-0
ceph-version=rook-ceph-osd-5: 14.2.16-0
ceph-version=rook-ceph-osd-6: 15.2.13-0
ceph-version=rook-ceph-osd-7: 14.2.16-0
ceph-version=rook-ceph-rgw-waw-hdd-redundant-3-object-a: 15.2.13-0
```

So mon, mgr, are at 15. osd.{1,2,6} are at 15. All other osds are at 14.

During its upgrade, osd.6 started crashlooping:

```
debug 2021-09-12T00:26:41.687+0000 7f91f980af00  4 rocksdb: EVENT_LOG_v1 {"time_micros": 1631406401688599, "jo
b": 1, "event": "recovery_started", "log_files": [67947]}
debug 2021-09-12T00:26:41.687+0000 7f91f980af00  4 rocksdb: [db/db_impl_open.cc:583] Recovering log #67947 mod
e 0
debug 2021-09-12T00:26:41.997+0000 7f91f980af00  3 rocksdb: [db/db_impl_open.cc:518] db.wal/067947.log: droppi
ng 1182006 bytes; Corruption: WriteBatch has wrong count
debug 2021-09-12T00:26:41.997+0000 7f91f980af00  4 rocksdb: [db/db_impl.cc:390] Shutdown: canceling all backgr
ound work
debug 2021-09-12T00:26:41.997+0000 7f91f980af00  4 rocksdb: [db/db_impl.cc:563] Shutdown complete
debug 2021-09-12T00:26:41.997+0000 7f91f980af00 -1 rocksdb: Corruption: WriteBatch has wrong count
debug 2021-09-12T00:26:41.997+0000 7f91f980af00 -1 bluestore(/var/lib/ceph/osd/ceph-6) _open_db erroring openi
ng db:
debug 2021-09-12T00:26:41.997+0000 7f91f980af00  1 bluefs umount
debug 2021-09-12T00:26:41.998+0000 7f91f980af00  1 bdev(0x55ad030a4380 /var/lib/ceph/osd/ceph-6/block) close
debug 2021-09-12T00:26:42.132+0000 7f91f980af00  1 bdev(0x55ad030a4000 /var/lib/ceph/osd/ceph-6/block) close
debug 2021-09-12T00:26:42.399+0000 7f91f980af00 -1 osd.6 0 OSD:init: unable to mount object store
debug 2021-09-12T00:26:42.399+0000 7f91f980af00 -1  ** ERROR: osd init failed: (5) Input/output error
```

This seems to be a case of https://lists.ceph.io/hyperkitty/list/ceph-users@ceph.io/thread/6UIPGV2OSPBGKQLV2IDNJAYYCPABYPZI/?sort=date .

Since we seem to have enough redundancy (we should, other than the yolo pool which as designed has no redundancy), I've just taken osd.6 out and will let recovery do it's thing.

```
[root@bc01n02 /]# ceph -w
  cluster:
    id:     ea847d45-da0b-4be0-8c77-2c2db021aaa0
    health: HEALTH_WARN
            client is using insecure global_id reclaim
            mon is allowing insecure global_id reclaim
            Degraded data redundancy: 187331/1605124 objects degraded (11.671%), 69 pgs degraded, 69 pgs under
sized
            3 pools have too few placement groups
            6 pools have too many placement groups
            1 daemons have recently crashed

  services:
    mon: 1 daemons, quorum a (age 30m)
    mgr: a(active, since 30m)
    osd: 8 osds: 7 up (since 26m), 7 in (since 16m); 68 remapped pgs
    rgw: 1 daemon active (waw.hdd.redundant.3.object.a)

  task status:

  data:
    pools:   14 pools, 665 pgs
    objects: 802.56k objects, 2.3 TiB
    usage:   4.2 TiB used, 34 TiB / 38 TiB avail
```

```
    pgs:     187331/1605124 objects degraded (11.671%)
             595 active+clean
             59  active+undersized+degraded+remapped+backfill_wait
             9   active+undersized+degraded+remapped+backfilling
             1   active+clean+scrubbing+deep+repair
             1   active+undersized+degraded

  io:
    client:   88 KiB/s rd, 972 KiB/s wr, 19 op/s rd, 47 op/s wr
    recovery: 105 MiB/s, 31 objects/s

2021-09-12 00:46:30.373555 mon.a [WRN] Health check update: Degraded data redundancy: 187432/1605124 objects d
egraded (11.677%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:46:35.374835 mon.a [WRN] Health check update: Degraded data redundancy: 187201/1605124 objects d
egraded (11.663%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:46:40.500101 mon.a [WRN] Health check update: Degraded data redundancy: 187073/1605136 objects d
egraded (11.655%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:46:46.394575 mon.a [WRN] Health check update: Degraded data redundancy: 186943/1605138 objects d
egraded (11.647%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:46:54.465108 mon.a [WRN] Health check update: Degraded data redundancy: 186740/1605172 objects d
egraded (11.634%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:47:00.381099 mon.a [WRN] Health check update: Degraded data redundancy: 186706/1605178 objects d
egraded (11.631%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:47:05.382316 mon.a [WRN] Health check update: Degraded data redundancy: 186475/1605178 objects d
egraded (11.617%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:47:10.383506 mon.a [WRN] Health check update: Degraded data redundancy: 186441/1605178 objects d
egraded (11.615%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:47:15.384851 mon.a [WRN] Health check update: Degraded data redundancy: 186246/1605180 objects d
egraded (11.603%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:47:20.386380 mon.a [WRN] Health check update: Degraded data redundancy: 186178/1605180 objects d
egraded (11.599%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:47:25.387964 mon.a [WRN] Health check update: Degraded data redundancy: 185989/1605188 objects d
egraded (11.587%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
2021-09-12 00:47:30.389543 mon.a [WRN] Health check update: Degraded data redundancy: 185918/1605192 objects d
egraded (11.582%), 69 pgs degraded, 69 pgs undersized (PG_DEGRADED)
```

I've also scaled down osd.6 and the operator on k8s to pause the upgrade. Once the backfill/recovery settles, I'll up the yolo pool for larger redundancy and look into setting `bluestore_fsck_quick_fix_on_mount` to false before resuming the update. That will hopefully let us safely finish the upgrade to 15.

```
[root@bc01n02 /]# ceph config get osd bluestore_fsck_quick_fix_threads
2
[root@bc01n02 /]# ceph config get osd bluestore_fsck_quick_fix_on_mount
true
```

#7 - 09/12/2021 09:48 AM - q3k

After setting ceph osd pool set device_health_metrics size 2 (was 3 and was likely newly created during a ceph device health run) everything reshuffled into active+clean. I'm now following pg autoscale hints and updating pg_num on pools to appease it.

That's gonna take a while. After that, I'm gonna run some scrubs and bluestore fscks on the cluster, then continue the upgrade to 15.

#8 - 09/12/2021 11:14 AM - q3k

Things got rebalanced, now a few PGs are scrubbing+deep+repair which is slightly concerning, but let's see how it goes:

```
[root@bc01n02 /]# ceph -s
  cluster:
    id:     ea847d45-da0b-4be0-8c77-2c2db021aaa0
    health: HEALTH_WARN
            client is using insecure global_id reclaim
            mon is allowing insecure global_id reclaim
            6 pools have too many placement groups

  services:
    mon: 1 daemons, quorum a (age 10h)
    mgr: a(active, since 10h)
    osd: 8 osds: 7 up (since 10h), 7 in (since 10h)
    rgw: 1 daemon active (waw.hdd.redundant.3.object.a)

  task status:

  data:
    pools:   14 pools, 737 pgs
    objects: 806.04k objects, 2.3 TiB
```

```
    usage:   4.7 TiB used, 33 TiB / 38 TiB avail
    pgs:     734 active+clean
             3   active+clean+scrubbing+deep+repair

  io:
    client:   58 KiB/s rd, 851 KiB/s wr, 3 op/s rd, 53 op/s wr
```

#9 - 09/12/2021 11:28 AM - q3k

Seems like the active+clean+scrubbing+deep+repair PGs are just periodic scrubbing, as the 3 active ones cleared up and then more got triggered.

Going ahead and updating the pg_nums on the 6 other pools that have too many of them, to appease the autotuner:

```
[root@bc01n02 /]# ceph osd pool autoscale-status
POOL                                     SIZE  TARGET SIZE  RATE  RAW CAPACITY   RATIO  TARGET RATIO
EFFECTIVE RATIO  BIAS  PG_NUM  NEW PG_NUM  AUTOSCALE
waw-hdd-redundant-3-object.rgw.control      0                 2.0        39123G  0.0000
                 1.0      64           8  warn
waw-hdd-redundant-3-object.rgw.meta       19221               2.0        39123G  0.0000
                 1.0      64           8  warn
waw-hdd-redundant-3-object.rgw.log         6719k              2.0        39123G  0.0000
                 1.0      64           8  warn
waw-hdd-redundant-3-object.rgw.buckets.index   37161k        2.0        39123G  0.0000
                 1.0      64           8  warn
.rgw.root                                  5930               2.0        39123G  0.0000
                 1.0      64           8  warn
waw-hdd-redundant-3-object.rgw.buckets.data     1403G         2.0        39123G  0.0717
                 1.0      64              warn
waw-hdd-redundant-3                        668.1G             2.0        39123G  0.0342
                 1.0      64              warn
waw-hdd-redundant-3-metadata               23358              2.0        39123G  0.0000
                 1.0      64              warn
waw-hdd-redundant-3-object.rgw.buckets.non-ec  138.4k         2.0        39123G  0.0000
                 1.0      64           8  warn
q3k-test                                     19               2.0        39123G  0.0000
                 1.0      64              warn
waw-hdd-redundant-q3k-3                    601.7G             2.0        39123G  0.0308
                 1.0      32              warn
waw-hdd-redundant-q3k-3-metadata            0                 2.0        39123G  0.0000
                 1.0      32              warn
waw-hdd-yolo-3                              0                 1.5        39123G  0.0000
                 1.0      32              warn
device_health_metrics                       0                 2.0        39123G  0.0000
                 1.0       1              on
[root@bc01n02 /]# for pool in waw-hdd-redundant-3-object.rgw.control waw-hdd-redundant-3-object.rgw.meta waw-h
dd-redundant-3-object.rgw.log waw-hdd-redundant-3-object.rgw.buckets.index .rgw.root waw-hdd-redundant-3-objec
t.rgw.buckets.non-ec; do ceph osd pool set $pool pg_num 8; done
set pool 2 pg_num to 8
set pool 4 pg_num to 8
set pool 6 pg_num to 8
set pool 8 pg_num to 8
set pool 9 pg_num to 8
set pool 15 pg_num to 8
```

#10 - 09/12/2021 11:32 AM - q3k

Done, now waiting for pgp_num/pg_num to go down from 64 to 8 as requested:

```
[root@bc01n02 /]# for pool in waw-hdd-redundant-3-object.rgw.control waw-hdd-redundant-3-object.rgw.meta waw-h
dd-redundant-3-object.rgw.log waw-hdd-redundant-3-object.rgw.buckets.index .rgw.root waw-hdd-redundant-3-objec
t.rgw.buckets.non-ec; do echo "$pool $(ceph osd pool get $pool pg_num) $(ceph osd pool get $pool pgp_num)"; do
ne
waw-hdd-redundant-3-object.rgw.control pg_num: 44 pgp_num: 44
waw-hdd-redundant-3-object.rgw.meta pg_num: 48 pgp_num: 48
waw-hdd-redundant-3-object.rgw.log pg_num: 48 pgp_num: 48
waw-hdd-redundant-3-object.rgw.buckets.index pg_num: 47 pgp_num: 47
.rgw.root pg_num: 49 pgp_num: 48
waw-hdd-redundant-3-object.rgw.buckets.non-ec pg_num: 48 pgp_num: 46
```

(this will probably take another hour or so)

#11 - 09/12/2021 11:44 AM - q3k

Okay, resize done. Cluster is almost healthy, now only complaining about global_id reclaim still being turned on (we'll turn it off after we finish upgrading all OSDs to Ceph 15).

Continuing upgrade. First, let's assume that the earlier Bluestore corruption was, as developers suggest, because the fsck happened multithreaded. We'll attempt to remediate things in two ways:

```
[root@bc01n02 /]# ceph config set osd bluestore_fsck_quick_fix_threads 1
```

This should disable multi-threaded quick-fix fscks.

Then, I'll run a bluestore-tool fsck on each Ceph 14 OSD before triggering the upgrade, just to make double sure that the fsck doesn't actually attempt to correct anything.

#12 - 09/12/2021 11:48 AM - q3k

Before taking out the OSDs one-by-one to run the fsck, I'm going to set noout to make sure Ceph doesn't auto-out these OSDs after some time, and noscrub to limit the amount of background scrubbing as I now start messing around.

```
[root@bc01n02 /]# ceph osd set noout
noout is set
[root@bc01n02 /]# ceph osd set noscrub
noscrub is set
```

Now, how do we run ceph-bluestore-tool on OSD pods while the OSD is actually torn down... The joys of containerization. I could always get Ceph 14 on the hosts and ceph-volume lvm activate them there, but that seems sketchy (can I even run ceph-bluestore-tool without a proper keyring/ceph.conf setup?).

#13 - 09/12/2021 11:49 AM - q3k

Oh, before I do that, I still need to yeet osd.6:

```
[root@bc01n02 /]# ceph osd status
ID  HOST                 USED  AVAIL  WR OPS  WR DATA  RD OPS  RD DATA  STATE
 0  dcr01s22.hswaw.net   604G  4984G      5     39.1k      2     9829   exists,up
 1  dcr01s24.hswaw.net   828G  4760G      4     65.5k      1        0   exists,up
 2  dcr01s24.hswaw.net   832G  4756G     14      145k      3     1638   exists,up
 3  dcr01s22.hswaw.net   570G  5018G     21      418k      2     1638   exists,up
 4  dcr01s24.hswaw.net   761G  4827G      2     18.3k      2     1638   exists,up
 5  dcr01s22.hswaw.net   598G  4990G      3     44.7k      2     4095   exists,up
 6  dcr01s24.hswaw.net     0      0       3     27.1k      1     3562   autoout,exists
 7  dcr01s22.hswaw.net   653G  4935G      3     31.1k      1     4095   exists,up
[root@bc01n02 /]# ceph osd purge 6
purged osd.6
[root@bc01n02 /]# ceph osd status
ID  HOST                 USED  AVAIL  WR OPS  WR DATA  RD OPS  RD DATA  STATE
 0  dcr01s22.hswaw.net   604G  4984G      0     4095       2     1638   exists,up
 1  dcr01s24.hswaw.net   828G  4760G     14      216k      1     4914   exists,up
 2  dcr01s24.hswaw.net   832G  4756G     17      455k      3     1638   exists,up
 3  dcr01s22.hswaw.net   570G  5018G      4      153k      2     3276   exists,up
 4  dcr01s24.hswaw.net   761G  4827G      3      247k      2     4914   exists,up
 5  dcr01s22.hswaw.net   598G  4990G      1      5733      1        0   exists,up
 7  dcr01s22.hswaw.net   653G  4935G      0      6552      1     9829   exists,up
```

#14 - 09/12/2021 11:54 AM - q3k

And zapped it manually on the host by doing lvremove/vgremove and

```
dd if=/dev/zero of=/dev/sde bs=10M count=100
```

.

#15 - 09/12/2021 12:28 PM - q3k

Tried upgrading osd.0 by first transforming the deployment into a sleep 3600, doing a ceph-bluestore-tool fsck in the resulting shell, and then restoring the deployment with Ceph upped to 15, but that caused corruption too:

```
debug    -23> 2021-09-12T12:22:25.780+0000 7fe5ec530700 -1 bluestore(/var/lib/ceph/osd/ceph-0) fsck error: #13
:b784b99d:::rbd_data.14.629e3a6f68f598.0000000000004668:head# - 1 zombie spanning blob(s) found, the first one
: Blob(0x558461935c70 spanning 5514 blob([!~10000] csum crc32c/0x1000) use_tracker(0x10000 0x0) SharedBlob(0x5
5846192ae70 sbid 0x0))
debug    -22> 2021-09-12T12:22:25.862+0000 7fe5ecd31700  5 prioritycache tune_memory target: 6000000000 mapped
: 1945747456 unmapped: 286720 heap: 1946034176 old mem: 4294693630 new mem: 4294693631
debug    -21> 2021-09-12T12:22:26.248+0000 7fe5ec530700 -1 bluestore(/var/lib/ceph/osd/ceph-0) fsck warning: #
18:89bc86b9:::rbd_header.7d81f930e5d406:head# has omap that is not per-pool or pgmeta
debug    -20> 2021-09-12T12:22:26.547+0000 7fe5ec530700 -1 bluestore(/var/lib/ceph/osd/ceph-0) fsck warning: #
18:ca58f550:::rbd_header.7760ba6b8b4567:head# has omap that is not per-pool or pgmeta
debug    -19> 2021-09-12T12:22:26.866+0000 7fe5ecd31700  5 prioritycache tune_memory target: 6000000000 mapped
: 2106179584 unmapped: 2031616 heap: 2108211200 old mem: 4294693631 new mem: 4294693631
```

```
debug    -18> 2021-09-12T12:22:26.866+0000 7fe5ecd31700  5 bluestore.MempoolThread(0x5583ee16ca98) _resize_sha
rds cache_size: 4294693631 kv_alloc: 2382364672 kv_used: 1249297728 meta_alloc: 1174405120 meta_used: 39169504
 data_alloc: 704643072 data_used: 0
debug    -17> 2021-09-12T12:22:27.104+0000 7fe5fe245f00  0 bluestore(/var/lib/ceph/osd/ceph-0) _fsck_check_obj
ects partial offload, done myself 39470 of 208225objects, threads 1
debug    -16> 2021-09-12T12:22:27.108+0000 7fe5fe245f00  1 bluestore(/var/lib/ceph/osd/ceph-0) _fsck_on_open c
hecking shared_blobs
debug    -15> 2021-09-12T12:22:27.513+0000 7fe5fe245f00  1 bluestore(/var/lib/ceph/osd/ceph-0) _fsck_on_open c
hecking pool_statfs
debug    -14> 2021-09-12T12:22:27.513+0000 7fe5fe245f00  5 bluestore(/var/lib/ceph/osd/ceph-0) _fsck_on_open m
arking per_pool_omap=1
debug    -13> 2021-09-12T12:22:27.513+0000 7fe5fe245f00  5 bluestore(/var/lib/ceph/osd/ceph-0) _fsck_on_open a
pplying repair results
debug    -12> 2021-09-12T12:22:27.590+0000 7fe5fe245f00 -1 rocksdb: submit_common error: Corruption: bad Write
Batch Delete code = 2 Rocksdb transaction:
Put( Prefix = O key = 0x7f800000000000000d10b31ed1217262'd_data.14.629e3a6f68f598.0000000000003f6c!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 13392)
Put( Prefix = O key = 0x7f800000000000000d11c7f0c3217262'd_data.14.629e3a6f68f598.0000000000003f88!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 10780)
Put( Prefix = O key = 0x7f800000000000000d11df1080217262'd_data.14.5712f265de286c.0000000000000ee3!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 7063)
Put( Prefix = O key = 0x7f800000000000000d11f246'U!rbd_data.14.629e3a6f68f598.00000000000044b1!='0xffffffffff
fffeffffffffffffffff'o' Value size = 10841)
Put( Prefix = O key = 0x7f800000000000000d103fecab217262'd_data.14.5712f265de286c.00000000000012d8!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 13945)
Put( Prefix = O key = 0x7f800000000000000d105e6794217262'd_data.14.aaaa29c3d54b6c.00000000000009a1!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 11040)
Put( Prefix = O key = 0x7f800000000000000d1080ed'{!rbd_data.14.629e3a6f68f598.0000000000003224!='0xffffffffff
fffeffffffffffffffff'o' Value size = 1319)
Put( Prefix = O key = 0x7f800000000000000d1d63c4'|!rbd_data.14.629e3a6f68f598.0000000000000073!='0xffffffffff
fffeffffffffffffffff'o' Value size = 10957)
Put( Prefix = O key = 0x7f800000000000000d1156e6c4217262'd_data.14.629e3a6f68f598.00000000000007c1!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 10650)
Put( Prefix = O key = 0x7f800000000000000d1dd4c8cc217262'd_data.14.62ebec4ec2a5ba.0000000000001dac!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 432)
Put( Prefix = O key = 0x7f800000000000000d2c315c'|!rbd_data.14.62ebec4ec2a5ba.0000000000000aa3!='0xffffffffff
fffeffffffffffffffff'o' Value size = 2693)
Put( Prefix = O key = 0x7f800000000000000d2cfcfa'y!rbd_data.14.89d04458c22ad0.00000000000005c4!='0xffffffffff
fffeffffffffffffffff'o' Value size = 7143)
Put( Prefix = O key = 0x7f800000000000000d2c76c5'$!rbd_data.14.629e3a6f68f598.0000000000004566!='0xffffffffff
fffeffffffffffffffff'o' Value size = 14552)
Put( Prefix = O key = 0x7f800000000000000d2d8137bf217262'd_data.14.629e3a6f68f598.000000000000323c!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 1525)
Put( Prefix = O key = 0x7f800000000000000d2ddc3c09217262'd_data.14.5712f265de286c.0000000000000f42!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 7463)
Put( Prefix = O key = 0x7f800000000000000d2e0120'?!rbd_data.14.629e3a6f68f598.000000000000325b!='0xffffffffff
fffeffffffffffffffff'o' Value size = 1242)
Put( Prefix = O key = 0x7f800000000000000d1f1deb'4!rbd_data.14.629e3a6f68f598.00000000000007c2!='0xffffffffff
fffeffffffffffffffff'o' Value size = 9534)
Put( Prefix = O key = 0x7f800000000000000d1f400789217262'd_data.14.629e3a6f68f598.0000000000050f9!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 7709)
Put( Prefix = O key = 0x7f800000000000000d1f5341ff217262'd_data.14.62ebec4ec2a5ba.0000000000001331!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 432)
Put( Prefix = O key = 0x7f800000000000000d1fbcca'%!rbd_data.14.5712f265de286c.000000000000028f!='0xffffffffff
fffeffffffffffffffff'o' Value size = 9093)
Put( Prefix = O key = 0x7f800000000000000d2fb638'.!rbd_data.14.89d04458c22ad0.0000000000000974!='0xffffffffff
fffeffffffffffffffff'o' Value size = 12949)
Put( Prefix = O key = 0x7f800000000000000d2fcbc31b217262'd_data.14.5712f265de286c.0000000000000f1a!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 14316)
Put( Prefix = O key = 0x7f800000000000000d383bee05217262'd_data.14.5712f265de286c.0000000000001a86!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 8432)
Put( Prefix = O key = 0x7f800000000000000d2d1da199217262'd_data.14.629e3a6f68f598.0000000000004f76!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 11844)
Put( Prefix = O key = 0x7f800000000000000d2d320b11217262'd_data.14.62ebec4ec2a5ba.0000000000001b44!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 432)
Put( Prefix = O key = 0x7f800000000000000d3b2673b5217262'd_data.14.5712f265de286c.000000000000194a!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 9752)
Put( Prefix = O key = 0x7f800000000000000d3a8cbf'&!rbd_data.14.5712f265de286c.00000000000010ae!='0xffffffffff
fffeffffffffffffffff'o' Value size = 6409)
Put( Prefix = O key = 0x7f800000000000000d3bd37cf5217262'd_data.14.5712f265de286c.0000000000001c4a!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 18433)
Put( Prefix = O key = 0x7f800000000000000d3b981f88217262'd_data.14.629e3a6f68f598.000000000000282c!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 13416)
Put( Prefix = O key = 0x7f800000000000000d2eb599'>!rbd_data.14.89d04458c22ad0.0000000000000128!='0xffffffffff
fffeffffffffffffffff'o' Value size = 7778)
```

```
Put( Prefix = O key = 0x7f800000000000000d2ed2cbc0217262'd_data.14.629e3a6f68f598.0000000000001ca3!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 5214)
Put( Prefix = O key = 0x7f800000000000000d2ed4c3de217262'd_data.14.89d04458c22ad0.0000000000003241!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 1267)
Put( Prefix = O key = 0x7f800000000000000d2edb74cb217262'd_data.14.629e3a6f68f598.0000000000002a08!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 5130)
Put( Prefix = O key = 0x7f800000000000000d2ee6c494217262'd_data.14.89d04458c22ad0.00000000000000db!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 3067)
Put( Prefix = O key = 0x7f800000000000000d2f067e'/!rbd_data.14.629e3a6f68f598.0000000000003242!='0xfffffffffff
ffffeffffffffffffffff'o' Value size = 421)
Put( Prefix = O key = 0x7f800000000000000d397d5ca1217262'd_data.14.25d8431322e111.0000000000000664!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 3485)
Put( Prefix = O key = 0x7f800000000000000d39b743'j!rbd_data.14.89d04458c22ad0.00000000000322b!='0xfffffffffff
ffffeffffffffffffffff'o' Value size = 1828)
Put( Prefix = O key = 0x7f800000000000000d1114f79d217262'd_data.14.62ebec4ec2a5ba.0000000000002ac1!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 432)
Put( Prefix = O key = 0x7f800000000000000d111ee9ba217262'd_data.14.89d04458c22ad0.000000000000015d!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 15859)
Put( Prefix = O key = 0x7f800000000000000d112ad7'$!rbd_data.14.5712f265de286c.0000000000006466!='0xfffffffffff
ffffeffffffffffffff'o' Value size = 2238)
Put( Prefix = O key = 0x7f800000000000000d1e5336'J!rbd_data.14.225ed8d6d5eeed.0000000000000071!='0xfffffffffff
ffffeffffffffffffff'o' Value size = 9928)
Put( Prefix = O key = 0x7f800000000000000d1e829607217262'd_data.14.629e3a6f68f598.0000000000001dda!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 8479)
Put( Prefix = O key = 0x7f800000000000000d1313f69f217262'd_data.14.89d04458c22ad0.00000000000004a8!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 8358)
Put( Prefix = O key = 0x7f800000000000000d2f37e78f217262'd_data.14.62ebec4ec2a5ba.0000000000002b60!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 432)
Put( Prefix = O key = 0x7f800000000000000d2f5494a8217262'd_data.14.629e3a6f68f598.0000000000004786!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 9011)
Put( Prefix = O key = 0x7f800000000000000d13f5ad'.!rbd_data.14.5712f265de286c.0000000000005f30!='0xfffffffffff
ffffeffffffffffffff'o' Value size = 704)
Put( Prefix = O key = 0x7f800000000000000d13f87d92217262'd_data.14.629e3a6f68f598.00000000000026e9!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 11964)
Put( Prefix = O key = 0x7f800000000000000d3beb9c87217262'd_data.14.5712f265de286c.0000000000000ff9!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 8419)
Put( Prefix = O key = 0x7f800000000000000d481bd809217262'd_data.14.62ebec4ec2a5ba.0000000000003540!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 432)
Put( Prefix = O key = 0x7f800000000000000d732c3a12217262'd_data.14.5712f265de286c.000000000000108a!='0xffffffff
ffffffffeffffffffffffffff'o' Value size = 2569)
debug    -11> 2021-09-12T12:22:27.590+0000 7fe5fe245f00  5 bluestore(/var/lib/ceph/osd/ceph-0) _fsck_on_open r
epair applied
debug    -10> 2021-09-12T12:22:27.590+0000 7fe5fe245f00  2 bluestore(/var/lib/ceph/osd/ceph-0) _fsck_on_open 2
08225 objects, 158375 of them sharded.
debug     -9> 2021-09-12T12:22:27.590+0000 7fe5fe245f00  2 bluestore(/var/lib/ceph/osd/ceph-0) _fsck_on_open 1
839921 extents to 1541569 blobs, 197832 spanning, 170932 shared.
debug     -8> 2021-09-12T12:22:27.590+0000 7fe5fe245f00  1 bluestore(/var/lib/ceph/osd/ceph-0) _fsck_on_open <
<<FINISH>>> with 162 errors, 60 warnings, 222 repaired, 0 remaining in 10.751054 seconds
debug     -7> 2021-09-12T12:22:27.809+0000 7fe5fe245f00  2 osd.0 0 journal looks like hdd
debug     -6> 2021-09-12T12:22:27.809+0000 7fe5fe245f00  2 osd.0 0 boot
debug     -5> 2021-09-12T12:22:27.836+0000 7fe5fe245f00  1 osd.0 64245 init upgrade snap_mapper (first start a
s octopus)
debug     -4> 2021-09-12T12:22:27.839+0000 7fe5e8528700  5 bluestore(/var/lib/ceph/osd/ceph-0) _kv_sync_thread
 utilization: idle 11.000537637s of 11.000539470s, submitted: 0
debug     -3> 2021-09-12T12:22:27.839+0000 7fe5e8528700 -1 rocksdb: submit_common error: Corruption: bad Write
Batch Delete code = 2 Rocksdb transaction:
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000008B_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000001DF8_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000001E6B_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_00000000000037AA_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_00000000000038DC_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000992E_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009930_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009936_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009944_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009948_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000994A_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000994C_' Value size = 95)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009952_' Value size = 95)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009954_' Value size = 95)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009958_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000995A_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000995C_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009964_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009968_' Value size = 96)
```

```
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000996A_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000996C_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009970_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009972_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009976_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009978_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000997A_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000997C_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000997E_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_0000000000009984_' Value size = 96)
Put( Prefix = m key = 0x0000000000000000000000000000000402'.SNA_13_000000000000998E_' Value size = 96)
Put( Prefix = S key = 'nid_max' Value size = 8)
Put( Prefix = S key = 'blobid_max' Value size = 8)
debug     -2> 2021-09-12T12:22:27.840+0000 7fe5fe245f00  1 snap_mapper.convert_legacy converted 2254 keys in 0
.00370446s
debug     -1> 2021-09-12T12:22:27.841+0000 7fe5e8528700 -1 /home/jenkins-build/build/workspace/ceph-build/ARCH
/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos8/DIST/centos8/MACHINE_SIZE/gigantic/release/15.2.13/rpm/el
8/BUILD/ceph-15.2.13/src/os/bluestore/BlueStore.cc: In function 'void BlueStore::_txc_apply_kv(BlueStore::Tran
sContext*, bool)' thread 7fe5e8528700 time 2021-09-12T12:22:27.840859+0000
/home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos8/DIST/c
entos8/MACHINE_SIZE/gigantic/release/15.2.13/rpm/el8/BUILD/ceph-15.2.13/src/os/bluestore/BlueStore.cc: 11868:
FAILED ceph_assert(r == 0)

 ceph version 15.2.13 (c44bc49e7a57a87d84dfff2a077a2058aa2172e2) octopus (stable)
 1: (ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0x158) [0x5583e2b07bd8]
 2: (()+0x507df2) [0x5583e2b07df2]
 3: (BlueStore::_txc_apply_kv(BlueStore::TransContext*, bool)+0x44f) [0x5583e30d103f]
 4: (BlueStore::_kv_sync_thread()+0x176f) [0x5583e30f66ef]
 5: (BlueStore::KVSyncThread::entry()+0x11) [0x5583e311e941]
 6: (()+0x814a) [0x7fe5fbfa314a]
 7: (clone()+0x43) [0x7fe5facdaf23]

debug      0> 2021-09-12T12:22:27.844+0000 7fe5e8528700 -1 *** Caught signal (Aborted) **
 in thread 7fe5e8528700 thread_name:bstore_kv_sync

 ceph version 15.2.13 (c44bc49e7a57a87d84dfff2a077a2058aa2172e2) octopus (stable)
 1: (()+0x12b20) [0x7fe5fbfadb20]
 2: (gsignal()+0x10f) [0x7fe5fac157ff]
 3: (abort()+0x127) [0x7fe5fabffc35]
 4: (ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0x1a9) [0x5583e2b07c29]
 5: (()+0x507df2) [0x5583e2b07df2]
 6: (BlueStore::_txc_apply_kv(BlueStore::TransContext*, bool)+0x44f) [0x5583e30d103f]
 7: (BlueStore::_kv_sync_thread()+0x176f) [0x5583e30f66ef]
 8: (BlueStore::KVSyncThread::entry()+0x11) [0x5583e311e941]
 9: (()+0x814a) [0x7fe5fbfa314a]
 10: (clone()+0x43) [0x7fe5facdaf23]
 NOTE: a copy of the executable, or `objdump -rdS <executable>` is needed to interpret this.

--- logging levels ---
   0/ 5 none
   0/ 1 lockdep
   0/ 1 context
   1/ 1 crush
   1/ 5 mds
   1/ 5 mds_balancer
   1/ 5 mds_locker
   1/ 5 mds_log
   1/ 5 mds_log_expire
   1/ 5 mds_migrator
   0/ 1 buffer
   0/ 1 timer
   0/ 1 filer
   0/ 1 striper
   0/ 1 objecter
   0/ 5 rados
   0/ 5 rbd
   0/ 5 rbd_mirror
   0/ 5 rbd_replay
   0/ 5 rbd_rwl
   0/ 5 journaler
   0/ 5 objectcacher
   0/ 5 immutable_obj_cache
   0/ 5 client
   1/ 5 osd
   0/ 5 optracker
```

```
   0/ 5 objclass
   1/ 3 filestore
   1/ 3 journal
   0/ 0 ms
   1/ 5 mon
   0/10 monc
   1/ 5 paxos
   0/ 5 tp
   1/ 5 auth
   1/ 5 crypto
   1/ 1 finisher
   1/ 1 reserver
   1/ 5 heartbeatmap
   1/ 5 perfcounter
   1/ 5 rgw
   1/ 5 rgw_sync
   1/10 civetweb
   1/ 5 javaclient
   1/ 5 asok
   1/ 1 throttle
   0/ 0 refs
   1/ 5 compressor
   1/ 5 bluestore
   1/ 5 bluefs
   1/ 3 bdev
   1/ 5 kstore
   4/ 5 rocksdb
   4/ 5 leveldb
   4/ 5 memdb
   1/ 5 fuse
   1/ 5 mgr
   1/ 5 mgrc
   1/ 5 dpdk
   1/ 5 eventtrace
   1/ 5 prioritycache
   0/ 5 test
  -2/-2 (syslog threshold)
  99/99 (stderr threshold)
--- pthread ID / name mapping for recent threads ---
  7fe5e5f29700 / rocksdb:dump_st
  7fe5e8528700 / bstore_kv_sync
  7fe5ec530700 /
  7fe5ecd31700 / bstore_mempool
  7fe5f57b9700 / signal_handler
  7fe5f67bb700 / admin_socket
  7fe5f6fbc700 / service
  7fe5f7fbe700 / msgr-worker-1
  7fe5fe245f00 / ceph-osd
  max_recent     10000
  max_new         1000
  log_file /var/lib/ceph/crash/2021-09-12T12:22:27.844834Z_69e55028-c5f1-4239-9db7-d2f739bc7d68/log
--- end dump of recent events ---
```

sigh , I think that's becuase I should've done --command repair instead of fsck. Let's see about possibly recovering this OSD now I guess, or just yeeting it out of the pool again...

#16 - 09/12/2021 12:29 PM - q3k

(after restart:)

```
debug 2021-09-12T12:27:17.322+0000 7ff41efe3f00  4 rocksdb: EVENT_LOG_v1 {"time_micros": 1631449637323125, "jo
b": 1, "event": "recovery_started", "log_files": [98044]}
debug 2021-09-12T12:27:17.322+0000 7ff41efe3f00  4 rocksdb: [db/db_impl_open.cc:583] Recovering log #98044 mod
e 0
debug 2021-09-12T12:27:17.592+0000 7ff41efe3f00  3 rocksdb: [db/db_impl_open.cc:518] db.wal/098044.log: droppi
ng 1124650 bytes; Corruption: bad WriteBatch Delete
debug 2021-09-12T12:27:17.592+0000 7ff41efe3f00  4 rocksdb: [db/db_impl.cc:390] Shutdown: canceling all backgr
ound work
debug 2021-09-12T12:27:17.592+0000 7ff41efe3f00  4 rocksdb: [db/db_impl.cc:563] Shutdown complete
debug 2021-09-12T12:27:17.592+0000 7ff41efe3f00 -1 rocksdb: Corruption: bad WriteBatch Delete
debug 2021-09-12T12:27:17.592+0000 7ff41efe3f00 -1 bluestore(/var/lib/ceph/osd/ceph-0) _open_db erroring openi
ng db:
debug 2021-09-12T12:27:17.592+0000 7ff41efe3f00  1 bluefs umount
debug 2021-09-12T12:27:17.592+0000 7ff41efe3f00  1 bdev(0x55d19a20a380 /var/lib/ceph/osd/ceph-0/block) close
debug 2021-09-12T12:27:17.755+0000 7ff41efe3f00  1 bdev(0x55d19a20a000 /var/lib/ceph/osd/ceph-0/block) close
```

```
debug 2021-09-12T12:27:18.011+0000 7ff41efe3f00 -1 osd.0 0 OSD:init: unable to mount object store
debug 2021-09-12T12:27:18.011+0000 7ff41efe3f00 -1  ** ERROR: osd init failed: (5) Input/output error
```

#17 - 09/12/2021 12:32 PM - q3k

Yeah, fuck it, marked it as out, waiting for rebalance again. *sigh*. But I think I'm also gonna re-introduce the two failed OSDs after this because I'm getting freaked out about how little data we have left. Actually, can I easily re-introduce them, given that we can't really run Rook now as it's halfway through a Ceph update?

I guess I'll try the next 14->15 OSD with the 'right' (as per the ML port and issue tracker) ceph-bluestore-tool command this time.

#18 - 09/12/2021 12:41 PM - q3k

Hm, seems like a fix actually did land for this in Ceph 15, 15.2.14 to be precise. However, there isn't a ceph/ceph dockerhub tag for this version? Ugh.

#19 - 09/12/2021 12:43 PM - q3k

Ah, that might explain it:

```
As of August 2021, new container images are pushed to quay.io registry only. Docker hub won't receive new cont
ent for that specific image but current images remain available.
```

So I guess we can try switching to quay.io/ceph/ceph:v15.2.14 for the next migration and see if that works.

#20 - 09/12/2021 09:49 PM - q3k

Okay, upgrading to quay.io/ceph/ceph:v15.2.14 seems to have been the solution to not shred OSDs. Whoops.

Now considering continuing the upgrade spree and bumping to 16, but I also kinda wanna go to sleep.

#21 - 09/13/2021 07:35 PM - q3k

Upgraded yesterday to 16: https://gerrit.hackerspace.pl/c/hscloud/+/1093

#22 - 09/14/2021 01:11 PM - q3k

Moved the ceph-waw3 radosgw to be in a proper realm/zonegroup so that we can use radosgw multisite to easily migrate all S3 users/buckets and data into k0: https://gerrit.hackerspace.pl/1095

#23 - 06/10/2022 10:36 PM - q3k

Yet another outage caused by this today. Kinda.

Power failure of all of W2A -> corrupt MON data on bc01n01. That was the only mon. Had to restore from OSDs again.

We would have had more mons if we trusted rook and/or finally moved into a static ceph deployment.

#24 - 06/10/2022 10:43 PM - q3k

This recovery also unearthed a ceph bug. If we start a mon with bind addrs

```
[v2:10.10.24.215:3300/0,v1:10.10.24.215:6789/0]
```

but the monmap addrs set to

```
v1:10.10.12.115:6789/0
```

we get an assertion failure:

```
/usr/include/c++/8/bits/stl_vector.h:950: std::vector<_Tp, _Alloc>::const_reference std::vector<_Tp, _Alloc>::
operator[](std::vector<_Tp, _Alloc>::size_type) const [with _Tp = entity_addr_t; _Alloc = std::allocator<entit
y_addr_t>; std::vector<_Tp, _Alloc>::const_reference = const  entity_addr_t&; std::vector<_Tp, _Alloc>::size_t
ype = long unsigned int]: Assertion '__builtin_expect(__n <this->size(), true)' failed.

1: /lib64/libpthread.so.0(+0x12b20) [0x7fa200b33b20]
2: gsignal()
3: abort()
4: /usr/lib64/ceph/libceph-common.so.2(+0x2da6a8) [0x7fa20309d6a8]
5: (Processor::accept()+0x5f7) [0x7fa20331b347]
6: (EventCenter::process_events(unsigned int, std::chrono::duration<unsigned long, std::ratio<1l, 1000000000l>
 >*)+0xcb7) [0x7fa203370e37]
7: /usr/lib64/ceph/libceph-common.so.2(+0x5b434c) [0x7fa20337734c]
8: /lib64/libstdc++.so.6(+0xc2ba3) [0x7fa20017eba3]
9: /lib64/libpthread.so.0(+0x814a) [0x7fa200b2914a]
10: clone()
```

Digging into this (thanks, Ghidra), that seems to be caused by

```
msgr->get_myaddrs().v[listen_socket.get_addr_slot()]
```

in https://github.com/ceph/ceph/blob/master/src/msg/async/AsyncMessenger.cc#L197 . In other words, that assertion failure is a Vector being indexed out of bounds due to the misconfiguration. Doing

```
monmaptool --addv a [v2:10.10.12.115:3300,v1:10.10.12.115:6789] monmap
```

fixed things.

#25 - 06/10/2022 10:46 PM - q3k

Anyway, current cluster state:

```
[root@rook-ceph-tools-bfcdb4794-xp5zw /]# ceph -s
  cluster:
    id:     ea847d45-da0b-4be0-8c77-2c2db021aaa0
    health: HEALTH_WARN
            26 daemons have recently crashed

  services:
    mon: 1 daemons, quorum a (age 2h)
    mgr: a(active, since 2h)
    osd: 6 osds: 6 up (since 2h), 6 in (since 9M)
    rgw: 1 daemon active (1 hosts, 1 zones)

  data:
    pools:   14 pools, 401 pgs
    objects: 2.20M objects, 5.6 TiB
    usage:   12 TiB used, 21 TiB / 33 TiB avail
    pgs:     338 active+clean
             49  active+clean+snaptrim_wait
             12  active+clean+snaptrim
             2   active+clean+scrubbing+deep+repair

  io:
    client:   3.9 MiB/s rd, 858 KiB/s wr, 172 op/s rd, 75 op/s wr
```

Let's wait for the snaptrims/scrubs to finish and then I'll consider adding two more mons still in Rook. The snaptrims taking this long are a bit suspicious, though. We'll see.

#26 - 06/11/2022 12:16 AM - q3k

Let's try to make these snaptrips faster.

```
# ceph tell osd.* config set osd_max_trimming_pgs 8
[...]
# ceph -s
[...]

    pgs:     349 active+clean
             42  active+clean+snaptrim
             9   active+clean+snaptrim_wait
             1   active+clean+scrubbing+deep+repair
```

#27 - 06/11/2022 10:01 AM - q3k

Almost done. I wonder what's up with the backlog buildup.

```
# ceph -s
[...]
    pgs:     397 active+clean
             4   active+clean+snaptrim
```

#28 - 06/11/2022 12:08 PM - q3k

Scaled up rook to three mons:

```
rook-ceph-mon-a-6d9d798fb5-gnm5c                                  1/1     Running     0          15h     10.10.24
.243   bc01n01.hswaw.net    <none>           <none>
rook-ceph-mon-e-55b6ff8fcf-qk9r8                                  1/1     Running     0          8m49s   10.10.25
.64    dcr01s24.hswaw.net   <none>           <none>
rook-ceph-mon-f-7d4dd7465-w7rv6                                   1/1     Running     0          8m26s   10.10.24
.129   dcr01s22.hswaw.net   <none>           <none>
```

```
    mon: 3 daemons, quorum a,e,f (age 4m)
```

This was done by temporarily cordoning bc01n02 to make sure no mon lands there.

One pg left in snaptrim, then I'm gonna call this a 'success'. Well, we're still stuck with non-CSI rook, but it's a bit healthier now.

#29 - 07/04/2022 01:08 PM - q3k

- Category set to hscloud