MAKING DISTRIBUTED STORAGE EASY: USABILITY IN CEPH LUMINOUS AND BEYOND

SAGE WEIL – RED HAT

2018.01.26

PLAN

- Ceph
- Luminous
- Simplify
- Automate
- Manage
- Mimic

CEPH IS...

- Object, block, and file storage in a single cluster
- All components scale horizontally
- No single point of failure
- Hardware agnostic, commodity hardware
- Self-managing whenever possible
- Free and open source software (LGPL)

CEPH IS HARD

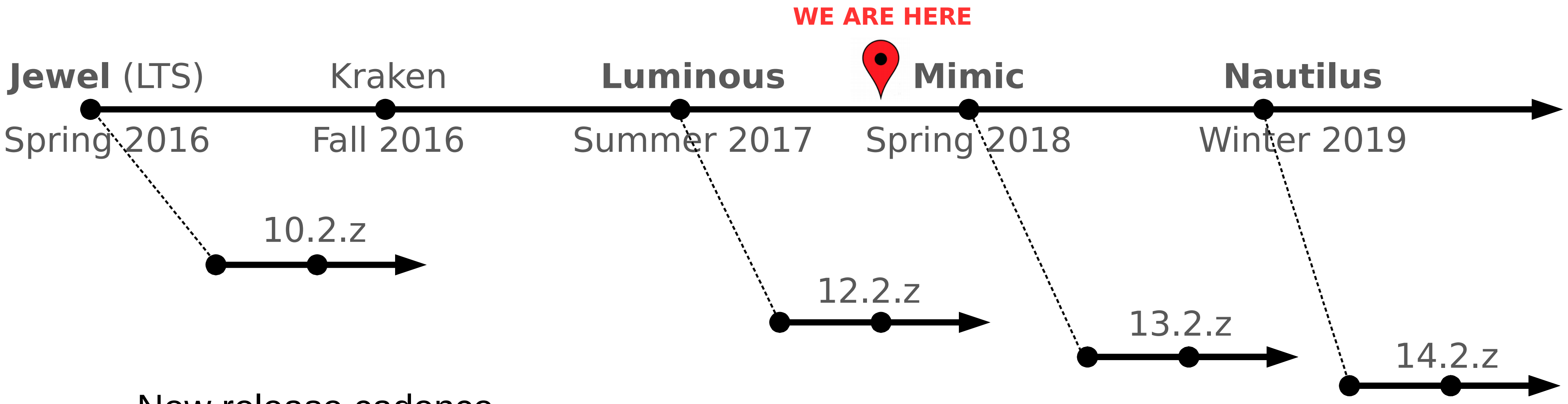
WE MUST
MAKE IT EASY

LUMINOUS

LUMINOUS GOODNESS

- RADOS
  - BlueStore (a new OSD backend)
    - stable and default
    - full data checksums
    - compression
  - Erasure coding overwrites
    - now usable by RBD, CephFS
  - ceph-mgr
    - scalability
    - prometheus, zabbix, restful
    - new web dashboard
  - AsyncMessenger by default
- RGW (object)
  - metadata search
  - compression and encryption
  - NFS gateway (v3 and v4)
- RBD (block)
  - HA iSCSI (finally!)
  - async mirroring improvements
- CephFS (file)
  - multiple MDS daemons
  - subtree pinning
  - auto directory fragmentation

CEPH RELEASES



New release cadence

- Named release every 9 months
- Backports for 2 releases
- Upgrade up to 2 releases at a time
  (e.g., Luminous → Nautilus)

SIMPLIFY

CEPH -S (BEFORE)

```
    cluster 8ba08162-a390-479c-a698-6d8911c4f451
     health HEALTH_OK
     monmap e2: 3 mons at
{a=172.21.9.34:6789/0,b=172.21.9.34:6790/0,c=172.21.9.34:6791/0}
            election epoch 8, quorum 0,1,2 a,b,c
      fsmap e15: 1/1/1 up {0=b=up:active}, 1 up:standby
        mgr active: x
     osdmap e21: 1 osds: 1 up, 1 in
      pgmap v191: 26 pgs, 3 pools, 4145 kB data, 23 objects
            494 GB used, 436 GB / 931 GB avail
                  26 active+clean
```

CEPH -S (AFTER)

```
  cluster:
    id:     0554f6f9-6061-425d-a343-f246020f1464
    health: HEALTH_OK

  services:
    mon: 1 daemons, quorum a
    mgr: x(active)
    mds: cephfs_a-1/1/1 up {[cephfs_a:0]=b=up:active}, 1 up:standby
    osd: 3 osds: 3 up, 3 in

  data:
    pools:   5 pools, 40 pgs
    objects: 42 objects, 4492 bytes
    usage:   1486 GB used, 1306 GB / 2793 GB avail
    pgs:     40 active+clean
```

HEALTH WARNINGS

```
health HEALTH_WARN
        4 pgs degraded
        5 pgs peering
        1 pgs recovering
        3 pgs recovery_wait
        recovery 609/5442 objects degraded (11.191%)
```

```
health: HEALTH_WARN
        Degraded data redundancy: 959/4791 objects degraded (20.017%), 5 pgs degraded
```

CLUSTER LOG (BEFORE)

```
cluster [INF] osdmap e20: 4 osds: 4 up, 3 in
cluster [INF] pgmap v142: 24 pgs: 3 active+recovery_wait+degraded, 21 active+clean; 56
 74 kB data, 1647 GB used, 1146 GB / 2793 GB avail; 818 kB/s wr, 230 op/s; 516/2256
objects degraded (22.872%); 0 B/s, 7 keys/s, 1 objects/s recovering
cluster [INF] pgmap v143: 24 pgs: 8 active+recovery_wait+degraded, 16 active+clean; 77
 19 kB data, 1647 GB used, 1145 GB / 2793 GB avail; 1428 kB/s wr, 577 op/s; 1021/2901
objects degraded (35.195%); 321 kB/s, 65 keys/s, 76 objects/s recovering
cluster [INF] pgmap v144: 24 pgs: 8 active+recovery_wait+degraded, 16 active+clean; 77
 30 kB data, 1647 GB used, 1145 GB / 2793 GB avail; 1090 kB/s wr, 483 op/s; 1021/3006
objects degraded (33.965%); 244 kB/s, 49 keys/s, 58 objects/s recovering
cluster [INF] pgmap v145: 24 pgs: 8 active+recovery_wait+degraded, 16 active+clean; 77
 30 kB data, 1647 GB used, 1145 GB / 2793 GB avail; 905 kB/s wr, 401 op/s; 1021/3006
objects degraded (33.965%); 203 kB/s, 41 keys/s, 48 objects/s recovering
cluster [INF] pgmap v146: 24 pgs: 5 active+recovery_wait+degraded, 19 active+clean; 80
 83 kB data, 1647 GB used, 1145 GB / 2793 GB avail; 0 B/s rd, 959 kB/s wr, 494 op/s;
505/3711 objects degraded (13.608%); 1006 kB/s, 56 keys/s, 90 objects/s recovering
```

CLUSTER LOG (AFTER)

```
cluster [WRN] Health check failed: Degraded data redundancy: 959/4791 objects degraded
(20.017%), 5 pgs degraded (PG_DEGRADED)
cluster [WRN] Health check update: Degraded data redundancy: 474/3399 objects degraded
(13.945%), 3 pgs degraded (PG_DEGRADED)
cluster [INF] Health check cleared: PG_DEGRADED (was: Degraded data redundancy: 474/3399
objects degraded (13.945%), 3 pgs degraded)
cluster [INF] Cluster is now healthy
```

CONFIGURATION

- >1400 configuration options
- minimal documentation
  - handful on https://docs.ceph.com
  - comments in config_opts.h (sometimes)
- mix of
  - user options
  - developer constants
  - debugging options to inject errors or debugging behavior
- difficult to determine relevant set of current options

- option schema (including docs) now embedded in code (options.cc)
  - ceph daemon <name> config help <option>
  - min/max, enum, or custom validators
- option levels: *basic*, *advanced*, and *dev*
- easy to identify changed options
  - ceph daemon <name> config diff
- configure cache sizes in bytes (not objects)
- similar levels + descriptions for perf counters

CENTRAL CONFIG (COMING IN MIMIC)

- ceph.conf management tedious and error-prone
  - tooling needed to manage at scale (puppet, chef, etc.)
- nobody likes ini files any more

- config stored on monitors
- new ‘ceph config …’ CLI
- prevent setting bogus values
- config changes at runtime
- “what is option X on daemon Y?”
- ‘assimilate-conf’ to import existing config files
- ceph.conf only (maybe) required for bootstrap
  - must identify monitor IPs
  - DNS SRV records can also do that

SIMPLIFY AUTH[NZ] SETUP

- cephx capabilities powerful but unfriendly
  - users must search docs for cap strings to copy/paste/modify

- ceph auth add client.foo mon ‘profile rbd’ osd ‘profile rbd’ ...

- ceph fs authorize <fsname> <entity/user> [rwp]
  - automatically applies to any data pools associated (now or later) with the file system

UPGRADES

```
$ ceph versions
{
    "mon": {
        "ceph version 12.2.2": 3
    },
    "mgr": {
        "ceph version 12.2.2": 2
    },
    "osd": {
        "ceph version 12.2.2": 7,
        "ceph version 12.2.1": 1
    },
    "mds": {},
    "overall": {
        "ceph version 12.2.2": 12,
        "ceph version 12.2.1": 1
    }
}
```

CLIENT COMPATIBILITY

- CRUSH tunables and other new/optional features affect client compat
  - often without admin realizing it
- new command declares compatibility
  - ceph osd set-require-min-compat-client <release>
  - prevent settings that break compat promise
  - cannot change compat promise if current settings do not allow it

```
$ ceph features
{
...
    "client": [
        "group": {
            "features": "0x107b84a842aca",
            "release": "hammer",
            "num": 3
        },
        "group": {
            "features": "0x40107b86a842ada",
            "release": "jewel",
            "num": 1
        },
        "group": {
            "features": "0x1ffddff8eea4fffb",
            "release": "luminous",
            "num": 5
        }
    ]
}
```

AUTOMATE

EASY STUFF

- MTU sized ping messages between OSDs
  - identify network/switch issues early
- disable auto-out on small clusters
- different (and sane) default values for HDDs and SSDs
- ceph-volume replacement for ceph-disk
  - adds support for dm-cache, (soon) VDO
  - LVM-based instead of GPT+udev based (reliable)

CEPH-MGR – WHAT

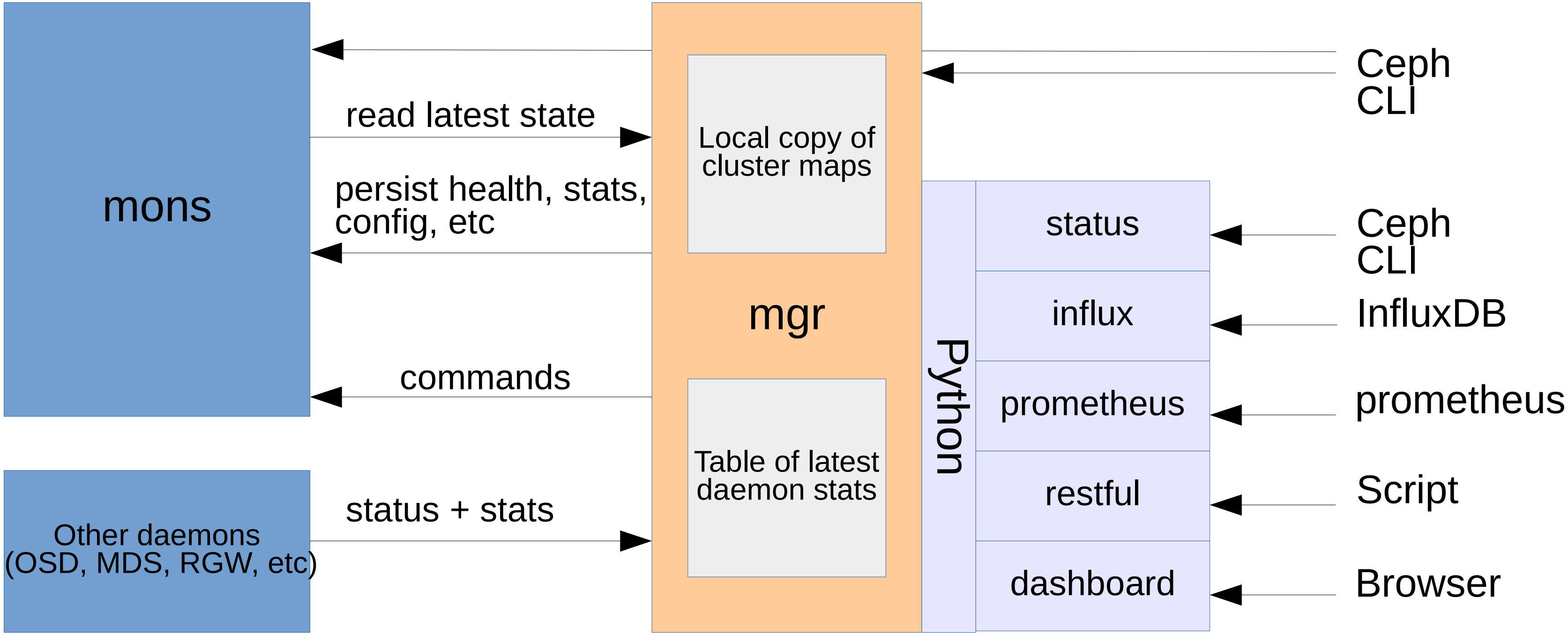
- a new core RADOS component
  - sibling of ceph-mon, ceph-osd
  - written in C++ to communicate natively (efficiently) with cluster
- mandatory
  - failed mgr affects reporting, introspection, APIs
  - does not affect not data path
- hosts **python modules** that implement monitoring/management

- initially added in Kraken, mandatory in Luminous

CEPH-MGR – WHY

- ceph-mon not a good home for high-level management
  - mon stability is very important – no sloppy 3rd party code
  - mon performance is important – minimize footprint, maximize scalability
  - mon’s state view is synchronous, expensive
- ceph-mgr has fast, async view of cluster state
  - lightweight and efficient
  - sufficient for introspection and management
- ceph-mon shrinks
  - drops stats responsibility
  - demonstrated scale of >10k OSDs (~40PB)

CEPH-MGR ARCHITECTURE

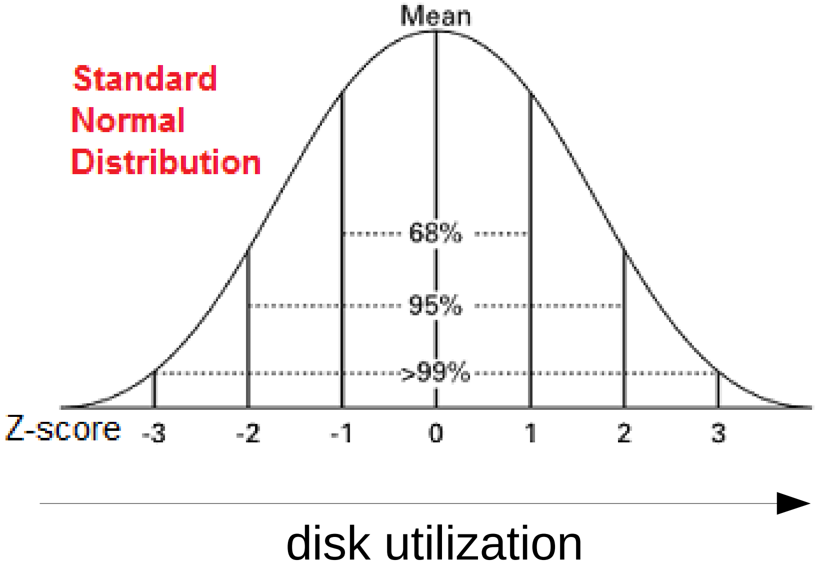


MODULES ARE EASY AND ROBUST

- easy
  - trivially implement new CLI commands (e.g., status)
  - expose cluster state (e.g., prometheus, influx, zabbix)
    - a few 100s of lines of code each
- robust
  - control the cluster (e.g., restful implements a full REST API) with cherrypy
  - dashboard module is a full web-based GUI
- Ceph handles the details
  - HA, failover, plumbing for cluster state, management hooks, …
  - modules ship with ceph itself
  - ‘ceph mgr module enable <name>’ to enable

BALANCER

- correct for normal variance in pseudorandom data placement
- builds and evaluates statistical model of (current or proposed) PG distribution
- automatically optimizes placement to minimize variance in OSD utilization
  - adjusts hidden CRUSH weights (backward compatible) or pg-upmap (luminous+)
  - throttles itself to avoid too much data movement at once
- ‘ceph balancer on’
  - commands to manually test if automated operation untrusted



PG_NUM (SHARDING)

- pg_num controls the shard count for pools
  - necessary for good performance
  - (used to be) necessary for balanced data distribution
  - affects resource utilization—many users end up with too many
  - implications for data reliability too
- picking pg_num for pools is “black magic”
  - not easy to provide generically applicable guidance
  - web-based tool helps, but...
- high stakes
  - resharding moves data around
  - can only be adjusted up

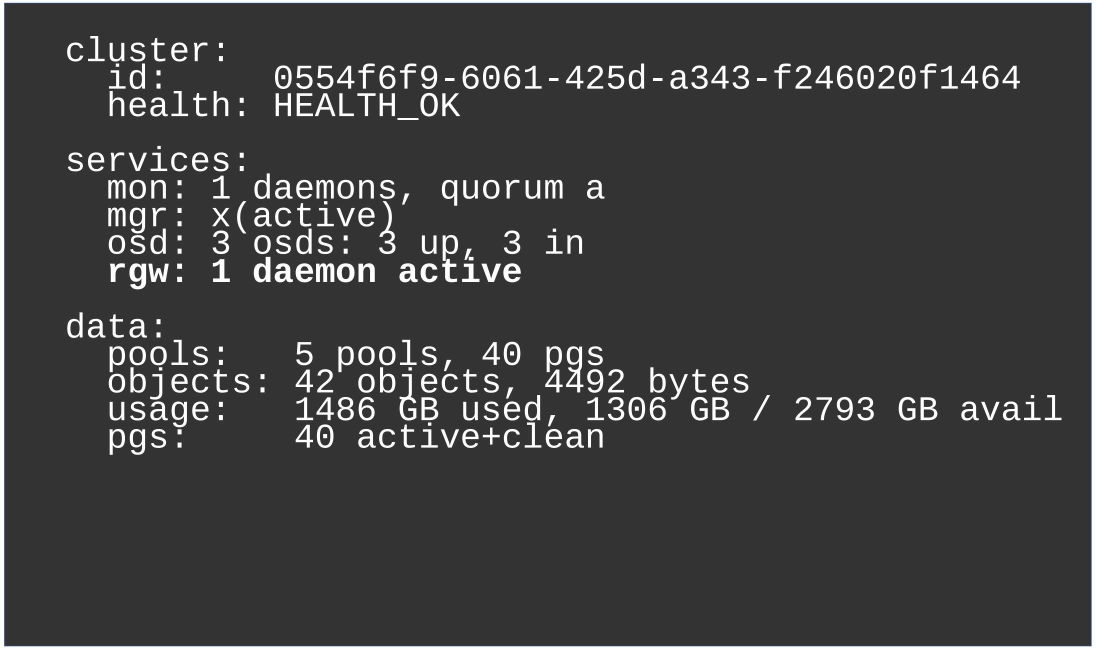
- This should be not be something the typical operator is thinking about!

MAKING PG_NUM A NON-ISSUE (MIMIC?)

- RADOS work in progress to allow PG merging
  - once pg_num can scale both up and down, most of the risk of automation goes away
- plan a mgr module to automatically adjust pg_num
  - utilization of pool (actual # of objects or bytes)
  - user intent (allow means for user to hint how much of cluster the pool or use-case is expected to consume)
- automated but conservative adjustments
  - throttle changes, just like the balancer module

SERVICEMAP

- generic facility for daemons to register with cluster
  - metadata (immutable)
    - host, version, etc.
  - status (mutable)
    - current task, progress, etc.
- in-tree users
  - radosgw
  - rbd-mirror daemon
- visibility in ‘ceph -s’
- will enable better insight into rgw multisite sync, rbd mirroring...

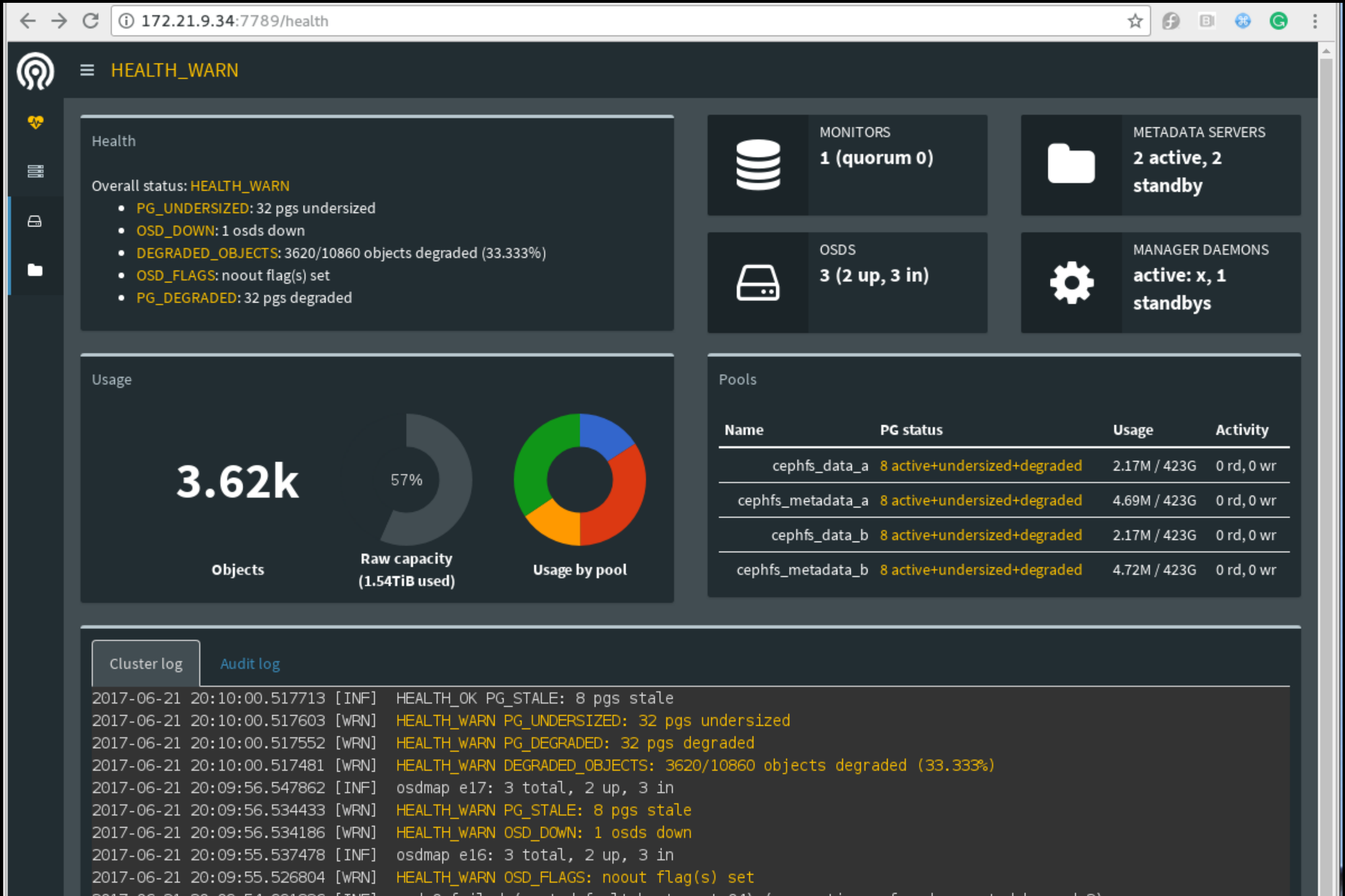


```
cluster:
  id:     0554f6f9-6061-425d-a343-f246020f1464
  health: HEALTH_OK

services:
  mon: 1 daemons, quorum a
  mgr: x(active)
  osd: 3 osds: 3 up, 3 in
  rgw: 1 daemon active

data:
  pools:   5 pools, 40 pgs
  objects: 42 objects, 4492 bytes
  usage:   1486 GB used, 1306 GB / 2793 GB avail
  pgs:     40 active+clean
```

MANAGE

DASHBOARD

- web-based UI for managing ceph
  - ‘ceph mgr module enable dashboard’
  - luminous version is read-only, no authentication
- front page similar to ‘ceph -s’ and ‘ceph -w’
- RBD
  - show pools, images
  - mirror daemons and mirroring status
- RGW
  - zonegroups, zones, daemons
- CephFS
  - file systems, clients, metadata ops sparklines, etc.



Health

Overall status: HEALTH_WARN

- PG_UNDERSIZED: 32 pgs undersized
- OSD_DOWN: 1 osds down
- DEGRADED_OBJECTS: 3620/10860 objects degraded (33.333%)
- OSD_FLAGS: noout flag(s) set
- PG_DEGRADED: 32 pgs degraded

MONITORS
1 (quorum 0)

METADATA SERVERS
2 active, 2 standby

OSDS
3 (2 up, 3 in)

MANAGER DAEMONS
active: x, 1 standbys

Usage

Pools

Name	PG status	Usage	Activity
cephfs_data_a	8 active+undersized+degraded	2.17M / 423G	0 rd, 0 wr
cephfs_metadata_a	8 active+undersized+degraded	4.69M / 423G	0 rd, 0 wr
cephfs_data_b	8 active+undersized+degraded	2.17M / 423G	0 rd, 0 wr
cephfs_metadata_b	8 active+undersized+degraded	4.72M / 423G	0 rd, 0 wr

```
2017-06-21 20:10:00.517713 [INF]  HEALTH_OK PG_STALE: 8 pgs stale
2017-06-21 20:10:00.517603 [WRN]  HEALTH_WARN PG_UNDERSIZED: 32 pgs undersized
2017-06-21 20:10:00.517552 [WRN]  HEALTH_WARN PG_DEGRADED: 32 pgs degraded
2017-06-21 20:10:00.517481 [WRN]  HEALTH_WARN DEGRADED_OBJECTS: 3620/10860 objects degraded (33.333%)
2017-06-21 20:09:56.547862 [INF]  osdmap e17: 3 total, 2 up, 3 in
2017-06-21 20:09:56.534433 [WRN]  HEALTH_WARN PG_STALE: 8 pgs stale
2017-06-21 20:09:56.534186 [WRN]  HEALTH_WARN OSD_DOWN: 1 osds down
2017-06-21 20:09:55.537478 [INF]  osdmap e16: 3 total, 2 up, 3 in
2017-06-21 20:09:55.526804 [WRN]  HEALTH_WARN OSD_FLAGS: noout flag(s) set
```

DASHBOARD

- designed for simplicity
  - rivets framework – low barrier to entry for contributors
  - shake out internal interfaces to ensure cluster state can be meaningfully surfaces in a UI
- example: pool tags
  - RADOS metadata associated with pools to identify application etc.
  - allows dashboard to identify which pools to present on RBD panel, etc.
  - will allow CLI and other tools to prevent user mistakes (e.g., reusing RBD pool for CephFS)
- out of tree management implementations awkward
  - separate tree; overhead of maintaining stable APIs
  - deployment complexity (dependencies, HA, etc.)

OPENATTIC → DASHBOARD V2

- openATTIC is SUSE’s external ceph management tool
  - featured, robust, out-of-tree
- **consensus** around developing full-featured, in-tree **dashboard v2**
  - cluster management operations (creating pools, file systems, configuring cluster, etc.)
  - embedding rich Grafana metrics dashboards (ala OpenAttic, ceph-metrics)
  - deployment tasks (expanding cluster, managing OSDs and other ceph daemons)
- initial work porting dashboard to angular2 up for review on github
- openATTIC team porting their backend API to ceph-mgr
- will become default as soon as superset of functionality is covered

PROVISIONING AND DEPLOYMENT

- dashboard v2 will include ability to orchestrate ceph itself
  - in kubernetes/openshift environments, provision OSDs, replace OSDs, etc.
  - some subset of functionality on bare metal deployments
- common tasks
  - expanding cluster to a new host or to new storage devices
  - replacing/reprovisioning failed OSDs

DEPLOYMENT TOOLS

- traditional ceph-deploy tool is very basic, limited

- ceph-ansible (Red Hat)
  - ansible-based
- DeapSea (SUSE)
  - salt-based
- (also puppet, chef, ...)

WHAT ABOUT CONTAINERS?

- ceph-ansible has basic container support
  - run daemons via docker...
- (most) people really want a container orchestrator (e.g., kubernetes)
  - stateful services (e.g., OSDs) are super annoying
  - Ceph has *lots* of stateless services (radosgw, ceph-mds, rbd-mirror, ceph-mgr. Also ganesha, samba, ...)
- real value for small, hyperconverged clusters
- container orchestrators as the new distributed OS

ROOK

- Kubernetes operator for ceph started by Quantum
  - uses native kubernetes interfaces
  - deploy ceph clusters
  - provision ceph storage (object, block, file)
- Smart enough to manage ceph daemons properly
  - don’t stop/remove mon containers if it breaks quorum
  - follow proper upgrade procedure for luminous → mimic
- Makes Ceph “easy” (for Kubernetes users)
  - control storage with kubernetes CRDs
- Plan to make Rook the recommended/default choice for ceph in kubernetes
  - dashboard will call out to kubernetes/rook to manage cluster daemons

MIMIC

MANAGEMENT
CONTAINERS
PERFORMANCE

COMING IN MIMIC

UX

- central config management
- slick deployment in Kubernetes with Rook
- vastly improved dashboard based on ceph-mgr and openATTIC
  - storage management and cluster management
- progress bars for recovery etc.
- PG merging (maybe)

Other

- QoS beta (RBD)
- CephFS snapshots
- cluster-managed NFS CephFS gateways
- Lots of performance work
  - new RGW frontend
  - OSD refactoring for ongoing optimizations for flash
  - Seastar, DPDK, SPDK

GET INVOLVED

- UX feedback wanted!
- Mailing list and IRC
  - http://ceph.com/IRC
- Github
  - https://github.com/ceph/
- Ceph Developer Monthly
  - first Weds of every month
  - video conference (Bluejeans)
  - alternating APAC- and EMEA-friendly times

- Ceph Days
  - http://ceph.com/cephdays/
- Meetups
  - http://ceph.com/meetups
- Ceph Tech Talks
  - http://ceph.com/ceph-tech-talks/
- ‘Ceph’ Youtube channel
  - (google it)
- Twitter
  - @ceph

CEPHALOCON APAC

2018.03.22 and 23

BEIJING, CHINA

https://ceph.com/cephalocon

THANK YOU

- Free and open source scalable distributed storage
- Minimal IT staff training!

Sage Weil

Ceph Project Lead / Red Hat

sage@redhat.com

@liewegas