

Morning Talks 27.9. 2017



Jiri Suchora

Just another storage

Agenda

- Introduction of CEPH, releases
- Basic design of CEPH
- Usage, modes
- (dis)Advantages
- Interconnection with other platforms
- Features
- How to start
- Lessons Learned, Perf Tunning, Monitoring, Benchmarking
- Enterprise distribution with support, real workload examples
- **DEMO**
- Questions?

Introduction of CEPH

- OpenSource project
- 88850 commits, 16847 pull requests, 885 authors (as for 26.9.2017)
- History:
 - 2007 as doctoral dissertation
 - 2012 Inktank was created
 - 2014 Acquisition by Red Hat
 - 2015 Ceph Community Advisory Board was formed for Ceph Project
 - Made by: Red Hat, Intel, SUSE, Mirantis, Cisco, IBM, CERN, Fujitsu, SanDisk, big bunch of individuals and many others...

Introduction cont.

- Name "Ceph" is a common nickname given to pet octopuses and derives from cephalopods
- Pools (logical partitions for storing objects)
- Replicated Objects
- Objects in Placement groups
- Balanced thru cluster



Stable versions

- Jewel – 10.2.x (LTS, until June 2018)
- Kraken - 11.2.x (just retired on August 2017)
- Luminous – 12.2.x(new LTS, no estimated retirement yet)
 - Recommended way for starting to play with

Design

- Cluster monitors (ceph-mon)
 - keep track of active and failed cluster nodes
- Metadata servers (ceph-mds)
 - store the metadata of inodes and directories
- Object storage device nodes (ceph-osd)
 - store the content of files in an XFS file system, or BlueStore
- RGW nodes (ceph-rgw)
 - RESTful gateways – S3 and OpenStack Swift APIs

Design cont.

- NFS Gateways
 - NFS access for RGW via NFS-Ganesha, still limited operations right now
- iSCSI Gateways
 - provide a Highly Available (HA) iSCSI target that exports RADOS Block Device (RBD) images as SCSI disks
- Ceph Manager Daemon (ceph-mgr)
 - provide WEB GUI Dashboard with additional monitoring and interfaces to external monitoring and management systems + bundle of modules (RESTful, Zabbix, Prometheus...)

Ceph-mgr daemon

The screenshot shows the Ceph Dashboard interface with the following sections:

- Health:** Overall status: **HEALTH_OK**.
- Daemons:**
 - MONITORS:** 3 (quorum 0, 1, 2)
 - METADATA SERVERS:** 1 active, 2 standby
 - OSDS:** 15 (15 up, 15 in)
 - MANAGER DAEMONS:** active: c-2, 2 standbys
- Usage:**
 - Objects:** 526k
 - Raw capacity:** (116GiB used) 8%
 - Usage by pool:** A large pink circle icon.
- Pools:** A table listing pools with their PG status, usage, and activity.
- Logs:** A log viewer showing recent messages from the cluster log and audit log.

Name	PG status	Usage	Activity
cephfs_data	256 active+clean	0 / 1.38T	0 rd, 0 wr
cephfs_metadata	16 active+clean	2.24k / 1.38T	0 rd, 0 wr
.rgw.root	8 active+clean	7.57k / 1.38T	0 rd, 0 wr
primary.rgw.control	8 active+clean	0 / 1.38T	0 rd, 0 wr
primary.rgw.meta	8 active+clean	2.43k / 1.38T	0 rd, 0 wr
primary.rgw.log	8 active+clean	50 / 1.38T	0 rd, 0 wr
primary.rgw.buckets.index	32 active+clean	0 / 1.38T	0 rd, 0 wr
primary.rgw.buckets.data	256 active+clean	526M / 1.38T	0 rd, 0 wr

Cluster log:

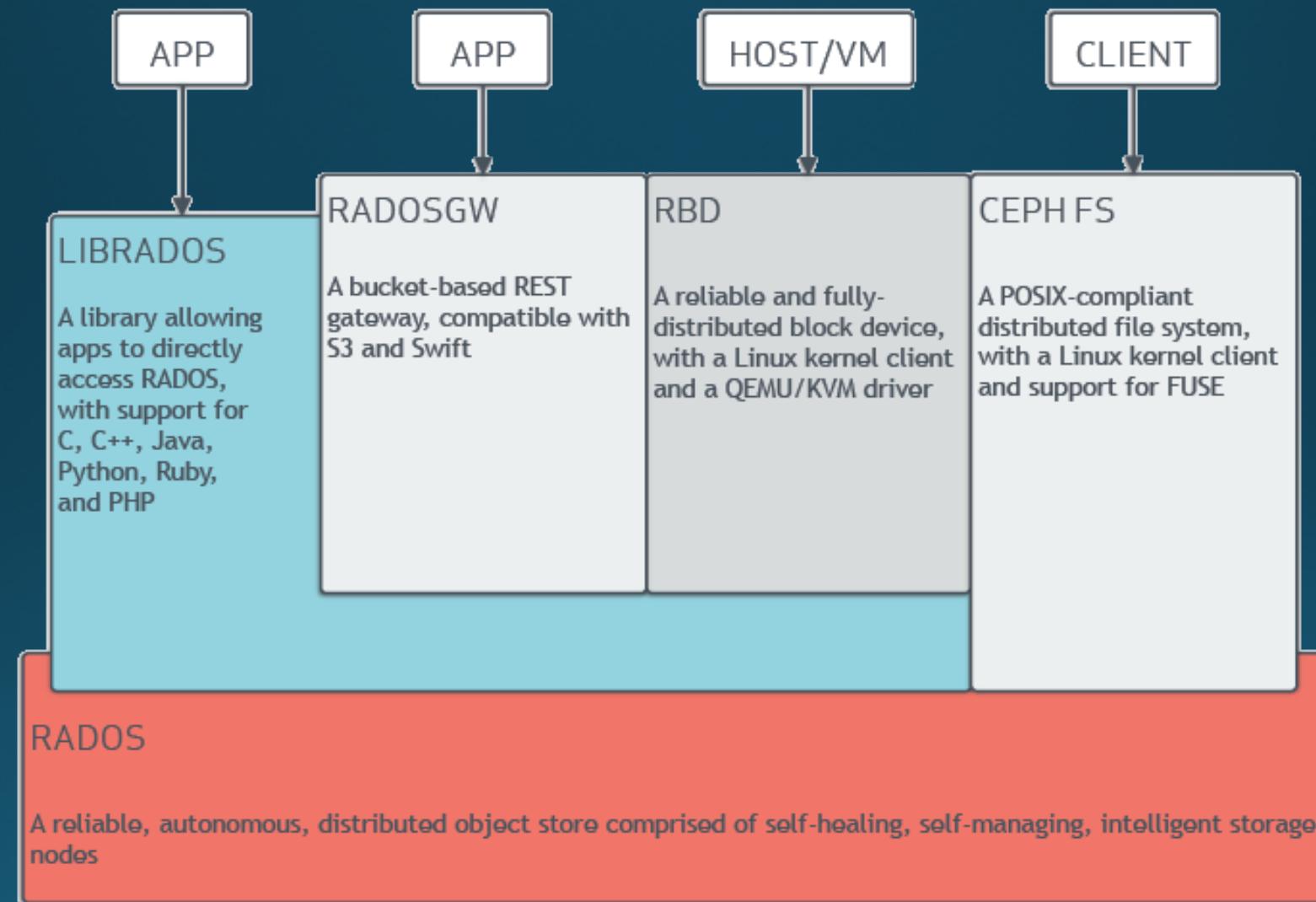
- 2017-09-24 22:02:44.980066 [INF] Manager daemon c-2 is now available
- 2017-09-24 22:02:44.147216 [INF] Activating manager daemon c-2
- 2017-09-24 22:02:44.147124 [INF] Active manager daemon c-2 restarted

Audit log:

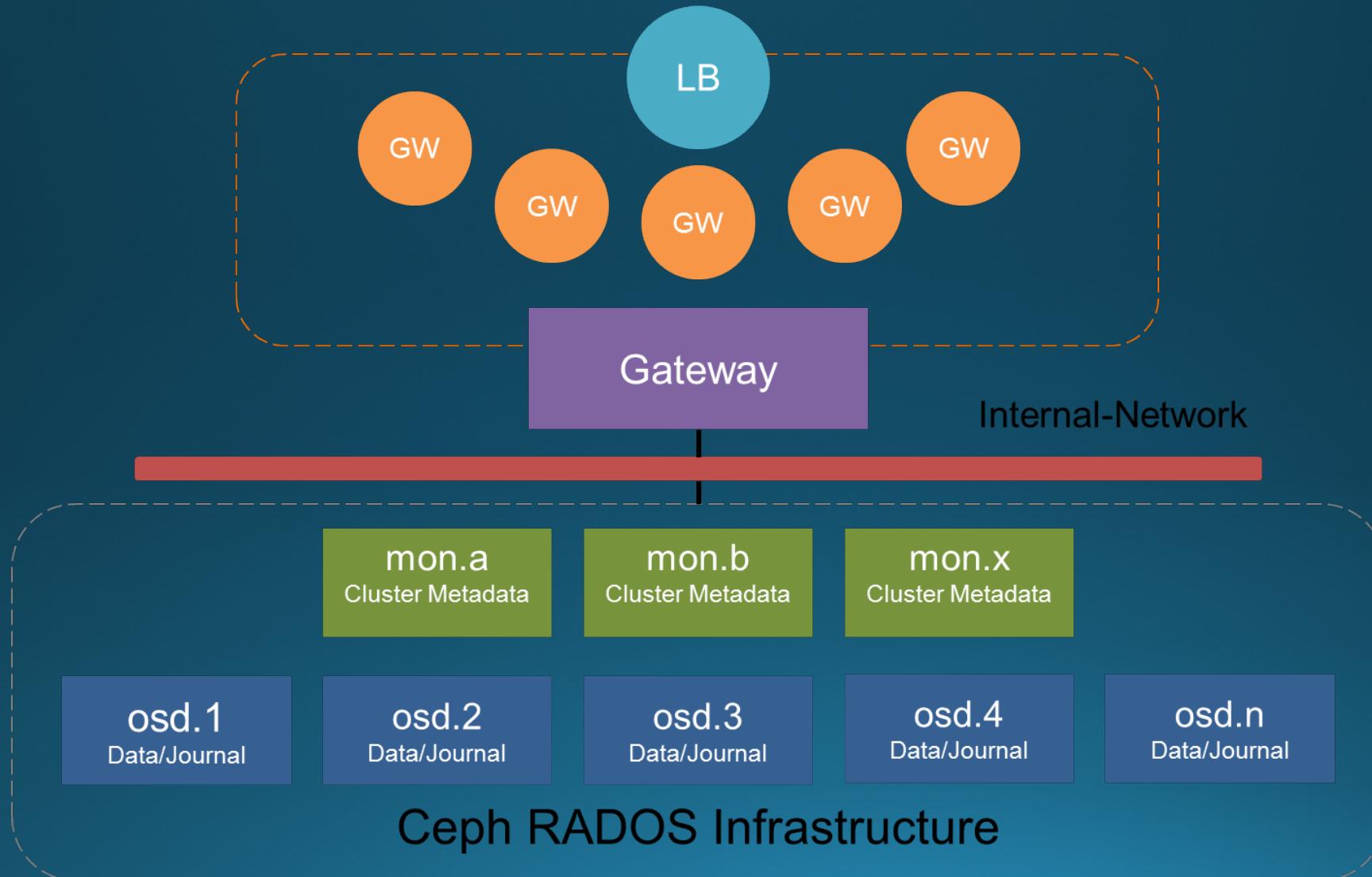
Usage, access to data

- Librados (library for direct access – support of Java, Ruby, Python, C,C++ ..)
- RBD (RADOS block device)
- S3 (AWS S3 API)
- Swift
- CephFS (POSIX compatible, mounts as filesystem)
- iSCSI(via iSCSI gateway)
- pNFS (ganesha)

Usage, modes



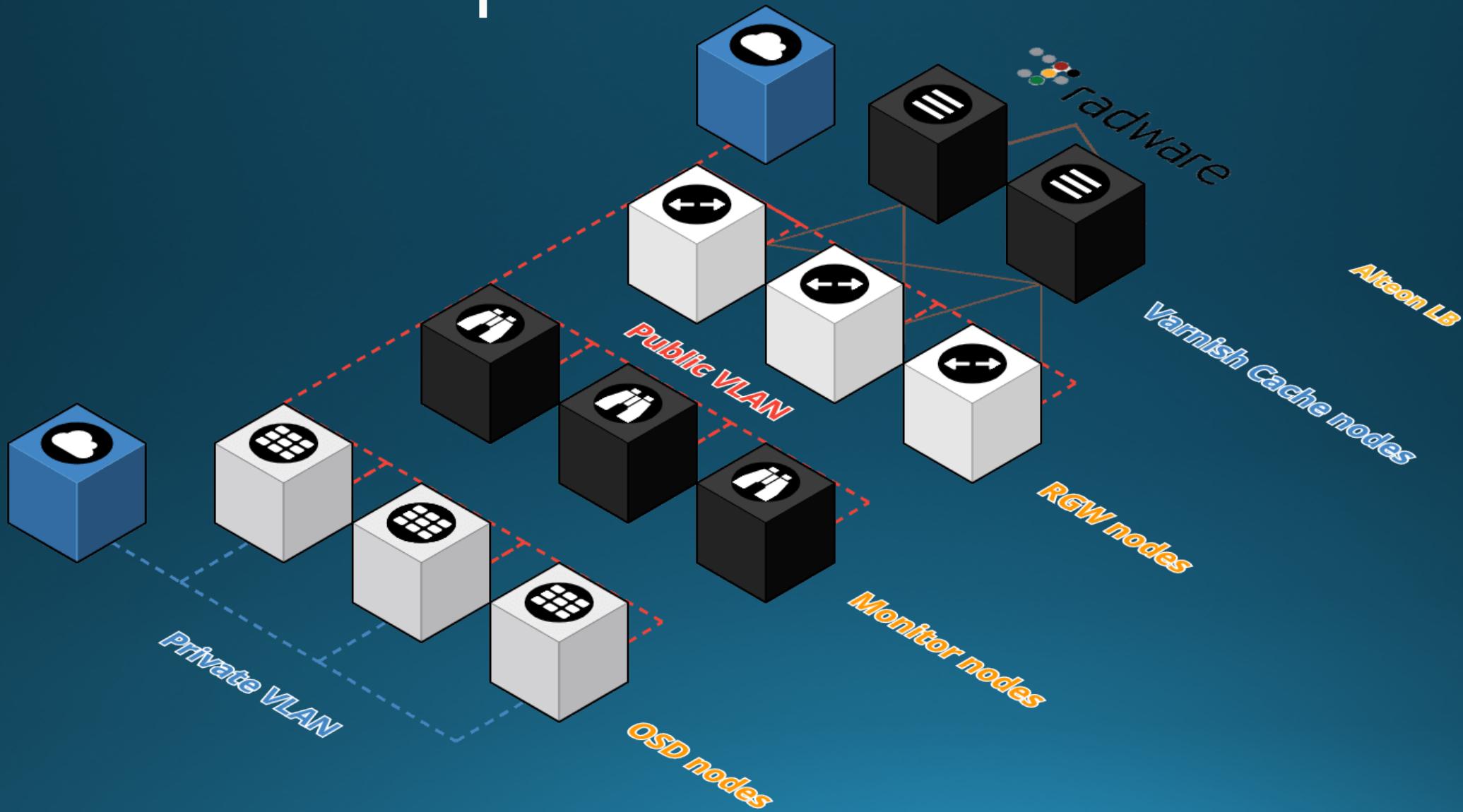
S3 access via RGW, LB appliance



S3 access via RGW, other ways?

- Haproxy instead of LB appliance (cost)
 - At least 2 nodes
 - Keepalive between them
- Varnish Cache
 - If RAM is not a problem
 - At least 2 nodes
 - LB in the front

RGW example



(dis)Advantages

- Scale-out vs Scale-up
- Fast self healing
- Commodity hardware
- Prevent vendor lock-in
- Open Source where feasible
- Reduce Total Cost of Ownership (TCO)
- Has to be learned, adopted
- Needs change of mind (conservative thinking)
- Not cure for everything -really depends on case!

Scalability

CERN as example:

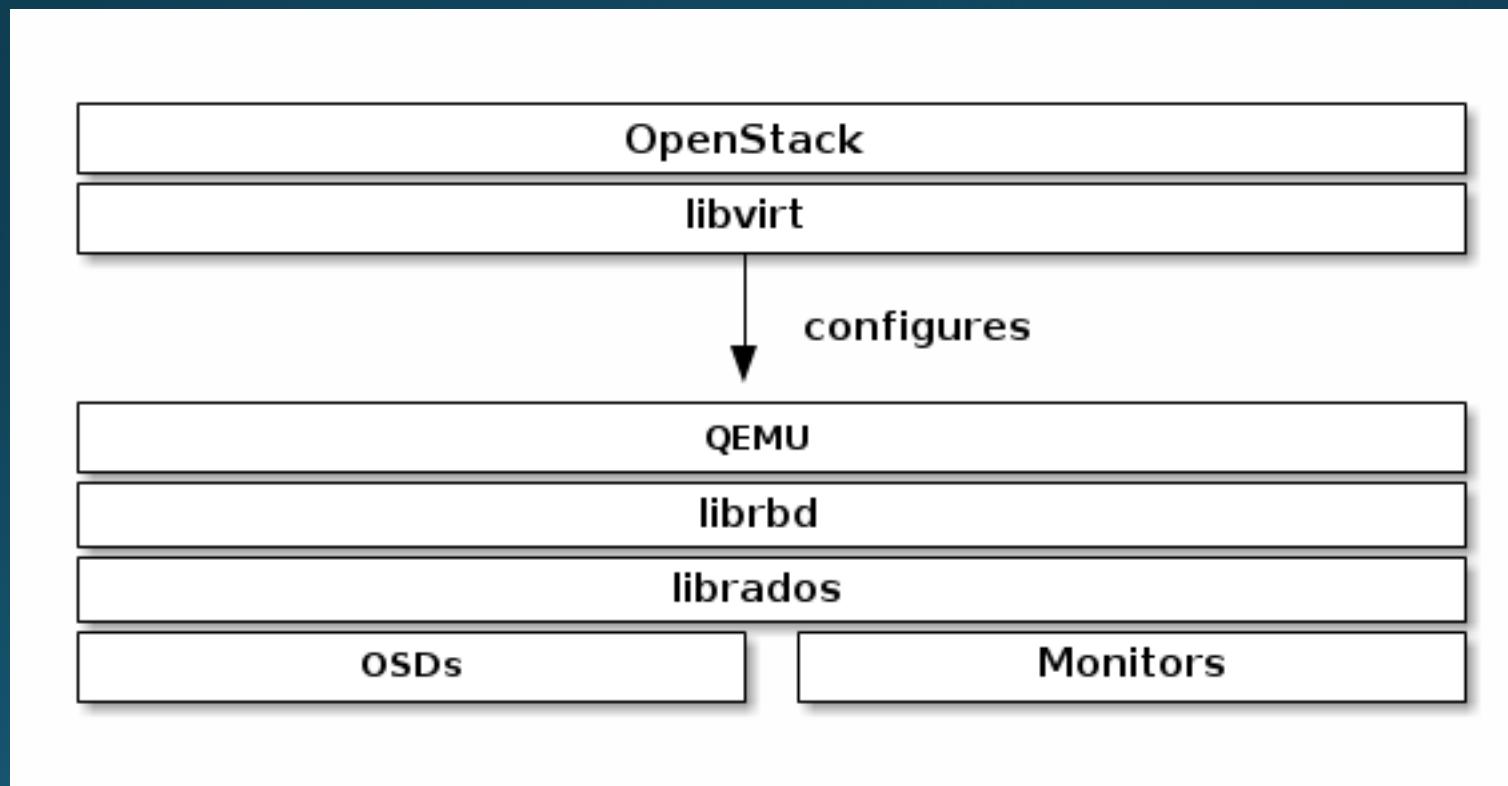
- 225 servers
- 48 6TB hard drives per server
- 10,800 OSDs in total, nearly 65 petabytes!
- 3 monitors and managers, collocated with OSDs

Interconnection with other platforms

- OpenShift, Kubernetes
- OpenStack (Cinder, Glance)
- RHEV/oVirt
- Apache CloudStack
- OpenNebula
- Ganeti
- Proxmox
- Dovecot

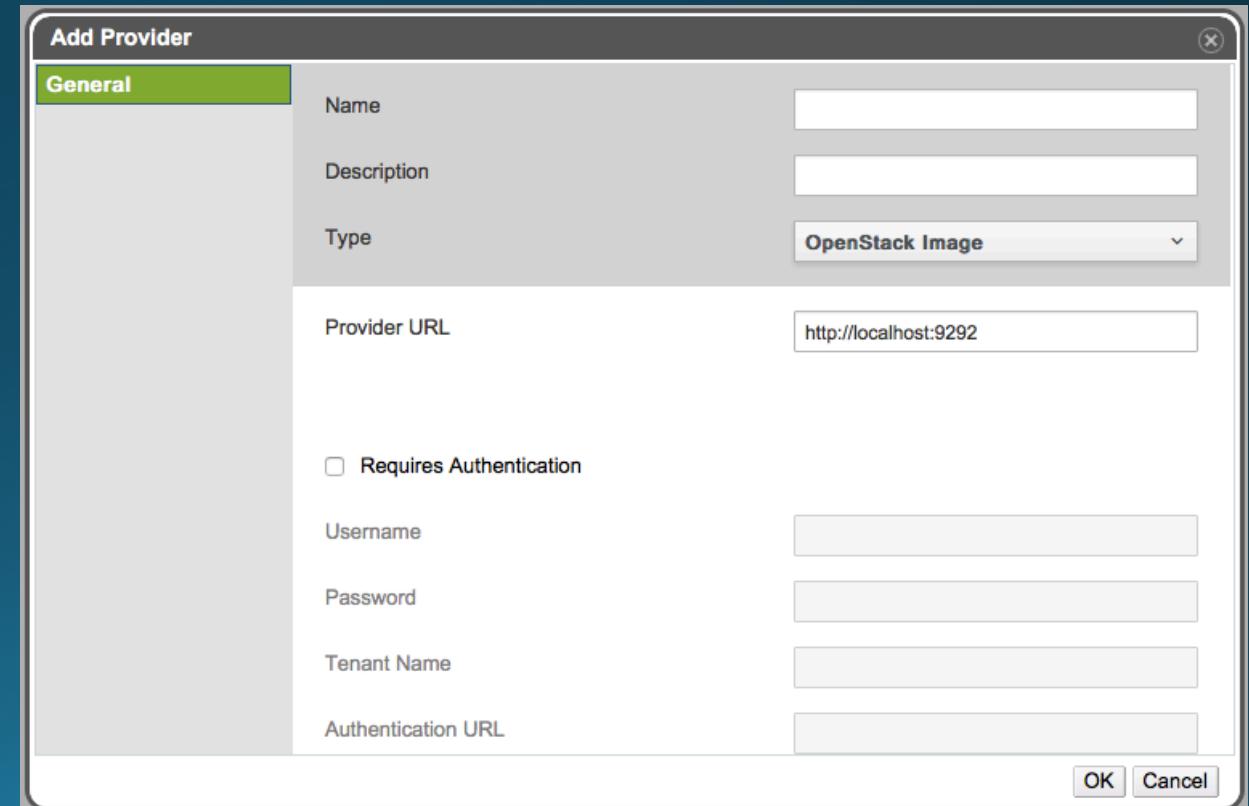
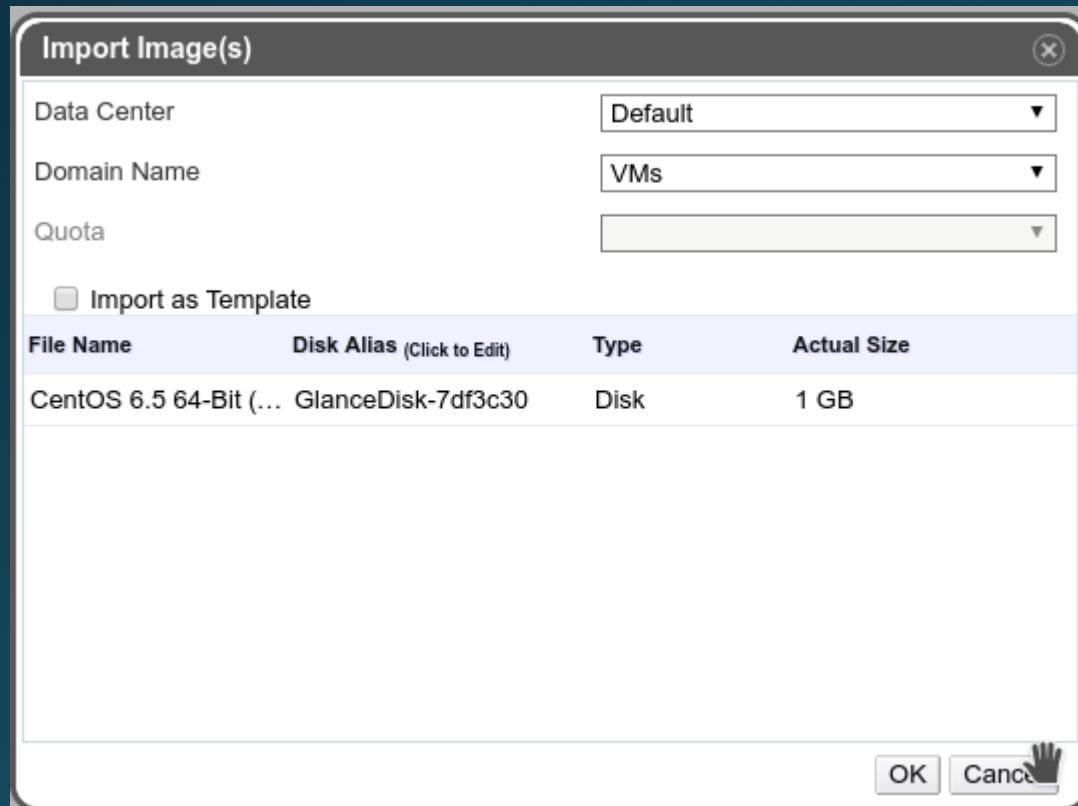
Ceph with OpenStack

- Cinder, Glance, via libvirt/librbd



Ceph with RHEV/oVirt

- through libvirt, which configures the QEMU interface to librbd



Ceph with OpenShift

- Persistent storage using Ceph RBD
- Persistent volumes (PVs) and persistent volume claims (PVCs) can share volumes across a single project
- Access via authorization key from CEPH
- Security (diff from NFS,GlusterFS) - user and group IDs defined in the pod definition or docker image are applied to the target physical storage

Ceph with OpenShift

The screenshot shows the OpenShift web interface for the 'WEB' project. The left sidebar includes links for Home, Overview, Applications, Builds, Resources, Storage (which is selected and highlighted in blue), and Monitoring. The main content area is titled 'Storage' with a 'Learn More' link. It features a 'Filter by label' input field and an 'Add' button. Below this is a section titled 'Persistent Volume Claims' with a 'Create Storage' button. A table lists five PVCs:

Name	Status	Capacity	Access Modes	Age
pvc-ceph-04	✓ Bound to volume pv-ceph-02	5 GiB	RWO (Read-Write-Once)	2 months
pvc-ceph-03	✓ Bound to volume pv-ceph-01	5 GiB	RWO (Read-Write-Once)	2 months
pvc-ceph-02	✓ Bound to volume pv-ceph-03	5 GiB	RWO (Read-Write-Once)	2 months
pvc-ceph-01	✓ Bound to volume pv-ceph-04	5 GiB	RWO (Read-Write-Once)	2 months
pvc-nfs-04	✓ Bound to volume pv-nfs-3	10 GiB	RWX (Read-Write-Many)	2 months

Ceph with OpenShift

The screenshot shows the OpenShift web interface with the following details:

- Header:** Project **WEB** (dropdown), Add to project, Help (dropdown), User (jiri) (dropdown).
- Sidebar:** Home, Overview, Applications, Builds, Resources.
- Secrets Page:** Title **Secrets** (Learn More), Create Secret button.
- Source Secrets:** A table with columns Name, Type, and Created.

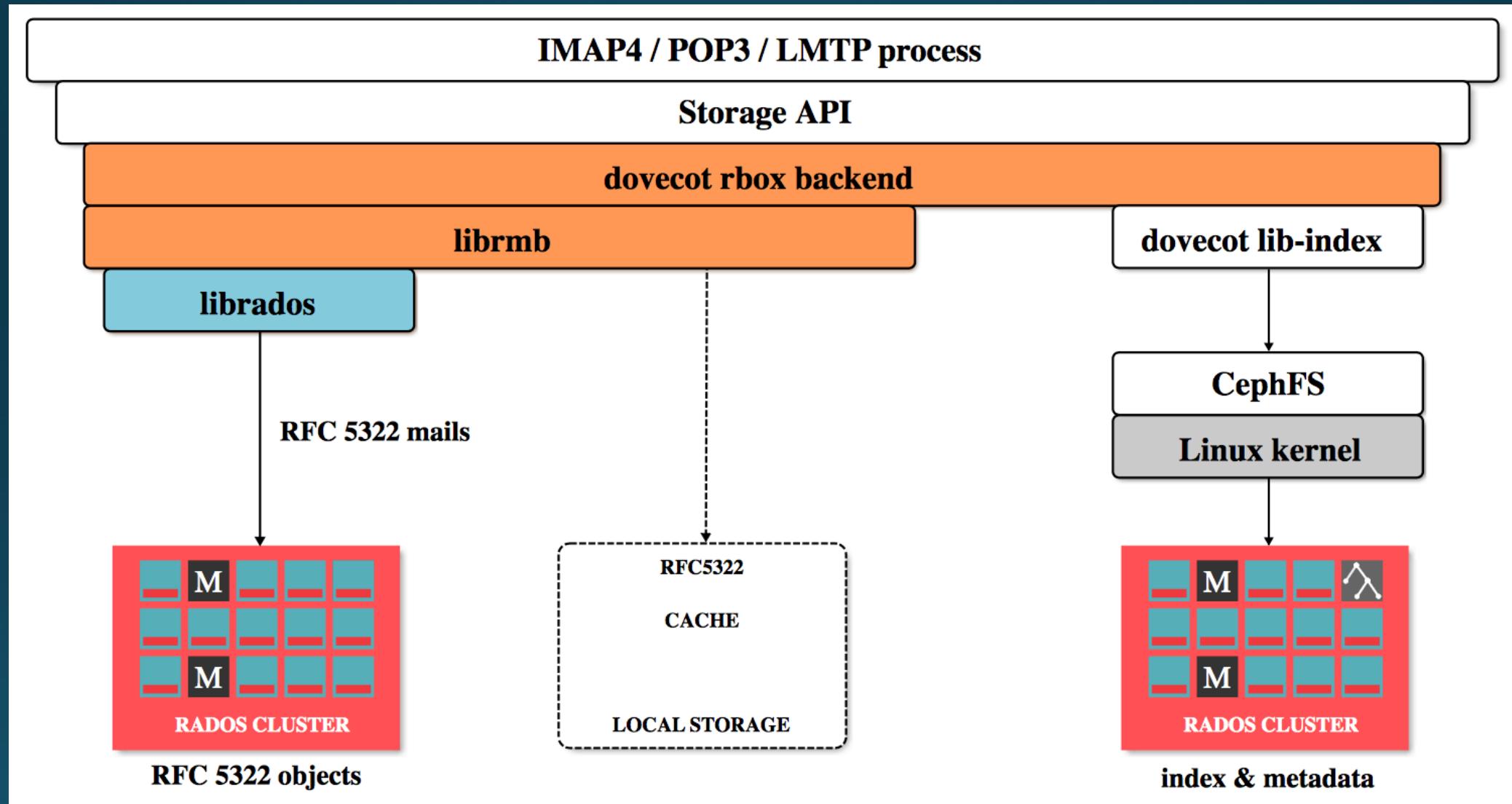
Name	Type	Created
ceph-secret	Opaque	2 months ago
- Image Secrets:** A table with columns Name, Type, and Created.

Name	Type	Created
builder-dockercfg-81bz	kubernetes.io/dockercfg	2 months ago

Ceph with Dovecot

- open-source IMAP and POP3
- 72% market share (openemailsurvey.org, 02/2017)
- Objectstore plugin obox (S3/swift), not opensource, Dovecot Pro
- Librmb (layer between dovecot backend and librados)
 - Deutsche Telekom, SUSE...
 - OpenSourced (<https://github.com/ceph-dovecot/dovecot-ceph-plugin>)
 - Being implemented in Deutsche Telekom
 - 1.3 petabyte net storage
 - ~39 million accounts, 6,7 billion emails

Ceph with Dovecot



Other features...

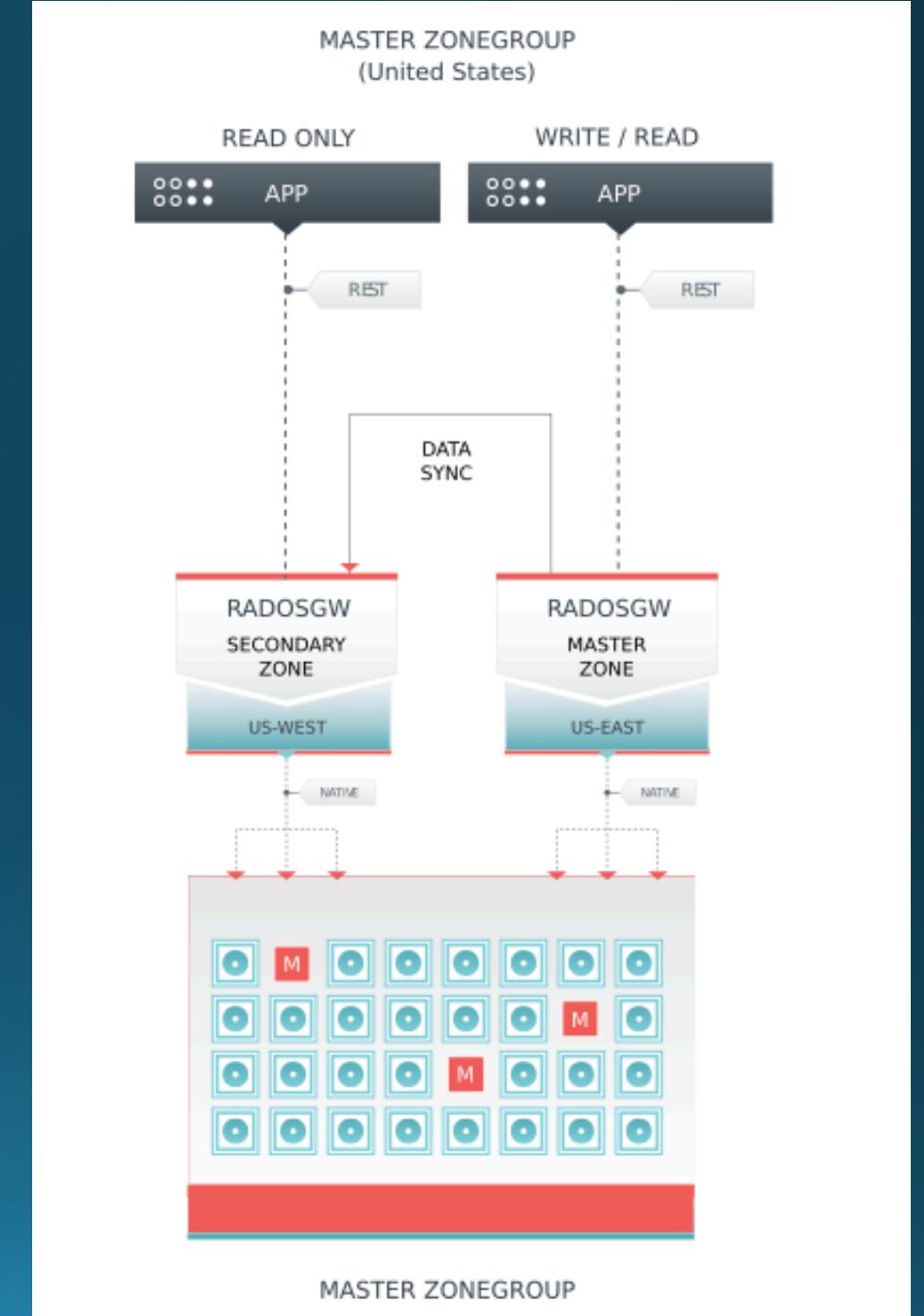
- Multisite Deployment (Realms, Zone groups, Zones), Disaster Recovery to Second site
- Snapshots (Pools, Objects)
- Quotas per:
 - CephFS directories,
 - RGW Buckets
 - RGW Users
- RGW Metadata Search – via Elasticsearch (from Luminous)
- Bluestore backend – better perf, inline compression...

Other features

- Storage tiering (eg.)
 - fast: all flash
 - medium: disks accelerated by some flash journals
 - slow: archive disks with collocated journals
 - Tiered CRUSHmap (eg.)
 - Bigger than 6 TB → Archive drive
 - Between 1.6TB and 6TB → Disk with flash journal
 - Smaller than 1.6TB → SSD assumed

Multisite Deployment

- Zone: Logical Group of one or more RGW instances
 - Zone group: one or more group, region
 - Realm: container for one or more Zone groups
-
- Synchronous/Asynchronous
 - Failover possible
 - Migration from Single to Multisite possible



How to start

- How to deploy?
 - Ceph-ansible (<https://github.com/ceph/ceph-ansible>)
 - Ceph-deploy (<https://github.com/ceph/ceph-deploy>)
 - DeepSea and Salt (for SUSE Enterprise Storage)
 - Crowbar (<http://crowbar.github.io>)
 - Ceph-docker (<https://github.com/ceph/ceph-docker>)
 - Manually

How to start

- Single VM -> just for testing
- Ceph in docker – all in one -> just for testing
<https://hub.docker.com/r/ceph/demo/>
- At least 3-3-3 for production

Lessons learned

- **Definitely not** VMs for production!!! (at least for OSD nodes)
- Local storage, **not** SAN/NAS (esp. with own logic, tiering)!!!
- SSD, SSD, SSD if possible ☺ (at least for journals)
- Perf. Testing before production – try to simulate workload as much as possible
- Tuning – there is no default recipe for everything, listen your CEPH cluster and tune it as you go!

Monitoring, Benchmarking

MONITORING

- Zabbix (native plugin for MGR module from Luminous)
- Nagios
- Grafana+Prometheus (native plugin for MGR module from Luminous)
- ELK

BENCHMARKING

- COSBench (<https://github.com/intel-cloud/cosbench>)
- CBT (<https://github.com/ceph/cbt>) - FIO on steroids
- AB (ApacheBench), WRK (<https://github.com/wg/wrk>)

COSBench tool by Intel

General Report

Op-Type	Op-Count	Byte-Count	Avg-ResTime	Avg-ProcTime	Throughput	Bandwidth	Succ-Ratio
op1: read	563 ops	563 KB	24,3 ms	24,28 ms	112,02 op/s	112,02 KB/S	99,82%
op2: write	62 ops	62 KB	22,02 ms	22,02 ms	12,35 op/s	12,35 KB/S	100%

The snapshot was taken at 21:21:57 with version 62.

Stages

Current Stage	Stages completed	Stages remaining	Start Time	End Time	Time Remaining	
w4-s1-init	init	1 wks	7 wkrs	init	completed	view details
w4-s2-main	main	1 wks	3 wkrs	read, write	running	view details

Performance Graph

throughput Graph

This graph plots throughput in op/s against time. The x-axis represents time from 0 to 62 seconds. The y-axis represents throughput from 0 to 150,000 op/s. Two data series are shown: read (orange line with circles) and write (blue line with squares). A dashed horizontal line at approximately 100,000 op/s represents the average throughput. Several data points are highlighted with callouts: (7.079, 775), (18,352, 148,186), (50, 91,303.17), and (62, 10,124.33). The read series shows higher peaks and more variability than the write series.

resTime Graph

This graph plots response time in sp (seconds per operation) against time from 0 to 62 seconds. The y-axis ranges from 0 sp to 500,000 sp. The read series (orange line with circles) has a single prominent peak at approximately 40,696 sp around time 6. The write series (blue line with squares) has two major spikes: one at approximately 413,677 sp around time 52, and another at approximately 28,788.67 sp around time 60. The response times for both series remain very low for most of the duration.

Enterprise

- Red Hat Ceph Storage (RHEL, Ubuntu)
- SUSE Enterprise Storage (SLES)
- OpenStack distributions (Mirantis, Canonical, ...)

Where Ceph is running?

- CERN
- Dropbox
- Bloomberg
- Deutsche Telekom
- Cisco
- ...
- Openstack Clouds (by: Mirantis, Canonical Cloud,...)

DEMO

- MGMT Web Dashboard
- COSBench (web controller, XML examples)
- RBD device for linux client
- CephFS for linux client
- Disaster test show

Location of this PDF...

- <https://github.com/yrjo/ceph-tweaks/blob/master/mtalks.pdf>

Do you need help?

- Trainings?
 - www.datascript.cz - OpenSource Trainings
 - Contact at: info [AT] datascript [DOT] cz
- Help needed?
 - PoC
 - Consultations
 - Design, build, support
- Contact me:
 -  jirka [AT] chlivek [DOT] com
 -  <https://www.linkedin.com/in/jirisuchora/>

Questions?



Thanks
For All the Fish!