Ceph - a software defined storage system

Ceph components and our deployment

CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

ceph

Presentation overview

Introduction:

- Ceph history
- Basic concepts

Ceph Architecture:

- Storage types
- Services
- Hardware for a Ceph cluster

Our Ceph systems:

- *Cephyr*
- *Mimer-Ceph*

Working with Ceph

Ceph

Ceph is a **distributed storage system** designed to provide excellent performance, reliability, and scalability.

Ceph is often used for **object storage, block storage, and file systems**.

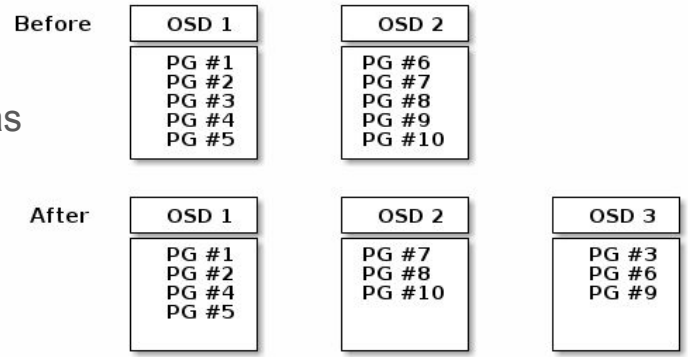
Ceph history

Ceph is the result of the PhD work by Sage Weil.
Ceph support is included in the Linux kernel since 2010.

Ceph have been the default choice of storage solution for use with cloud solutions like OpenStack for a very long time.

Major milestones:

- 2014, Firefly: Erasure coding available (only fully replicated up to this point)
- 2016, Jewel: CephFS declared stable
- 2017, Kraken: BlueStore declared stable (earlier FileStore was used)
- 2017: Luminous: pg-upmap balancer available



Ceph history - C3SE / e-Commons

Already the first generation of SNIC Science Cloud used Ceph (for block storage).

When designing *Vera* (2018), we chose to also use the filesystem part of Ceph (CephFS) for cluster storage. We were quite early taking this step, but it worked out well.

Cephyr was deployed using Ceph Ansible on CentOS 7. Ceph version: Luminous

It was used for *Vera* as cluster filesystem (/cephyr) for home directories and project storage, as block and object storage for our node of *SNIC Science Cloud (SSC)* and as block storage for *Swestore* pools (*Mare4)*.

Cephyr has been updated several times and is now running Rocky 8 and Ceph Reef.

Ceph history - C3SE / e-Commons

Mimer-Ceph is part of the *Alvis* phase 2 procurement (2020-2021) and was designed for use as a second tier behind our WEKA storage system *Mimer (Mimer-Weka)*.

When *Cephyr* was getting closer to end of support contract we migrated the both the /cephyr filesystem (summer 2023) as well as the block storage for *Mare4* to *Mimer-Ceph*.
The west node of *SSC* was brought down for major updates and redeployment during spring 2024, which then naturally moved to *Mimer-Ceph*.

Mimer-Ceph was deployed on CentOS 8 using Ceph Pacific, and have been updated to now run Ceph Reef on Rocky 9.

The current stable version of Ceph is Squid. Our version, Reef will be EoL in August 2025, we will therefore plan for updates during the summer.

Ceph concepts

Ceph is at the bottom an object storage system, all storage types are built up of collections of objects. These can be replicated and distributed across the cluster in different ways.

A core concepts in Ceph is **RADOS** - a **R**eliable, **A**utonomous, **D**istributed, **O**bject-storage service of **S**elf-healing storage nodes. (https://ceph.io/assets/pdfs/weil-rados-pdsw07.pdf)
RADOS is the mechanism for *storing* data.

The second important concept is **CRUSH** - Controlled, Scalable, Decentralized Placement of Replicated Data. (https://ceph.io/assets/pdfs/weil-crush-sc06.pdf)
While traditional storage systems depend on centralized services to keep track of where different pieces of data is stored, CRUSH provides an algorithm, used by clients as well as servers, for *calculation* of the *placement*.

The CRUSH rule provide a CRUSH map, that is updated and distributed whenever something change in the storage cluster.

Ceph concepts

Ceph virtually and dynamically partitions the storage using storage **pools**. When creating a pool you define the **storage class** (HDD, SSD) and redundancy level (**replicated**, **erasure code** profile) of the pool. These can not be changed after creation of the pool.

You also specify the number of (initial) **placement groups** (pg), CRUSH is then used to put objects in placement groups. The number of placement groups can be updated (at runtime) for a pool, and an auto-scale functionality is available (using pg-upmap).

For **replicated** pools N-number of copies of data are created and distributed over different placement groups.

For **erasure coded** pools data is split into K chunks and M parity chunks (K+M), these are distributed over different placement groups.
When you create an erasure code profile, you also specify **failure doman** (host, chassis etc.)

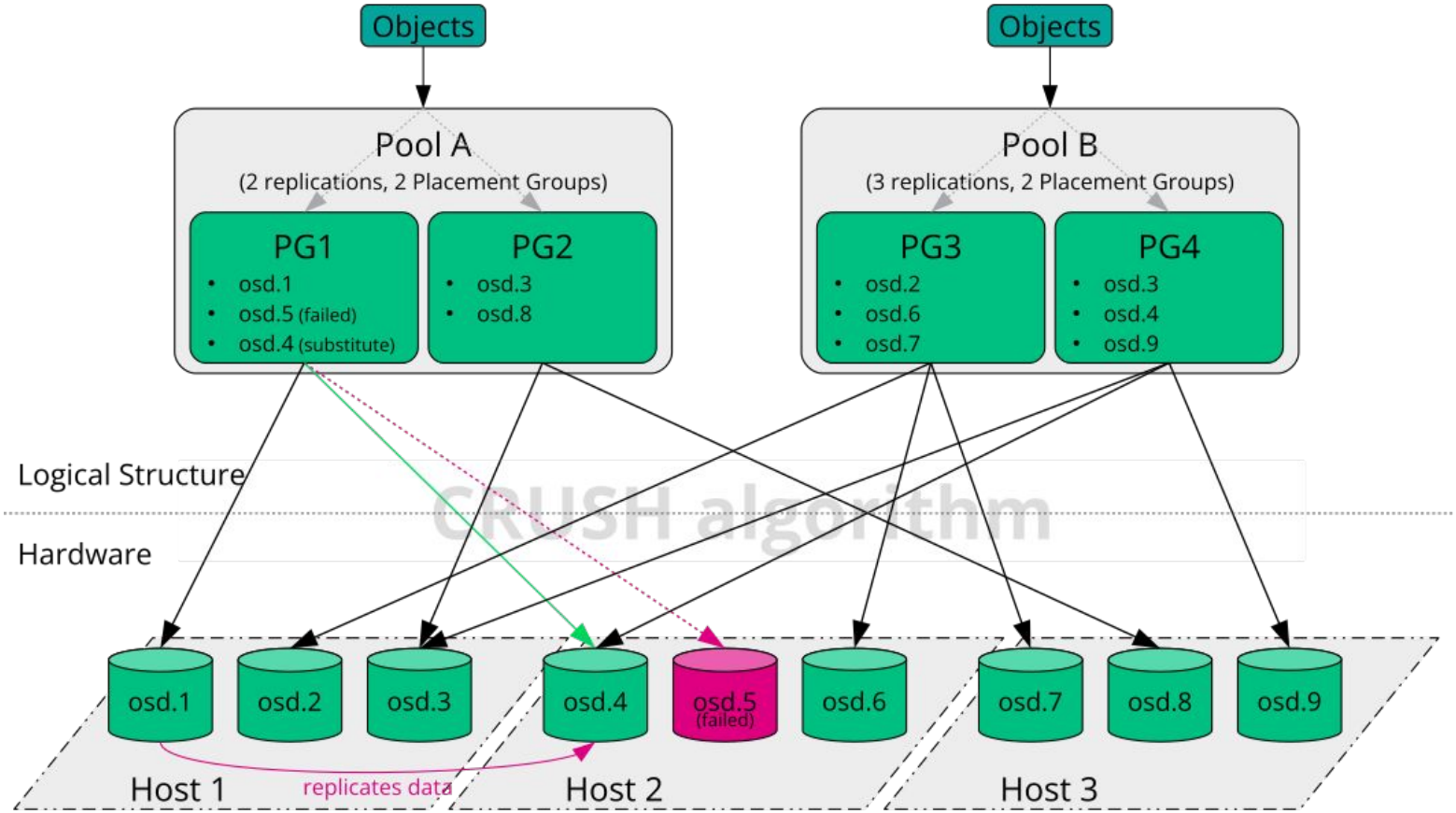


Image from SUSE: https://documentation.suse.com/ses/7/html/ses-all/images/data-structure-example.png

Storage types provided

- Block storage: RBD (RADOS Block Device)
- Object storage compatible with S3: RGW (RADOS GateWay)
- Native filesystem: CephFS
- CephFS can also be exported as NFS (uses Ganesha)
- An exporter for SMB is in development

Block Storage

- Came first, in the 1960s
- Are HDDs are SSDs that are physically attached to servers
- Presents the raw blocks to the server as a volume
- It is presented as a raw device (a block device)
- Can be used :
  - with filesystem
    - raw block storage can first be **partitioned**
    - partitioning defines logical sections of the storage
    - volume or partition is then **formatted** (Linux: ext4, xfs, btrfs Windows: NTFS, FAT32, exFAT)
  - without filesystem
    - Some applications use raw block storage directly (e.g., databases or virtual machine disk images, high-performance applications) for performance reasons (raw device mapping or block device access)

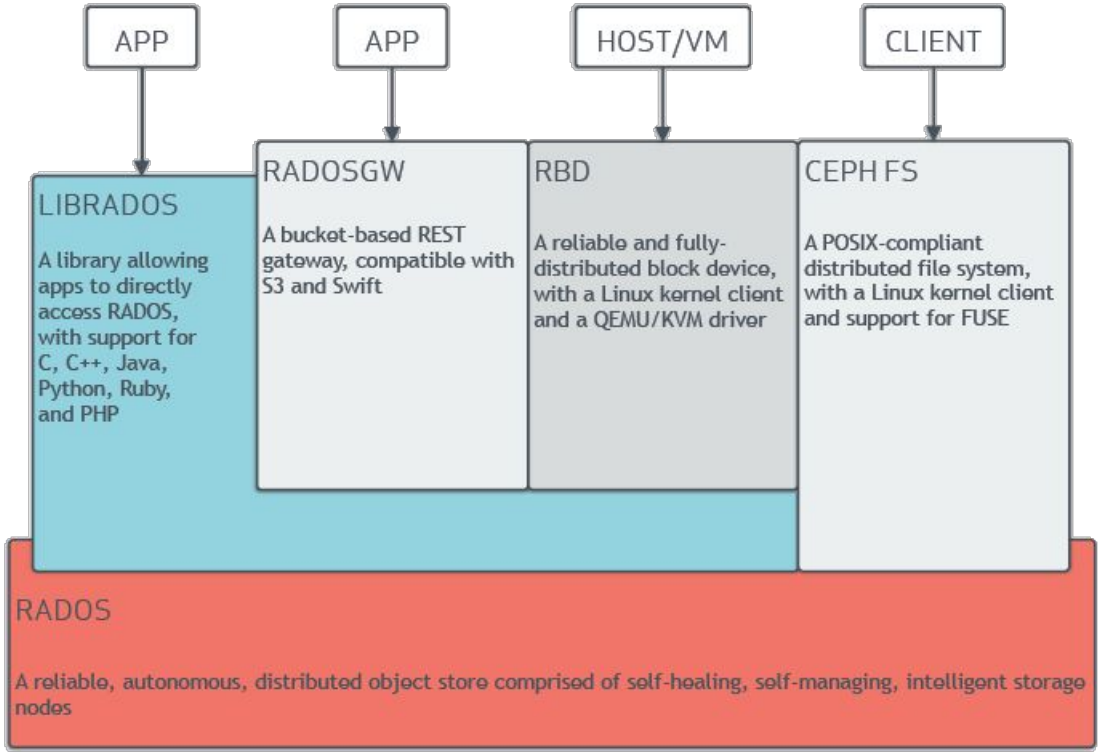
File Storage

- Is built on top of block storage
- Is a file level storage
- Stores data as files organized in a hierarchical structure (directories, subdirectories)
- Is not to be confused with filesystem
    - it **organizes data blocks** into files and directories **on a disk or partition.**
- Is a storage service or technology that allows you to store and retrieve files over a network or locally
- Simplicity makes it a great solution for sharing a large number of files and folders within an organization
- Provides access to files (with metadata) and handles file-sharing, permissions, locking, etc
- In most cases, especially in network-based file storage (like NFS, SMB, NAS), file storage is associated with a file server
- When accessed locally the OS itself plays the role of managing file storage

Object Storage

- Came last, New Kid On The Block
- No hierarchical directory structure
- Stores all data as objects in a flat structure
- Designed for **unstructured data** such as media, documents
- Logs, backups, application binaries and VM images
- Does deliberate tradeoffs for high durability, vast scale, and low cost
- Relatively “cold” data (warm?) and is mainly used for archival and backup
- Data access is normally provided via a RESTful API, relatively slow compared to other storage types
- When a portion of the file is updated, an entire object needs to be updated, unlike in block- or file storage, where only the corresponding block or part of a file is updated
- Hence object storage is well suited for the write-once and read many applications (static content, photo or video repository, data-sets)

Ceph as Software Defined Storage



The Ceph services

The main Ceph services are:

- MON: The Ceph monitor keeps track of status etc. An odd numbers of MON:s are required (for quorum). (3 in *Cephyr*, 5 in *Mimer-Ceph*)
- MGR: The Ceph manager performs all Ceph cluster operations. It have several modules, both mandatory and optional. Only a few MGR:s are needed. (3 in *Cephyr*, 2 in *Mimer-Ceph*)
- OSD: The Object Storage Daemon service handles the actual storage devices and storing of data objects. There is usually a one-to-one mapping between physical disks and OSD daemons.
- RGW: The Ceph RADOS Gateway services provide endpoints for Object storage, like S3. You can have one or more RGW:s per target (pool), and one RGW can serve multiple targets.
- MDS: The Ceph MetaData Service is required when you have a CephFS. Each CephFS need their own MDS service(s). You can have one or more MDS:s per filesystem (/cephyr have three, two active one standby)

The Ceph services

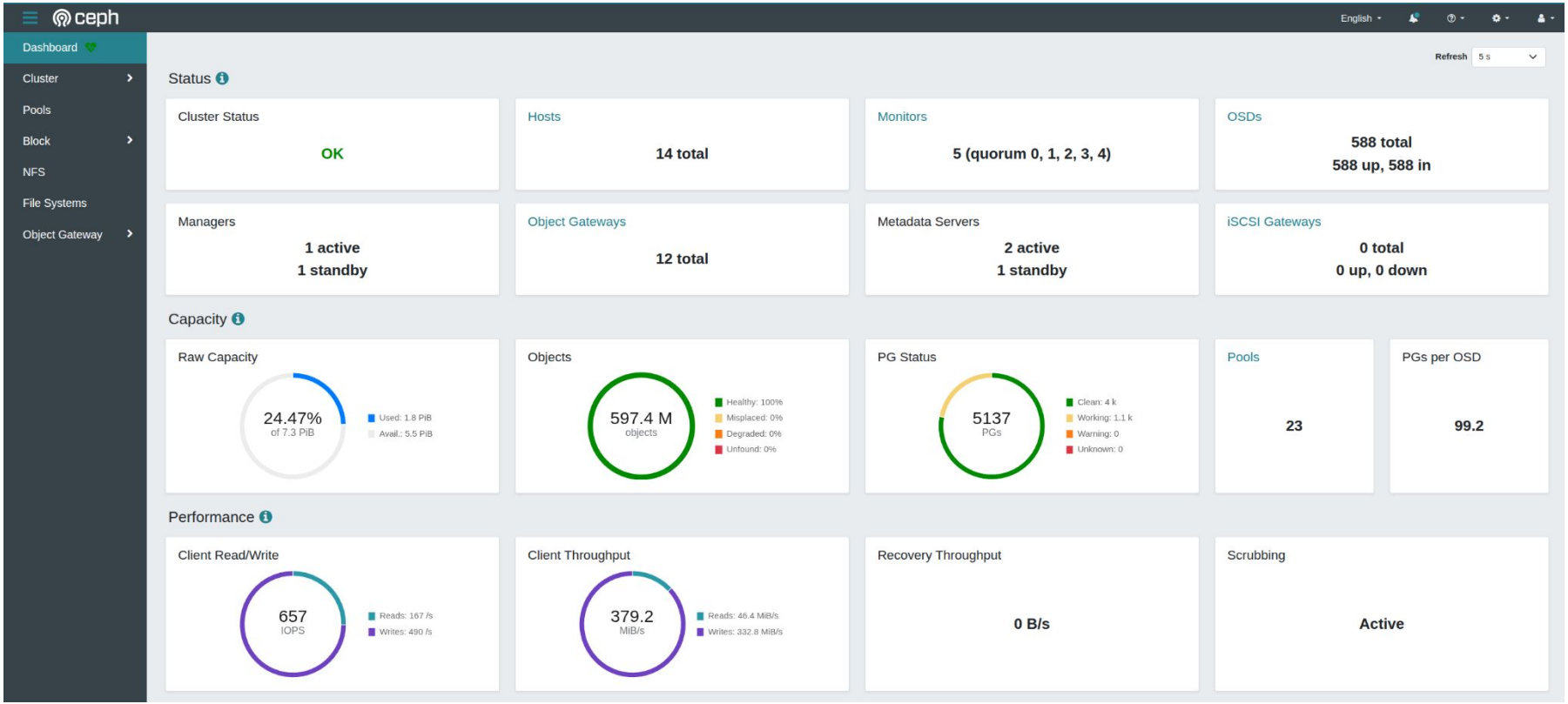
Additional services that can be found in a Ceph cluster are:

- NFS: Ganesha based services to export a filesystem over NFS. Can be deployed with High Availability.
- iSCSI gateway

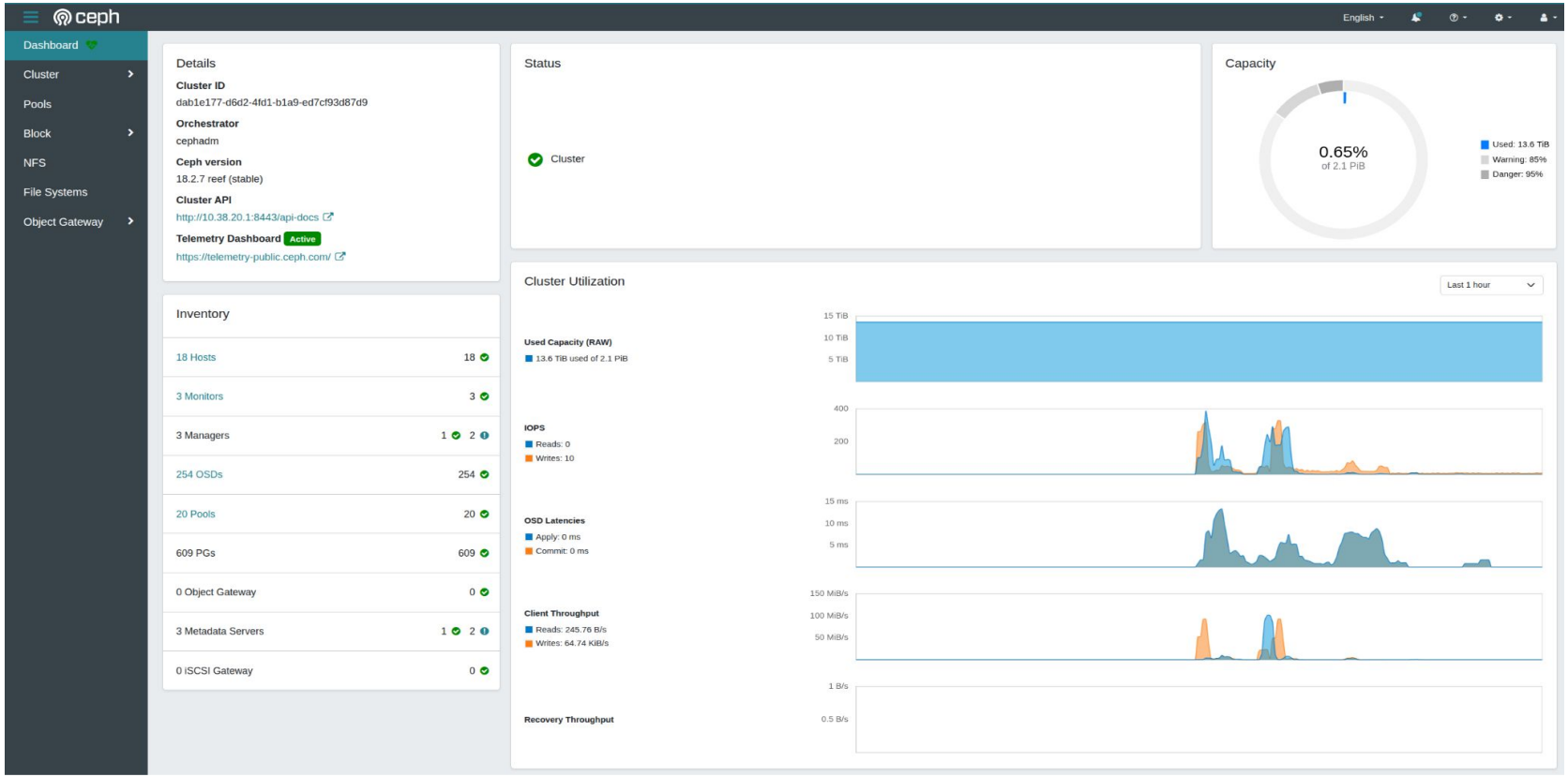
In addition several other tools are used:

- Prometheus and node-exporter to gather data
- Grafana for displaying data (in the Dashboard)

The Ceph Dashboard



The Ceph Dashboard



CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

Hardware for a Ceph cluster

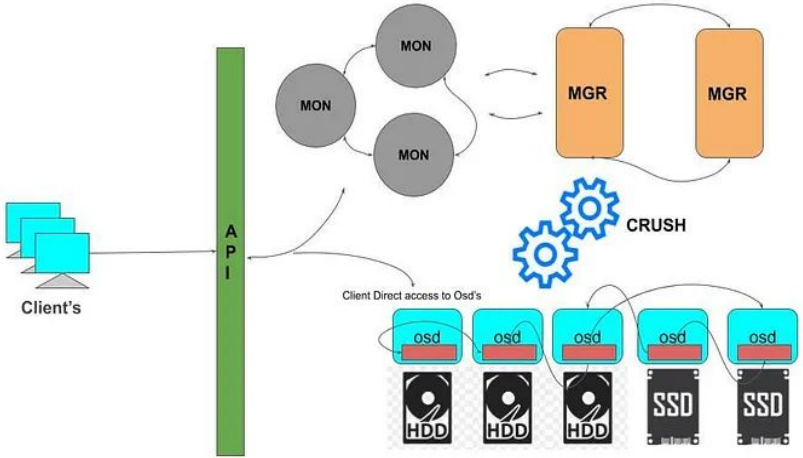
As Ceph is a software defined storage system, it can tailored for different needs and deployed in different ways.

The basic components are:

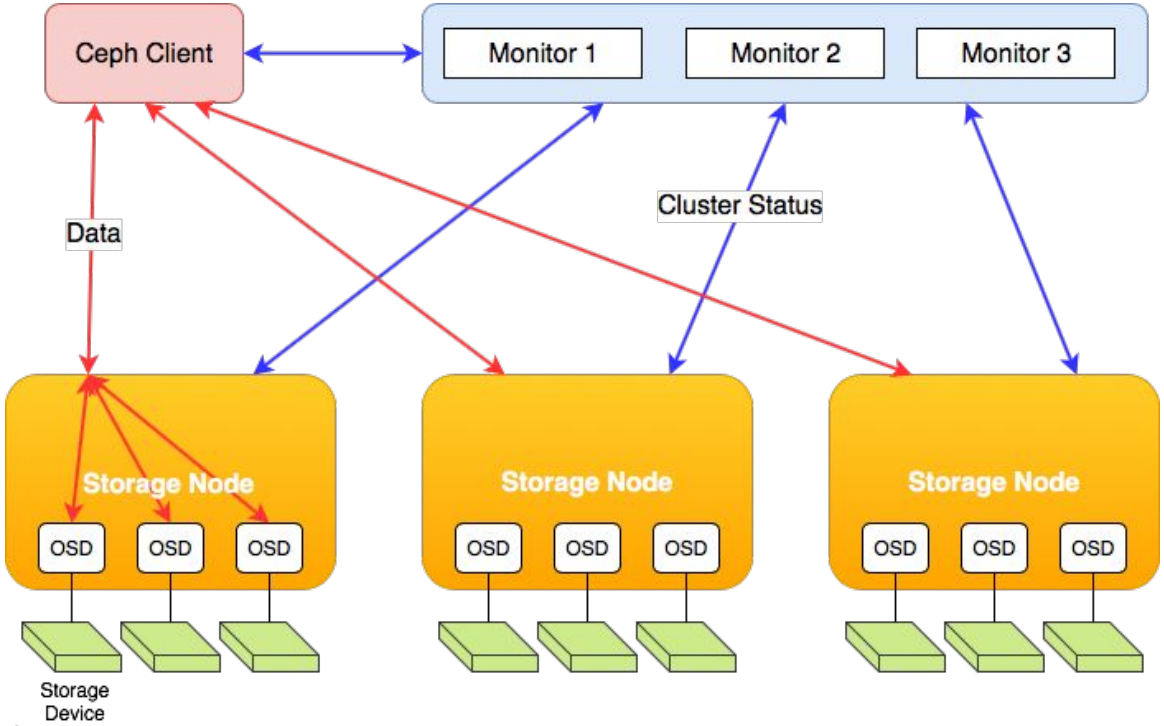
- Servers with disk or or servers with disk chassis in JBOD-mode
  No RAID-controllers needed or wanted, just present the physical drives directly to Ceph
  One physical disk <-> one OSD service
  - Physical disks can be spinning disks (HDD), SSD or NVMe
  - Can use faster media (say SSD or NVMe for HDD pools) for journals and OSD database
- One fast Ethernet network for intra cluster operations
- One fast Ethernet network for client access dependent on needs and optimization.

Hardware for a Ceph cluster

- Servers to run MON- MGR- and MDS services
  - MDS only needed if CephFS is used
- Servers to run RGW services
- Servers to run NFS services (depends on existing CephFS)
- Can be run on stand-alone hardware for each server type
- or all type of services can be combined as desired
- all dependant on performance needs, cost considerations, hardware specs etc.



Hardware for a Ceph cluster

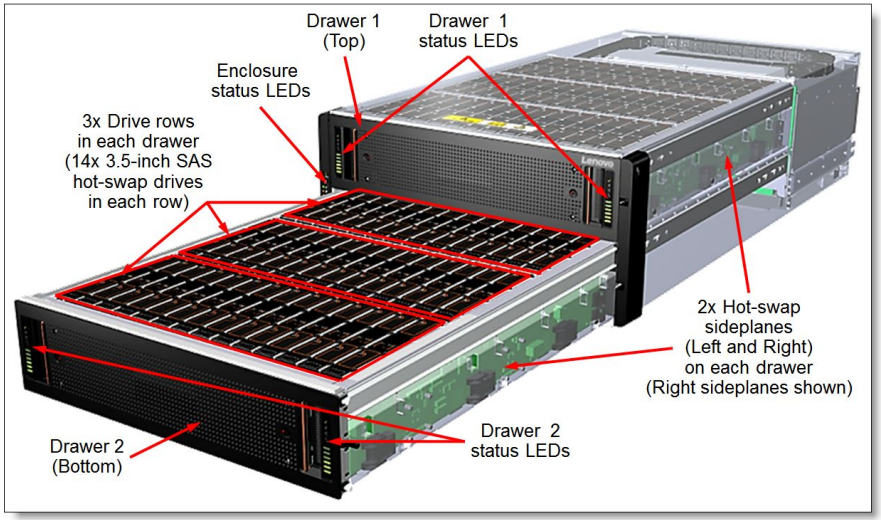


Cephyr

- 13 OSD:servers
  - 7 with 12x 10TB HDD + 2x 1.92 TB NVMe
    2 x 10 core Intel Xeon 4114, 96 GB memory
  - 6 with 24x 10TB HDD + 2x 1.92 TB NVMe (extension)
    2 x 16 core Intel Xeon 4216, 384 GB memory
  - 2x 25Gbit Ethernet for cluster and client networks, respectively
  - In total 2 PB HDD
- 3 servers for MON, MGR and MDS combined
  - 1 x 6 core Intel Xeon 3104, 48 GB memory
  - 2x 25Gbit Ethernet for client network
- 2 servers for RGW (and NFS testing)
  - 2 x 6 core Intel Xeon 3106, 96 GB memory
  - 2x 25Gbit Ethernet for client and *external* networks, respectively
- 2 x 25Gbit-switch, each with dual 100Gbit uplinks

Mimer-Ceph

- 14 Servers
  - 2 x 32 core Intel Xeon 6338, 3x 800GB NVMe
  - 10 servers with 256 GB memory
  - 4 servers with 384 GB memory (extra for MDS:s and one MGR running Prometheus and Grafana)
  - MGR, MON, MDS and RGW spread out but together with OSD
  - 2x 25Gbit Ethernet for client network
  - 2x 100Gbit Ethernet for cluster network
- 7 Storage chassis (JBOD)
  - 84x 14 TB HDD
  - In total 8 PB HDD
  - Each JBOD serves 2 servers
- 2x 100Gbit-switch, each with dual 100Gbit uplinks



CHALMERS UNIVERSITY OF TECHNOLOGY | E-COMMONS

Working with Ceph

- To monitor our Ceph systems, the Dashboards are a good start
  - *Cephyr*: http://10.38.20.1:8443/
  - *Mimer-Ceph:* https://10.38.20.101:8443/
- *Mimer-Ceph* was deployed using, and *Cephyr* migrated to **cephadm**
  - cephadm is a tool to deploy, and manage Ceph cluster servers and services
  - cephadm *complements* standard Ceph commands, it is not a replacement
  - using cephadm all Ceph services run as containers (Podman for us)
  - to work with our Ceph clusters, you log in to one of our Ceph (mgmt) servers and enter the cephadm environment (container) using `cephadm shell`
- Inside the the shell, use ceph commands
- `ceph orch` is the cephadm part inside the shell!

Working with Ceph

- **ceph status** (or **ceph -s** for short)

```
[root@mimer-osd06 ~]# cephadm shell
Inferring fsid 5406fed0-d52b-11ec-beff-7ed30a54847b
Inferring config /etc/ceph/ceph.conf
Using ceph image with id '0f5473a1e726' and tag '<none>' created on 2025-05-07 17:48:39 +0000 UTC
quay.io/ceph/ceph@sha256:1b9158ce28975f95def6a0ad459fa19f1336506074267a4b47c1bd914a00fec0
[ceph: root@mimer-osd06 /]# ceph status
  cluster:
    id:     5406fed0-d52b-11ec-beff-7ed30a54847b
    health: HEALTH_OK

  services:
    mon: 5 daemons, quorum mimer-osd01,mimer-osd02,mimer-osd03,mimer-osd04,mimer-osd05 (age 30h)
    mgr: mimer-osd01.pkndxs(active, since 10d), standbys: mimer-osd02.jzjame
    mds: 2/2 daemons up, 1 standby
    osd: 588 osds: 588 up (since 9d), 588 in (since 3w)
    rgw: 12 daemons active (6 hosts, 1 zones)

  data:
    volumes: 1/1 healthy
    pools:   23 pools, 5137 pgs
    objects: 597.25M objects, 1.5 PiB
    usage:   1.8 PiB used, 5.5 PiB / 7.3 PiB avail
    pgs:     4635 active+clean
             310  active+clean+scrubbing
             192  active+clean+scrubbing+deep

  io:
    client:   11 MiB/s rd, 10 MiB/s wr, 89 op/s rd, 503 op/s wr
```

- **ceph -w** (watch) to follow logs

Working with Ceph

- **ceph health detail** - to show more info if problems
- **ceph df** - get pool info, usage etc.

```
--- RAW STORAGE ---
CLASS     SIZE    AVAIL     USED  RAW USED  %RAW USED
hdd    7.3 PiB  5.5 PiB  1.8 PiB   1.8 PiB      24.46
TOTAL  7.3 PiB  5.5 PiB  1.8 PiB   1.8 PiB      24.46

--- POOLS ---
POOL                            ID   PGS   STORED  OBJECTS     USED  %USED  MAX AVAIL
.mgr                             1     1  1.1 GiB      316  1.8 GiB      0    1.6 PiB
cephfs_metadata                  6    32   19 GiB   18.43M   58 GiB      0    1.6 PiB
ec-isa_pool_CephFS               7  2048  734 TiB  311.68M  826 TiB  14.04    4.1 PiB
ec-isa_pool_CephRGW             12    32      0 B        0      0 B      0    4.1 PiB
.rgw.root                       14    32  160 KiB        4  192 KiB      0    4.1 PiB
default.rgw.log                 15    32   10 KiB      231  677 KiB      0    1.6 PiB
default.rgw.control             16    32      0 B        8      0 B      0    1.6 PiB
default.rgw.meta                17     8   12 KiB       18  195 KiB      0    1.6 PiB
default.rgw.buckets.index       18     8  7.9 GiB      889   24 GiB      0    1.6 PiB
default.rgw.buckets.data        19    32  154 KiB        2  480 KiB      0    1.6 PiB
project_backup.rgw.data         20  2048  661 TiB  217.53M  754 TiB  12.98    4.1 PiB
swesrc                          24    32  3.9 MiB        9  7.1 MiB      0    1.6 PiB
swesrc_data                     25   256   24 TiB    1.54M   27 TiB   0.54    4.1 PiB
dcache                          26    32  5.8 GiB    1.51M   17 GiB      0    1.6 PiB
dcache_data                     27   256  177 TiB   46.01M  195 TiB   3.72    4.1 PiB
c3se_rbd                        28    32   17 KiB        6   86 KiB      0    1.6 PiB
c3se_rbd_data                   29    32   40 KiB        1   48 KiB      0    4.1 PiB
cirrus-cec-1.cinder-backup      35    32   11 KiB        4   53 KiB      0    1.6 PiB
cirrus-cec-1.cinder-backup.data 36    32   40 KiB        1   48 KiB      0    4.1 PiB
cirrus-cec-1.cinder-volumes     37    32  1.5 TiB  385.25k  4.3 TiB   0.09    1.6 PiB
cirrus-cec-1.ephemeral-vms      39    32   11 KiB        3   44 KiB      0    1.6 PiB
cirrus-cec-1.glance-images      41    32  216 KiB      139  1.7 MiB      0    1.6 PiB
cirrus-cec-1.glance-images.data 42    32  1.2 TiB  160.54k  812 GiB   0.02    4.1 PiB
```

Working with Ceph

- **ceph osd df** - get individual osd info, usage etc.

```
[ceph: root@mimer-osd06 /]# ceph osd df|head -10 ; ceph osd df |tail -10
ID   CLASS  WEIGHT    REWEIGHT  SIZE     RAW USE  DATA     OMAP      META     AVAIL    %USE   VAR   PGS  STATUS
  0    hdd  12.75069   1.00000   13 TiB  3.1 TiB  3.1 TiB    21 KiB   14 GiB  9.6 TiB  24.63  1.01  103      up
  1    hdd  12.75069   1.00000   13 TiB  3.1 TiB  3.0 TiB    12 KiB   14 GiB  9.7 TiB  24.13  0.99   97      up
  2    hdd  12.75069   1.00000   13 TiB  3.1 TiB  3.0 TiB    22 KiB   13 GiB  9.7 TiB  24.22  0.99  105      up
  3    hdd  12.75069   1.00000   13 TiB  3.3 TiB  3.3 TiB    20 KiB   14 GiB  9.5 TiB  26.02  1.06  107      up
  4    hdd  12.75069   1.00000   13 TiB  3.1 TiB  3.0 TiB    16 KiB   12 GiB  9.7 TiB  23.89  0.98   97      up
  5    hdd  12.75069   1.00000   13 TiB  3.5 TiB  3.4 TiB    23 KiB   15 GiB  9.3 TiB  27.19  1.11  109      up
  6    hdd  12.75069   1.00000   13 TiB  3.1 TiB  3.0 TiB   599 MiB   12 GiB  9.7 TiB  24.08  0.98   95      up
  7    hdd  12.75069   1.00000   13 TiB  3.5 TiB  3.4 TiB    20 KiB   15 GiB  9.3 TiB  27.29  1.12  106      up
  8    hdd  12.75069   1.00000   13 TiB  3.4 TiB  3.4 TiB    19 KiB   16 GiB  9.3 TiB  26.92  1.10  109      up
580    hdd  12.75069   1.00000   13 TiB  3.1 TiB  3.0 TiB    32 KiB   13 GiB  9.7 TiB  24.24  0.99   96      up
581    hdd  12.75069   1.00000   13 TiB  2.9 TiB  2.8 TiB    16 KiB   12 GiB  9.9 TiB  22.70  0.93   91      up
582    hdd  12.75069   1.00000   13 TiB  3.4 TiB  3.3 TiB    22 KiB   14 GiB  9.4 TiB  26.60  1.09  106      up
583    hdd  12.75069   1.00000   13 TiB  3.0 TiB  3.0 TiB   598 MiB   12 GiB  9.7 TiB  23.83  0.97   98      up
584    hdd  12.75069   1.00000   13 TiB  3.1 TiB  3.0 TiB    25 KiB   15 GiB  9.7 TiB  24.16  0.99   94      up
585    hdd  12.75069   1.00000   13 TiB  3.3 TiB  3.3 TiB    20 KiB   13 GiB  9.5 TiB  25.93  1.06  102      up
586    hdd  12.75069   1.00000   13 TiB  3.1 TiB  3.0 TiB    25 KiB   12 GiB  9.7 TiB  23.93  0.98  100      up
587    hdd  12.75069   1.00000   13 TiB  3.2 TiB  3.1 TiB    21 KiB   15 GiB  9.6 TiB  24.84  1.02   97      up
                        TOTAL  7.3 PiB  1.8 PiB  1.8 PiB    80 GiB  7.5 TiB  5.5 PiB  24.46
MIN/MAX VAR: 0.88/1.13  STDDEV: 1.19
```

Working with Ceph

- **ceph fs status**

```
cephfs - 395 clients
======
RANK  STATE              MDS                 ACTIVITY     DNS    INOS   DIRS   CAPS
 0    active  cephFS.mimer-osd06.fbtklw  Reqs:  533 /s  10.1M  9766k  1119k  1063k
 1    active  cephFS.mimer-osd07.iclael  Reqs:  127 /s  10.1M  9773k  1138k   892k
       POOL           TYPE     USED  AVAIL
 cephfs_metadata    metadata  57.4G  1685T
ec-isa_pool_CephFS    data     826T  4214T
       STANDBY MDS
cephFS.mimer-osd08.kfdvcq
MDS version: ceph version 18.2.7 (6b0e988052ec84cf2d4a54ff9bbbc5e720b621ad) reef (stable)
```

- **ceph orch ps**

```
[ceph: root@mimer-osd01 /]# ceph orch ps --daemon_type mds
NAME                          HOST         PORTS  STATUS         REFRESHED  AGE  MEM USE  MEM LIM  VERSION  IMAGE ID      CONTAINER ID
mds.cephFS.mimer-osd06.fbtklw  mimer-osd06         running (9d)     7m ago   3y    40.5G        -  18.2.7   0f5473a1e726  30e144292689
mds.cephFS.mimer-osd07.iclael  mimer-osd07         running (9d)     7m ago   3y    41.4G        -  18.2.7   0f5473a1e726  4750f2b96c90
mds.cephFS.mimer-osd08.kfdvcq  mimer-osd08         running (9d)     7m ago   3y    26.1M        -  18.2.7   0f5473a1e726  906ddca63b35
```

Working with Ceph

- **ceph orch daemon restart mds.cephFS.mimer-osd08.kfdvcq** - restarting a redundant server is safe and sometimes needed
- **ceph orch host ok-to-stop mimer-osd12** - check if a host is safe to restart
- **ceph orch host maintenance enter mimer-osd12** - put a host in maintenance mode, ready to restart etc. (--force needed for some hosts!)
- **ceph orch host maintenance exit mimer-osd12** - put a host back in production (usually we see errors on the first try, second try works…)
- **ceph balancer off** - turn off auto balancer before (longer) stops of a server to prevent unneeded data movement
- **ceph balancer on** - turn it back on
- **ceph balancer status** - check if it is enabled, and if balancing is needed
- **podman ps** - check currently running containers on any Ceph cluster host
- **ceph tell mds.0 client ls | grep *123456789*** - find the IP of a CephFS client

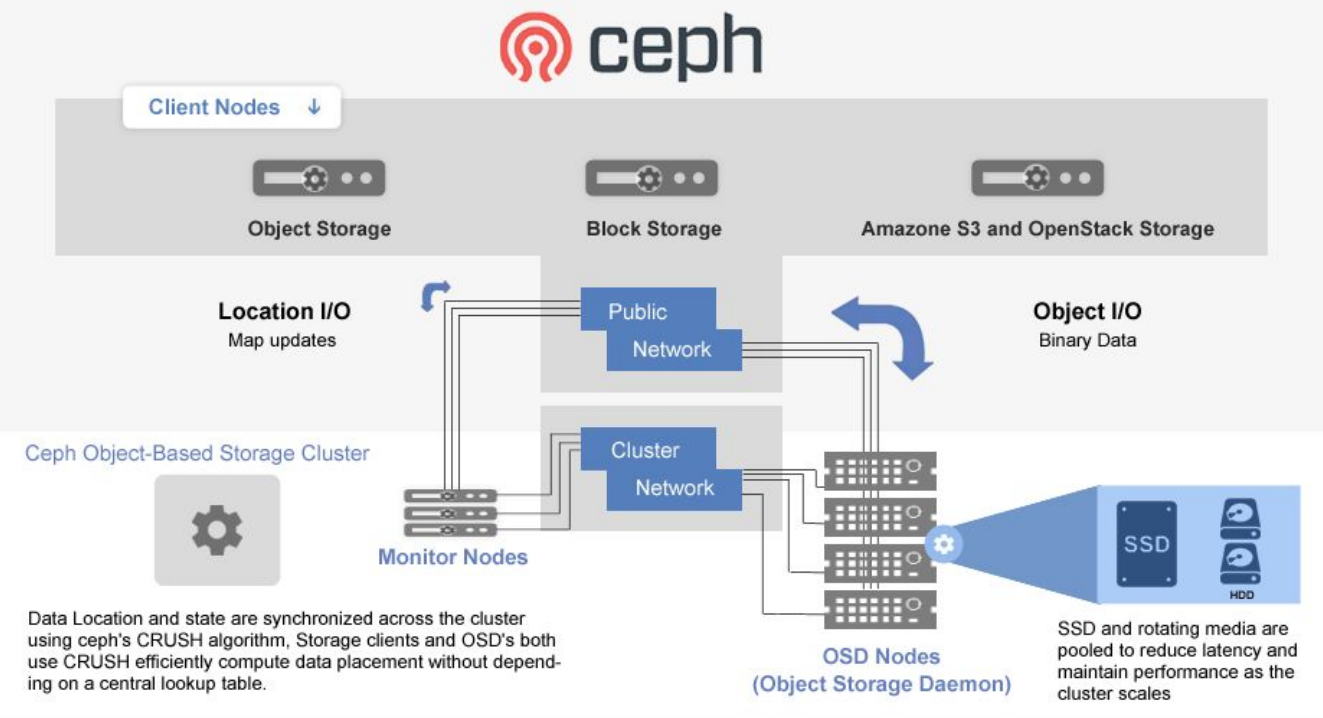
Disk replacement

- Disable osd_spec so that ceph orch doesn't attempt to automatically add a new OSD when it sees the new drive ID. Can't use osd_spec because it doesn't work with flash partitions
- Remove the old OSD
  - mgr: **ceph orch osd rm *12345* --replace**
  - mgr: **ceph orch osd rm status**
- Replace OSD reusing the DB volume on flash LVM partition
  - osd-host: **ceph-volume lvm prepare --bluestore --no-systemd --osd-id *12345* --data /dev/mapper/mpath*xx* --block.db /dev/ceph-*xxx*/osd-db-*yyy***
  - mgr: **ceph orch daemon add osd mimer-osd*zz*:/dev/mapper/mpath*xx***
  - mgr: **ceph orch daemon start osd.*12345***
- Flash drive replacement can be done with osd_spec
  - Zap all affected OSDs
  - **ceph orch apply -i osd_spec_osd*XX*.yml --dry-run**

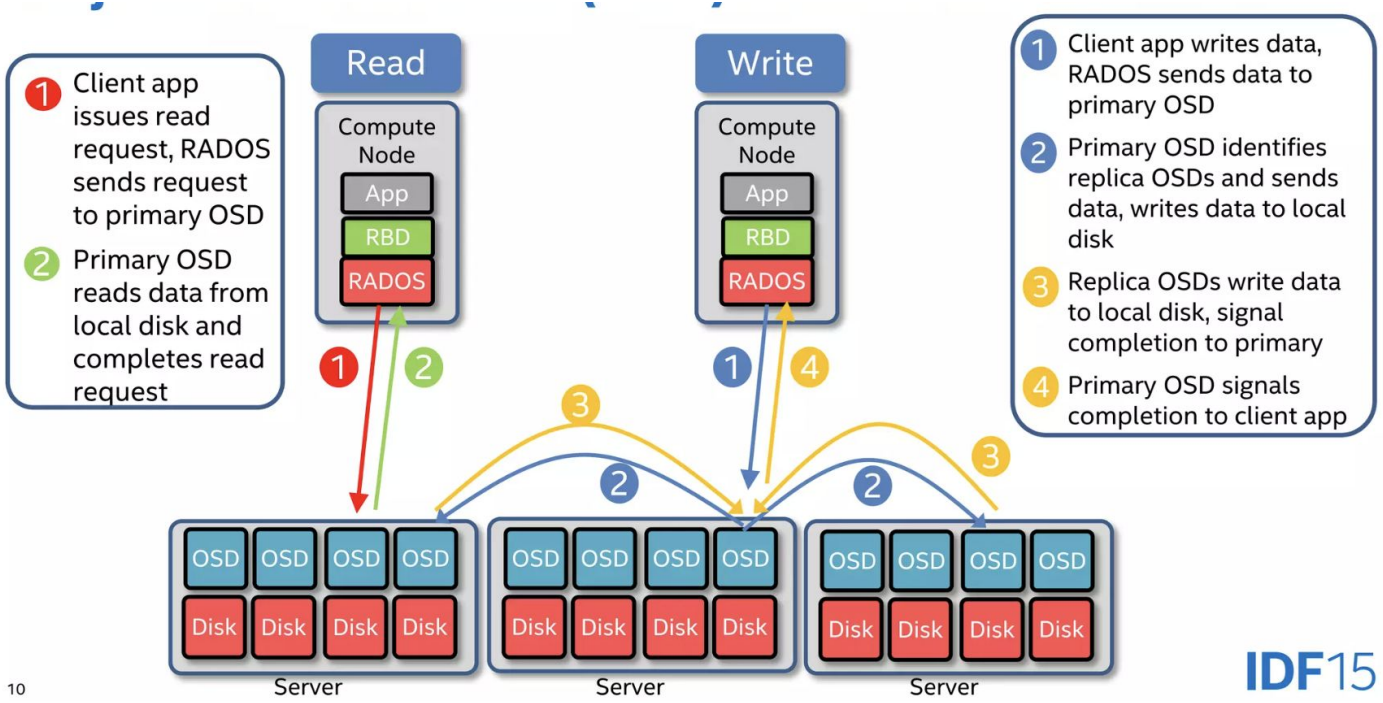
https://url.c3se.chalmers.se/system/Mimer/Ceph-Disk-Replacement/

Extra slides

Ceph Components



Ceph Components



IDF15

Example osd_spec.yml

```
---
placement:
  host_pattern: "mimer-osdXX"
service_id: osd_spec
service_type: osd
spec:
  data_devices:
    rotational: 1
  db_devices:
    rotational: 0
```


THE
END