

# OpenStack Ceph 연동

가상화 WG

1. Ceph란?
2. Ceph 구조
3. Ceph 구성요소
4. Ceph Data Placement
5. Ceph 배포
6. OpenStack Cinder Backend로 Ceph 연동

# 1. Ceph란?

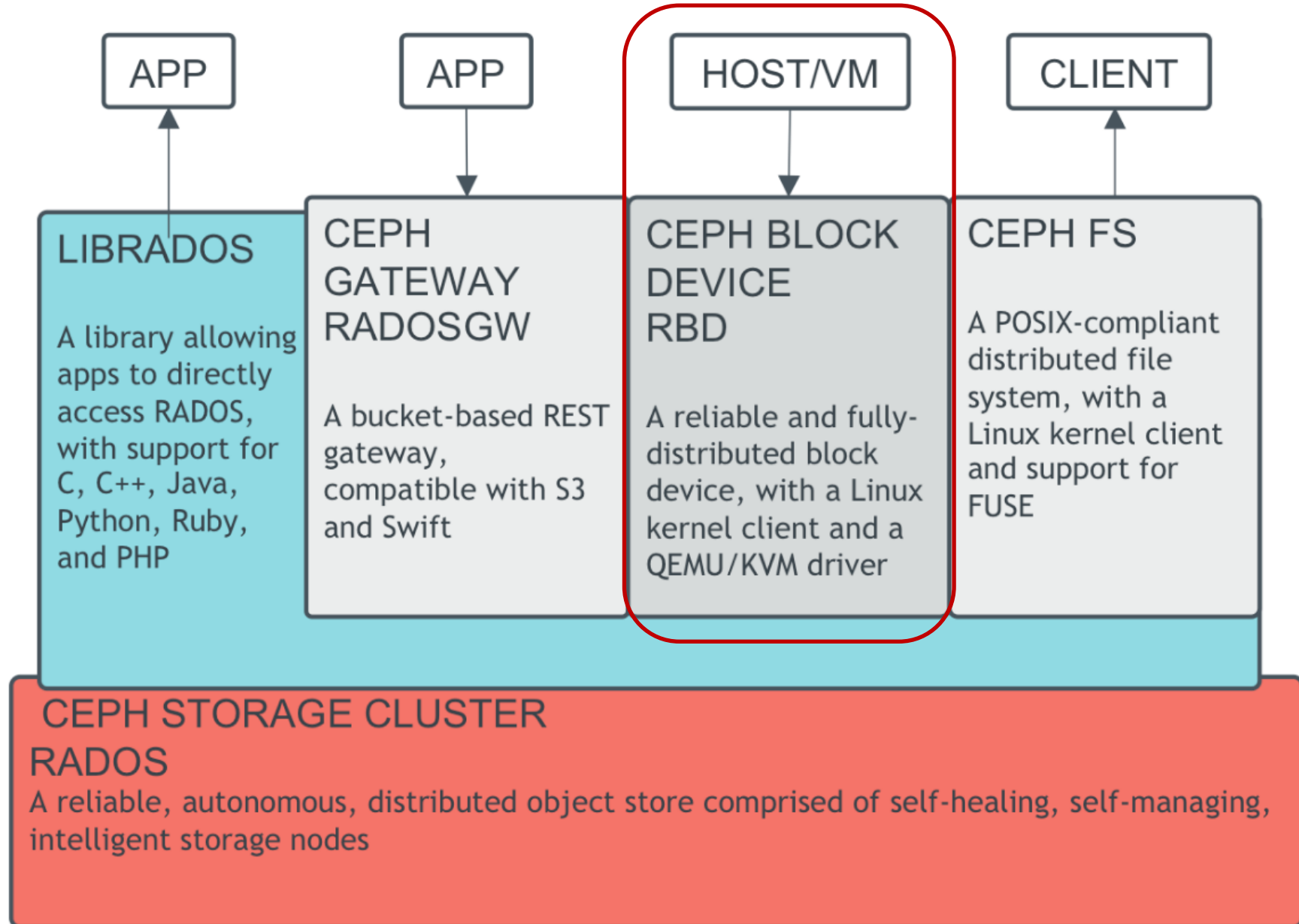
---

- 2003년 California 대학에서 시작한 Storage 프로젝트
- 대규모 클러스터 컴퓨팅 컴퓨팅 환경에서 병렬 분산 파일 시스템을 Target
- 오픈소스 이며 다양한 방식으로 클러스터 간의 통신 지원
- Unified Storage : 스토리지 시스템을 이루고 있는 모든 파트의 다양한 모듈 통합
- Object 스토리지로 확장성이 뛰어나고 size에 제약이 없음
- Object 스토리지 데몬(OSD), 모니터 데몬(MON)으로 구성, 자신 스스로 관리



# 1. Ceph란?

- Unified Storage : 스토리지 시스템을 이루고 있는 모든 파트의 다양한 모듈 통합

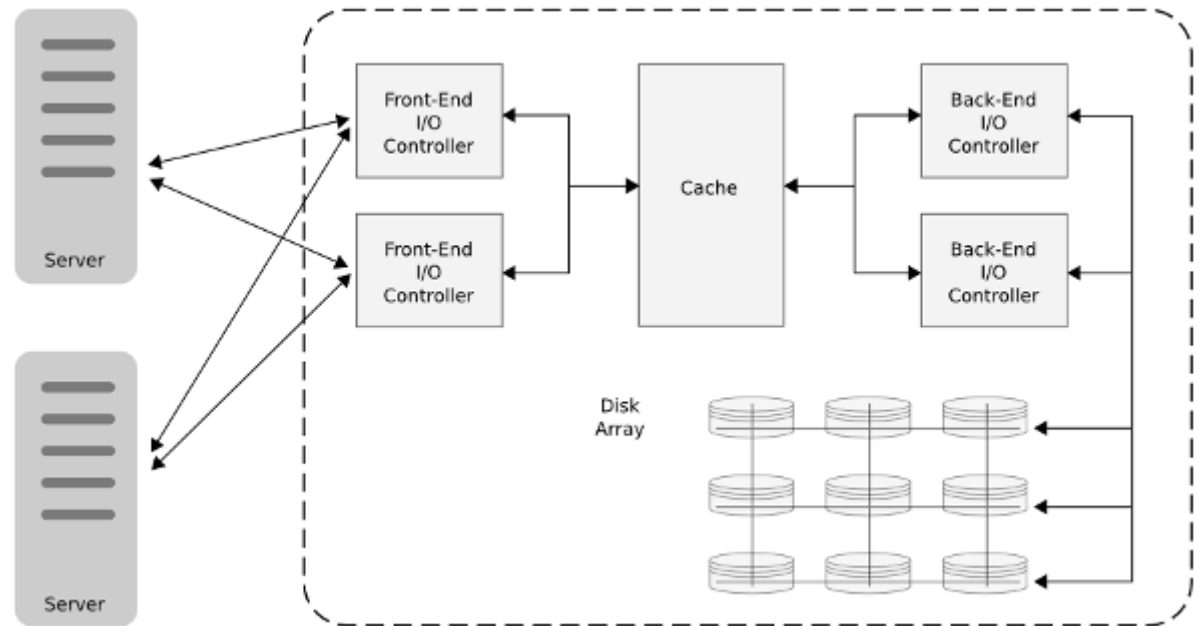


# 일반 스토리지 구조

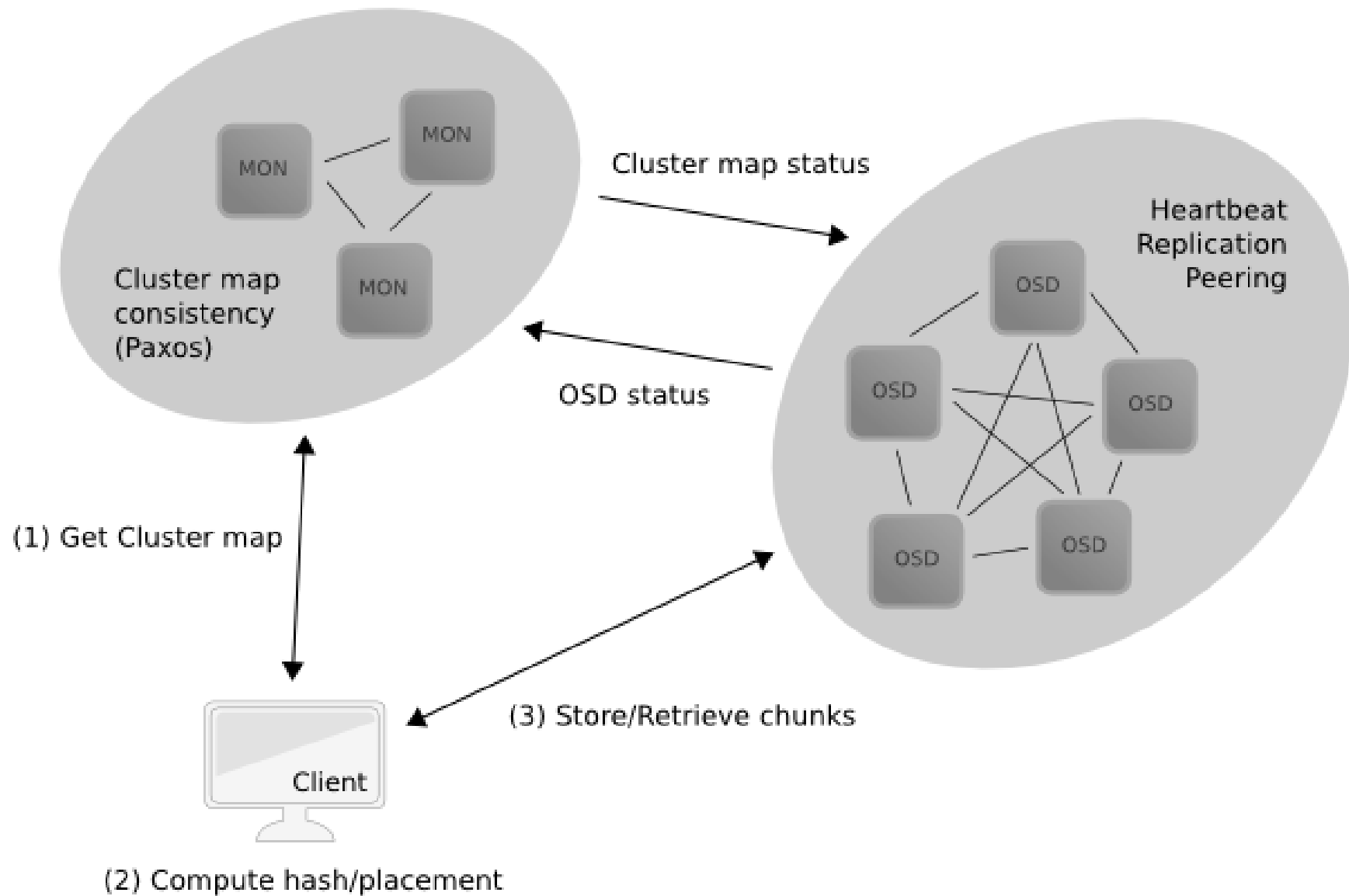
- RAID 기반의 디스크 이중화, 삼중화
- Master 서버가 클러스터의 metadata를 관리
- Nodes에서 실제 데이터를 저장, 이 데이터는 게이트웨이나 front-end 서버를 통해서 접근 가능
- front-end 서버는 Nodes에 데이터가 어디에 있는지 알기 위해 Master에 접근

## • 문제점

- Master 서버, front-end 서버의 이중화 필요
- nodes 데이터 I/O 높은 성능을 위해 충분한 메모리가 필요하고 적절한 데이터 분산 정책이 필요

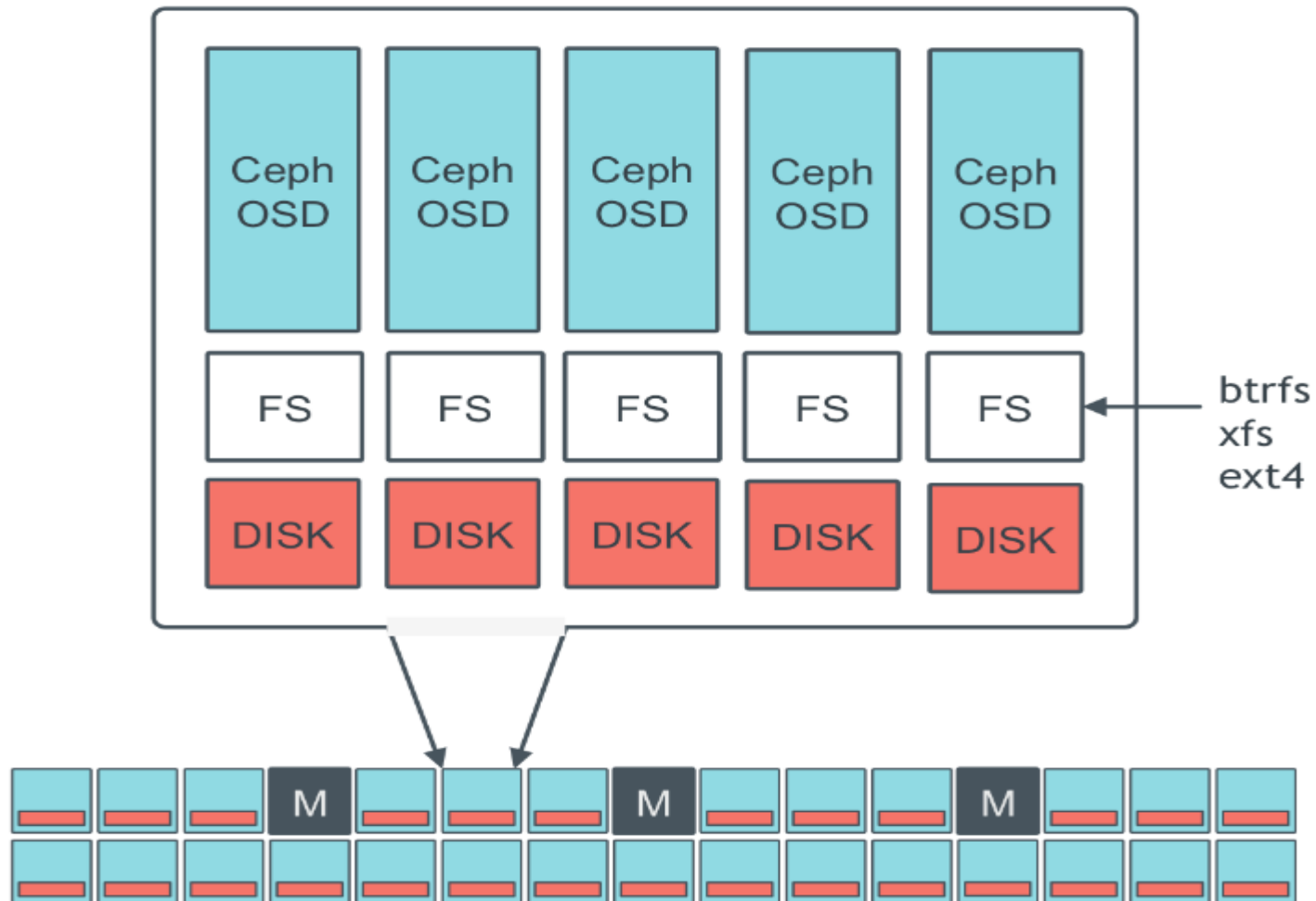


# 확장성과 고가용성을 고려한 스토리지 모델

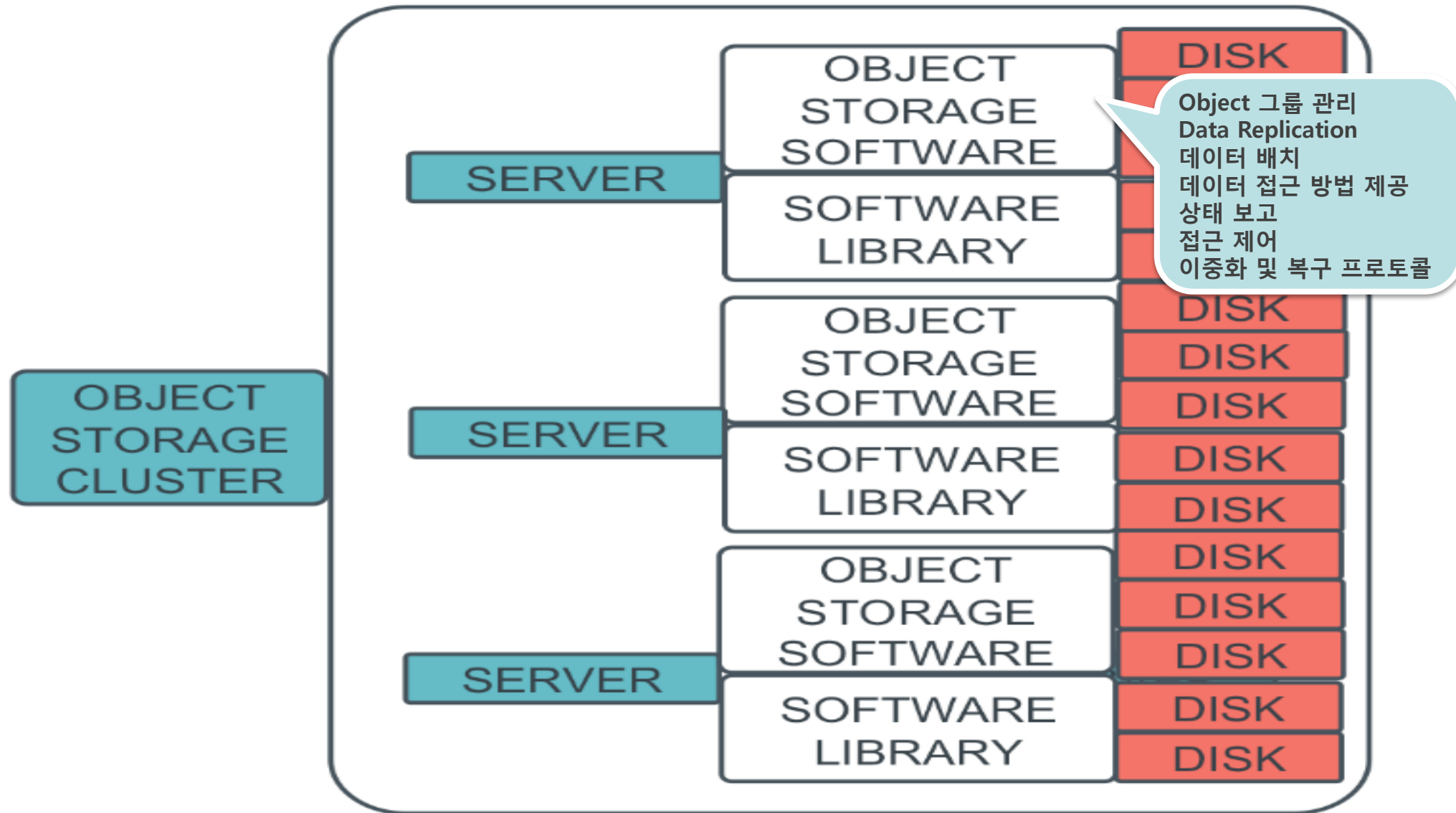


## 2. Ceph 구조

- RADOS(Reliable Autonomic Distributed Object Store)기반 클러스터 스토리지
- 모니터 데몬 : 서버에 설치되는 데몬으로 클러스터 내 홀수 개수로 3개 이상 필요
- OSD(Object Storage Daemon) : 모든 스토리지 노드에 설치되는 데이터 관리 데몬



## 2. Ceph 구조

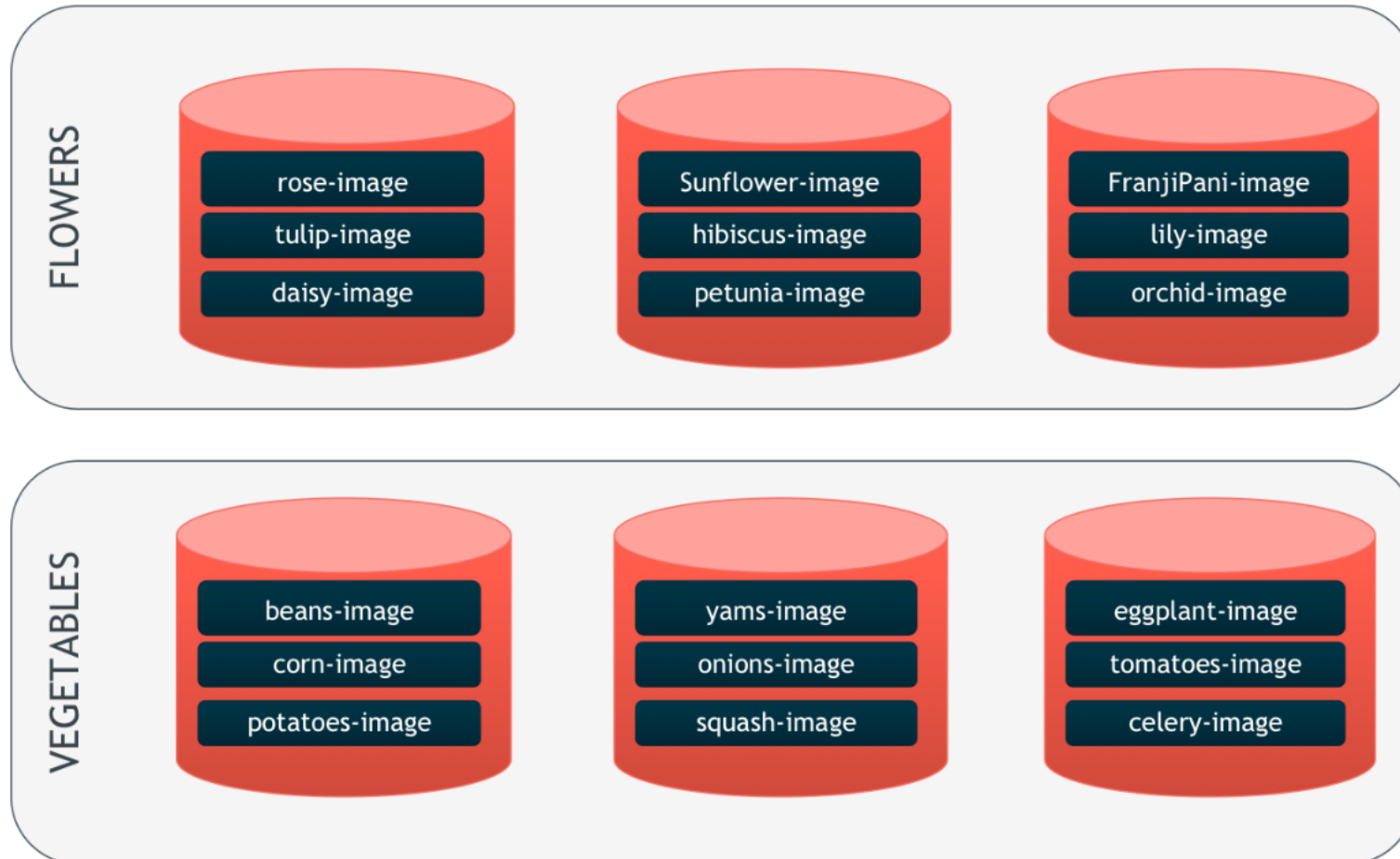




## 2. Ceph 구조

### • Logical Groups

- 특정 Category 기반에 논리적인 그룹으로 objects를 묶음
- Swift에서는 Container, Amazon에서는 bucket이라고 함
- 그룹 단위로 리플리케이션 설정과 배치 방식을 결정하여 성능 향상을 기대



### 3. Ceph 구성요소

---

- **OSD**

- primary 이거나 secondary 노드
- primary node에서는 Data replication, coherency, balancing, healing, recovery 담당
- secondary node는 primary node에 의해 관리되나 자신 스스로 primary node가 될 수 있음
- 지원하는 파일 type : XFS, BTRFS, EXT4

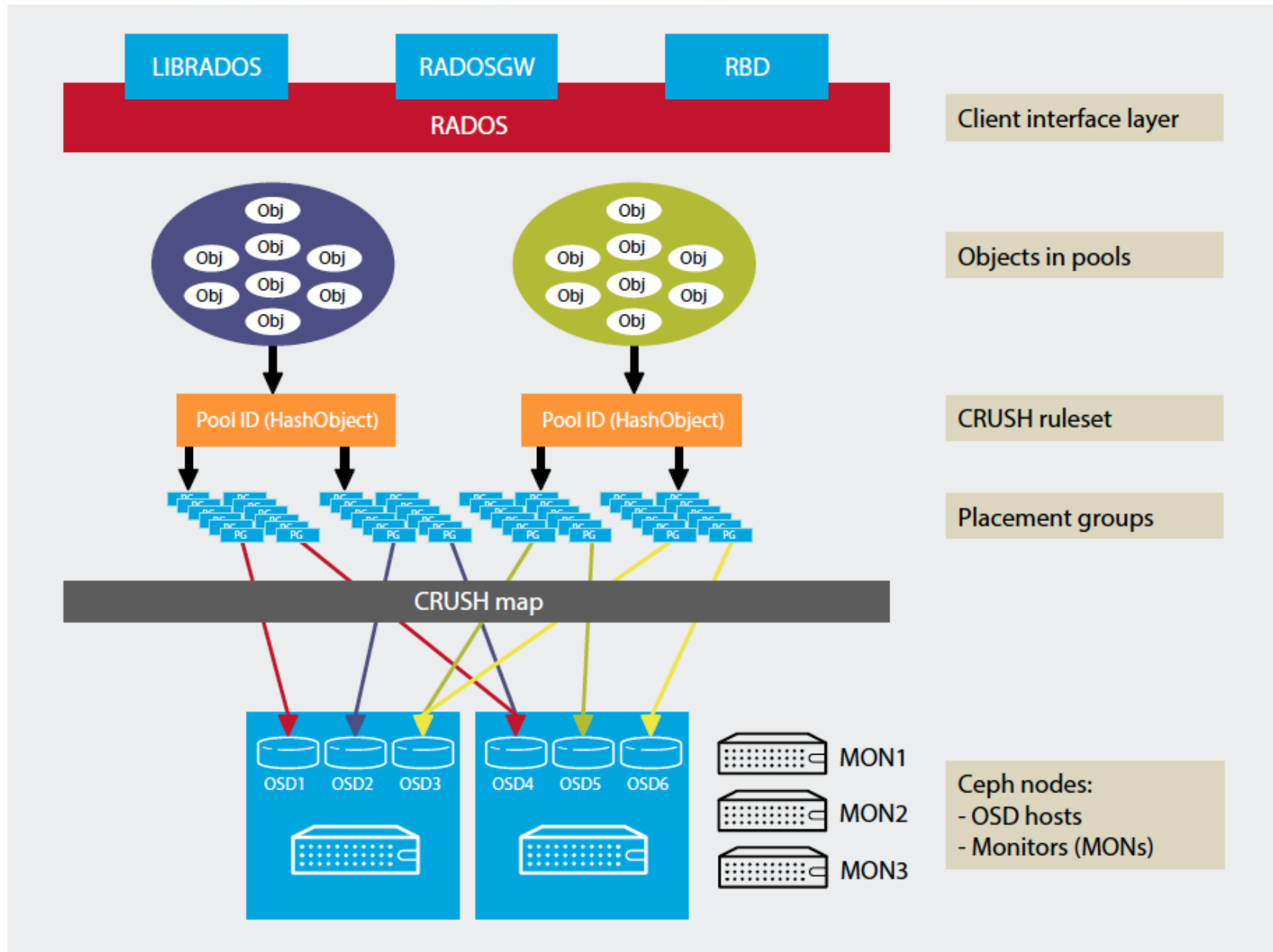
- **클러스터 맵(Cluster Map)**

- Monitor Map, OSD Map, Placement group Map 등을 모두 포함(Map : 상태정보)
- 클러스터의 상태와 총 스토리지 용량, 사용량 등을 관리

- **Monitor**

- 클러스터 맵, CRUSH 맵의 Master 본을 관리
- 데이터를 Read/Write를 하기 위해 Object가 있는 위치 확인, 그 후에는 OSD랑 직접 연동

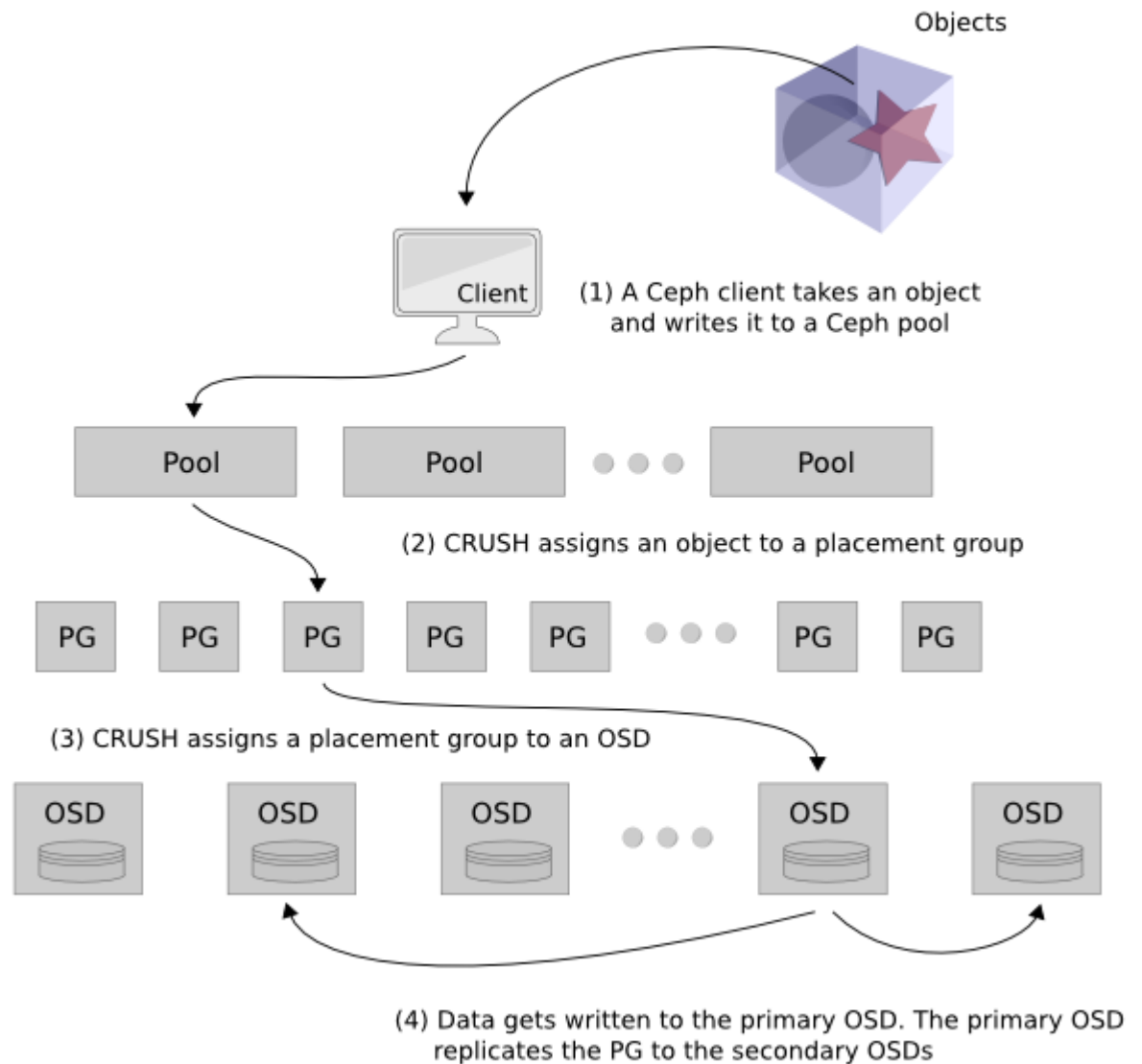
### 3. Ceph 구성요소



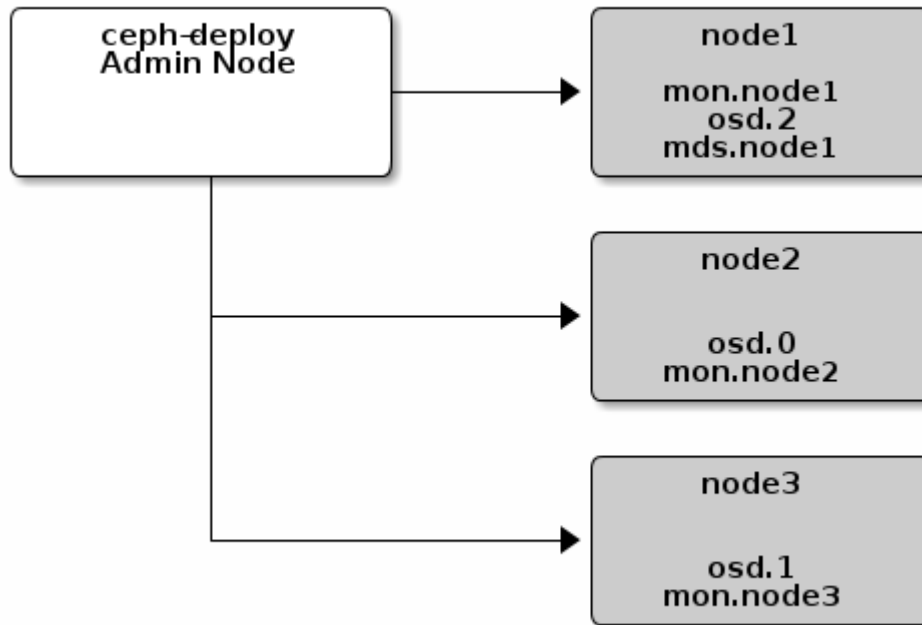
## 4. Ceph Data Placement

### • Objects에서 OSD까지

- (1) Object이름에 hash 함수 적용
- (2)  $\text{Pool ID} + \text{hash}(\text{Object}) = \text{PG}$
- (3) PG 정보를 가지고 CRUSH 알고리즘 적용하여 primary OSD 선택
- (4) 필요하다면 이후 OSD도 추가 선택



## 5. Ceph 배포



### (1) Ceph deploy 패키지 설치

```
[root@ceph1 ~]# yum localinstall ceph-deploy-1.5.37-0.noarch.rpm
Loaded plugins: fastestmirror, langpacks
Examining ceph-deploy-1.5.37-0.noarch.rpm: ceph-deploy-1.5.37-0.noarch
Marking ceph-deploy-1.5.37-0.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package ceph-deploy.noarch 0:1.5.37-0 will be installed
--> Finished Dependency Resolution
```

## 5. Ceph 배포

### (2) Cluster 생성

```
[root@ceph1 ~]# ceph-deploy new ceph1 ceph2 ceph3
[ceph_deploy.conf][DEBUG ] found configuration file at: /root/.cephdeploy.conf
[ceph_deploy.cli][INFO   ] Invoked (1.5.37): /usr/bin/ceph-deploy new ceph1 ceph2 ceph3
[ceph_deploy.cli][INFO   ] ceph-deploy options:
[ceph_deploy.cli][INFO   ] username                : None
[ceph_deploy.cli][INFO   ] func                    : <function new at 0x20479b0>
[ceph_deploy.cli][INFO   ] verbose                 : False
[ceph_deploy.cli][INFO   ] overwrite_conf          : False
[ceph_deploy.cli][INFO   ] quiet                   : False
[ceph_deploy.cli][INFO   ] cd_conf                 : <ceph_deploy.conf.CephDeployConf instance at 0x1e75dd0>
[ceph_deploy.cli][INFO   ] cluster                 : ceph
[ceph_deploy.cli][INFO   ] ssh_copykey             : True
[ceph_deploy.cli][INFO   ] mon                     : ['ceph1', 'ceph2', 'ceph3']
[ceph_deploy.cli][INFO   ] public_network          : None
[ceph_deploy.cli][INFO   ] ceph_conf               : None
[ceph_deploy.cli][INFO   ] cluster_network         : None
[ceph_deploy.cli][INFO   ] default_release         : False
[ceph_deploy.cli][INFO   ] fsid                    : None
[ceph_deploy.new][DEBUG  ] Creating new cluster named ceph
[ceph_deploy.new][INFO   ] making sure passwordless SSH succeeds
[ceph1][DEBUG  ] connected to host: ceph1
[ceph1][DEBUG  ] detect platform information from remote host
[ceph1][DEBUG  ] detect machine type
[ceph1][DEBUG  ] find the location of an executable
```

## 5. Ceph 배포

### (3) Monitor 배포

```
ceph@ceph1 ~]$ ceph-deploy mon create ceph1 ceph2 ceph3
ceph_deploy.conf[DEBUG] found configuration file at: /home/ceph/.cephdeploy.conf
ceph_deploy.cli[INFO] Invoked (1.5.37): /bin/ceph-deploy mon create ceph1 ceph2
ceph3
ceph_deploy.cli[INFO] ceph-deploy options:
ceph_deploy.cli[INFO]   username                : None
ceph_deploy.cli[INFO]   verbose                  : False
ceph_deploy.cli[INFO]   overwrite_conf          : False
ceph_deploy.cli[INFO]   subcommand               : create
ceph_deploy.cli[INFO]   quiet                    : False
ceph_deploy.cli[INFO]   cd_conf                  : <ceph_deploy.conf.cephdep
oy.Conf instance at 0x2b5db90>
ceph_deploy.cli[INFO]   cluster                  : ceph
ceph_deploy.cli[INFO]   mon                      : ['ceph1', 'ceph2', 'ceph3
']
ceph_deploy.cli[INFO]   func                     : <function mon at 0x2ac87d
>
ceph_deploy.cli[INFO]   ceph conf                 : None
```

```
[ceph3][DEBUG] "outside_quorum": [],
[ceph3][DEBUG] "quorum": [],
[ceph3][DEBUG] "rank": 2,
[ceph3][DEBUG] "state": "electing",
[ceph3][DEBUG] "sync_provider": []
[ceph3][DEBUG] }
[ceph3][DEBUG] *****
*****
[ceph3][INFO] monitor: mon.ceph3 is running
[ceph3][INFO] Running command: sudo ceph --cluster=ceph --admin-daemon /var/run/ce
ph/ceph-mon.ceph3.asok mon_status
```



## 5. Ceph 배포

### (4) bootstrap(일련의 설치) key 생성

```
ceph@ceph1 ~]$ ceph-deploy gatherkeys ceph1
ceph_deploy.conf[DEBUG] found configuration file at: /home/ceph/.cephdeploy.conf
ceph_deploy.cli[INFO] Invoked (1.5.37): /bin/ceph-deploy gatherkeys ceph1
ceph_deploy.cli[INFO] ceph-deploy options:
ceph_deploy.cli[INFO] username                : None
ceph_deploy.cli[INFO] verbose                : False
ceph_deploy.cli[INFO] overwrite_conf         : False
ceph_deploy.cli[INFO] quiet                  : False
ceph_deploy.cli[INFO] cd_conf                : <ceph_deploy.conf.cephdep
loyp.Conf instance at 0x1af7fc8>
ceph_deploy.cli[INFO] cluster                : ceph
ceph_deploy.cli[INFO] mon                    : ['ceph1']
ceph_deploy.cli[INFO] func                   : <function gatherkeys at 0
x1a5e1b8>
```

### (5) 각 노드의 Disk /dev/sdb를 활용해서 OSD 설치

```
ceph@ceph1 ~]$ ceph-deploy osd create ceph1:sdb ceph2:sdb ceph3:sdb
ceph_deploy.conf[DEBUG] found configuration file at: /home/ceph/.cephdeploy.conf
ceph_deploy.cli[INFO] Invoked (1.5.37): /bin/ceph-deploy osd create ceph1:sdb ce
ph2:sdb ceph3:sdb
ceph_deploy.cli[INFO] ceph-deploy options:
ceph_deploy.cli[INFO] username                : None
ceph_deploy.cli[INFO] block_db                : None
ceph_deploy.cli[INFO] disk                    : [('ceph1', '/dev/sdb', No
ne), ('ceph2', '/dev/sdb', None), ('ceph3', '/dev/sdb', None)]
```



## 5. Ceph 배포

### (6) Ceph 상태 확인 : `sudo ceph -s`

```
[ceph@ceph1 ~]$ sudo ceph -s
cluster 0eb6fed4-3a33-4f96-9d29-18e6913f8c68
health HEALTH_OK ①
monmap e1: 3 mons at {ceph1=10.10.10.243:6789/0,ceph2=10.10.10.244:6789/0,ceph3=10.10.10.245:6789/0} ②
election epoch 10, quorum 0,1,2 ceph1,ceph2,ceph3 ③
osdmap e14 ④ 3 osds: 2 up, 2 in
flags sortbitwise,require_jewel_osds
pgmap v24: 64 pgs, 1 pools, 0 bytes data, 0 objects
68528 kB used, 197 GB / 197 GB avail
64 active+clean
[ceph@ceph1 ~]$
```

1: 클러스터의 상태

2: epoch는 각 Map의 버전

3: Monitor 노드는 3개

4: OSD 노드 3개

## 5. Ceph 배포

### (8) Cinder가 사용할 Pool 생성

```
[ceph@ceph1 ~]$ ceph osd pool create cephvols 128
pool 'cephvols' created
[ceph@ceph1 ~]$ sudo rados df
```

pool name	rd	rd KB	KB	objects	clones	degraded	unfound
cephvols	0	0	0	0	0	0	0
rbd	0	0	0	0	0	0	0
	0	0	4	10240			
total used		72704		0			
total avail		207441888					
total space		207514592					

### (9) Cinder가 cephvols에 접근할 수 있는 권한 부여

```
[ceph@ceph1 ~]$ sudo ceph auth get-or-create client.cinder mon 'allow r' osd 'allow
class-read object_prefix rbd_children, allow rwx pool=cephvols' -o /etc/ceph/ceph.cl
ient.cinder.keyring
```

## [실습] 6. OpenStack Cinder Backend로 Ceph 연동

(1) Cinder, nova-compute가 설치된 노드에서 ceph 연동 패키지 설치

```
Controller, compute# yum -y install ceph-common
```

(2) (controller만) Cinder keyring , ceph.conf 설정 파일 다운로드 및 권한 설정

```
controller# cd /etc/ceph
```

```
controller# wget http://10.10.10.30/data/ceph.client.cinder.keyring
```

```
controller# wget http://10.10.10.30/data/ceph.conf
```

```
controller# chown cinder.cinder ceph.client.cinder.keyring ceph.conf
```

```
controller# chmod 0640 ceph.client.cinder.keyring ceph.conf
```

## [실습] 6. OpenStack Cinder Backend로 Ceph 연동

### (3) **Controller**의 /etc/cinder/cinder.conf 에 backend로 ceph연동 정보 입력

```
...
enabled_backends = lvm, glusterfs, rbd
...
[lvm]
volume_backend_name=lvm
volume_driver=cinder.volume.drivers.lvm.LVMVolumeDriver
iscsi_ip_address=10.10.10.51
iscsi_helper=lioadm
volumes_dir=/var/lib/cinder/volumes
...
[glusterfs]
volume_backend_name=glusterfs
...
[rbd]
volume_driver = cinder.volume.drivers.rbd.RBDDriver
volume_backend_name=rbd
rbd_pool = cephvols
rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_flatten_volume_from_snapshot = False
rbd_max_clone_depth = 5
rbd_store_chunk_size = 4
rados_connect_timeout = -1
rbd_user = cinder
rbd_secret_uuid = b63b856c-2fb7-42cd-b4a7-aec0f5e996b7
```

wget class.example.com/data/cinder.conf.rbd  
다운로드 해서 복사 붙여넣기 활용

## [실습] 6. OpenStack Cinder Backend로 Ceph 연동

### (4) Nova Compute가 설치된 노드에서 libvirt가 ceph 연동하기 위한 인증정보

```
compute# cd ~
```

```
compute# wget http://10.10.10.30/data/client.cinder.key
```

```
compute# wget http://10.10.10.30/data/cinder-ceph.xml
```

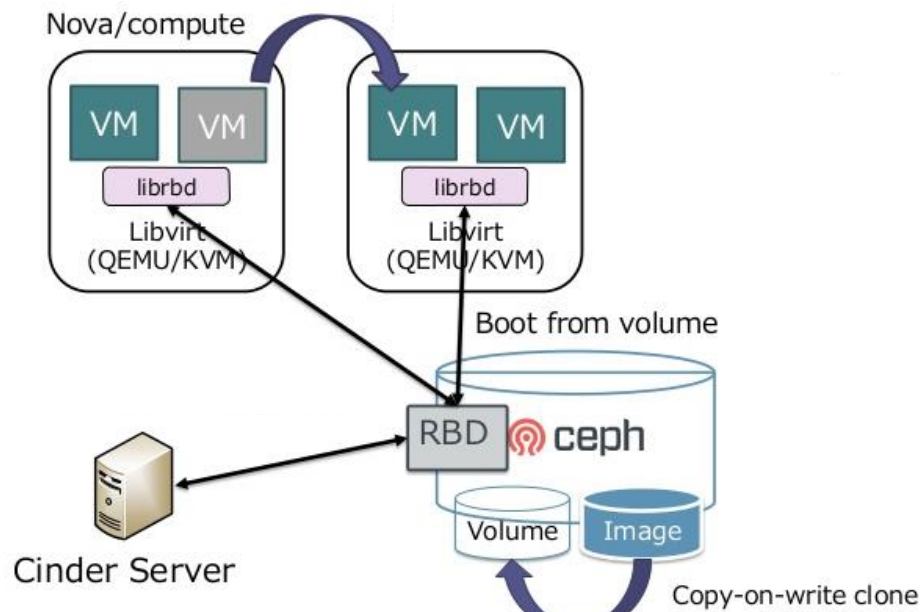
```
compute# wget http://10.10.10.30/data/myuuid.txt
```

```
<secret ephemeral='no' private='no'>
  <uuid>b63b856c-2fb7-42cd-b4a7-aec0f5e996b7</uuid>
  <usage type='ceph'>
    <name>client.cinder secret</name>
  </usage>
</secret>
```

### (5) Key를 virsh 명령어로 libvirt에 전달

```
compute# virsh secret-define --file cinder-ceph.xml
```

```
compute# virsh secret-set-value --secret ~~~~uuid입력~~~~ --base64 $(cat client.cinder.key)
```



## [실습] 6. OpenStack Cinder Backend로 Ceph 연동

### (6) Ceph 관련 유저, UUID 정보를 nova (compute node) 에 설정 및 재기동

```
compute# vi /etc/nova/nova.conf
```

```
.....
```

```
rbd_user = cinder
```

```
rbd_secret_uuid = 97c76c77-1376-432c-ac8b-e6a35f3318a9
```

```
....
```

```
compute# systemctl restart openstack-nova-compute
```

```
controller1# systemctl restart openstack-cinder-api.service
```

```
controller1# systemctl restart openstack-cinder-volume.service
```

controller의 cinder.conf 에 있는 정보와 일치토록

## [실습] 6. OpenStack Cinder Backend로 Ceph 연동

(7) Ceph backend에 연결되는 type 생성(ceph) – GUI에서도 가능

```
controller1# source keystonerc_admin
```

```
controller1# cinder list
```

```
controller1# cinder type-list
```

```
controller1# cinder type-create rbd
```

```
controller1# cinder type-key rbd set volume_backend_name=rbd
```

```
controller1# cinder type-list
```

```
controller1# cinder create --display-name cephvol1 --volume-type rbd 1
```

강사가 ceph에 실제 데이터가 보이는지 확인

```
# rbd -p cephvols ls
```