



OpenSourceOsijek

[Open Source Osijek](http://OpenSourceOsijek.hr)

**Licenca : GPL**

**Autor: Hrvoje Horvat**

# CEPH sustav

## Sadržaj

CEPH storage.....	4
NAS i SAN .....	4
Što je uopće NAS sustav ? .....	4
A što je SAN sustav .....	8
Zbog čega uopće koristiti NAS ili SAN sustave.....	9
U čemu je problem sa standardnim NAS ili SAN sustavima .....	11
Slijedeći korak: Klasterski i/ili redundantni NAS sustavi .....	11
Redundantni ili Klasterizirani NAS Sustavi .....	12
Redundantni ili Klasterizirani SAN Sustavi .....	14
ZFS - negdje između .....	14
Proces učenja .....	17
Što je slijedeće .....	18
Object storage .....	19
CEPH .....	19
Prednosti CEPH-a .....	21
Kako se podaci distribuiraju unutar cijelog CEPH clustera.....	22
Kako se zapisuju podaci na CEPH cluster .....	25
CEPH Block Device (Rados Block Device) tj. RBD .....	29
Licenca .....	34
CEPH Storage with integration in Proxmox VE.....	35
Introduction.....	35
Advantages.....	35
What we need ? .....	36
What we should be aware of using Ceph RBD (as Block Device) inside Proxmox VE .....	37
CEPH Logical scheme .....	37
Planning .....	40
Considerations.....	40
Journal .....	41
Sizing .....	41
OSD Server nodes .....	42
Basic Calculation .....	42
IOPS Calculations .....	42
CEPH Recovery time .....	43
Latency.....	44
CEPH installation inside Proxmox VE (on Proxmox VE nodes) .....	45
Initial steps .....	45
Prerequisites .....	45
CEPH Cluster creation .....	46
Monitor Servers .....	48
Proxmox VE and CEPH Auth.....	49
Checking status .....	50
Preparing DISK drives for CEPH .....	51
First step.....	51
CLI Commands .....	54
CEPH Pools .....	55

Mounting CEPH RBD to Proxmox VE .....	56
Adding a new nodes/disks/OSDs to CEPH Cluster.....	59
Optimizing CEPH .....	61
Configuration part (what we will change) .....	62
Monitoring and configuring CEPH.....	63
2.....	<b>Error! Bookmark not defined.</b>
3.....	<b>Error! Bookmark not defined.</b>
Latency considerations.....	68
Disk performance calculation .....	68
Page Tools.....	<b>Error! Bookmark not defined.</b>

# CEPH storage

Prije nego pređemo na distribuirani, redundantni klusterski CEPH storage (sustav za mrežnu pohranu podataka) prvo moramo razumjeti standardne sustave za pohranu i dijeljenje podataka te sustave koji osiguravaju zalihost (redundanciju). Potom ćemo preći na CEPH. Pokušati ću na što brži i trivijalniji način objasniti sve ove osnovne pojmove i tehnologije.

## NAS i SAN

### Što je uopće NAS sustav ?

NAS (Engl. Network Attached Storage) odnosno “mrežno spojena spremišta podataka” osiguravaju nam prostor za spremanje podataka, preko mreže. Ovo su zapravo mrežni dijeljeni sustavi za spremanje podataka, koji rade na razini datoteka (i naravno direktorija) koje pohranjujemo na njih i to preko mrežnih protokola za dijeljenje datoteka.

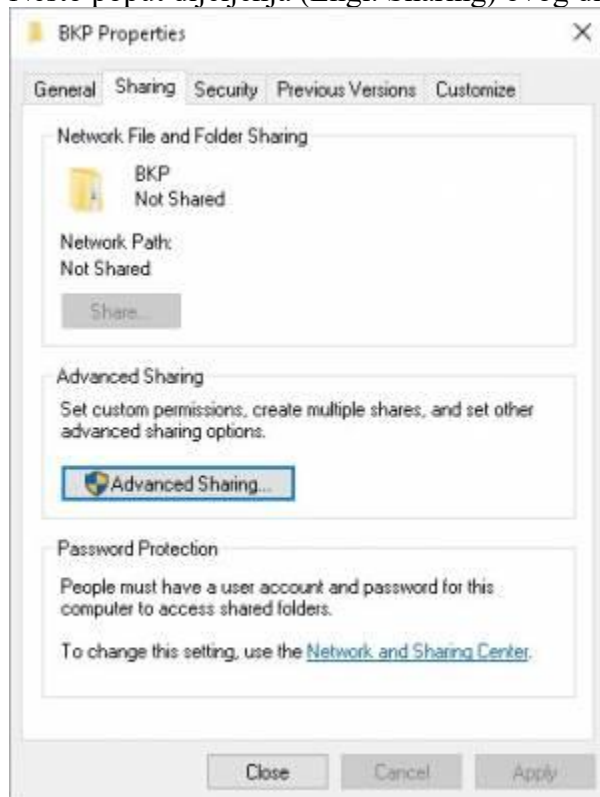
Svako dijeljenje datoteka preko mreže (Engl. Network Share), korištenjem nekog od mrežnih protokola koji postoje za tu namjenu, možemo nazvati upotrebom kao NAS sustava.

Dijeljeni pristup datotekama preko mreže omogućavaju nam sljedeći mrežni protokoli. Navesti ćemo one najčešće u upotrebi:

- **NFS** ([Network Files System](#)) - koristi se uglavnom na Linux/Unix operacijskim sustavima (ili ponekad u Windows okruženju). Open source varijanta podrazumjeva korištenje nekog od **nfs** daemona (servisa)
- **SMB/CIFS** ([Server Message Block / Common Internet File System](#)) - koristi se uglavnom na Windows ili Linux okruženjima. Koristi se osim za dijeljenje datoteka i za dijeljenje pisača, i drugih uređaja te dodatnih funkcionalnosti, preko mreže.
  - Open source rješenje se zove [samba](#)
  - **Windows Share** je integriran u sve Windows operacijske sustave, s ograničenjem od maksimalno 10 paralelnih (otvorenih) konekcija na Windows dijeljeni direktorij ako se radi o verziji Windowsa koja NIJE : Windows Server: 2003/2003 R2/2008/2008 R2/2012/2012(R2)
- **AFP** ([Apple Filing Protocol](#)) - koristi se za dijeljenje datoteka na Mac OS računalima.
- Istoj kategoriji pripadaju i **FTP** ([File Transfer Protocol](#)) i **TFTP** ([Trivial File Transfer Protocol](#)) protokoli, s time da su oni jednostavniji i nemaju naprednije mogućnosti kao gore navedeni.
- Često se koristi i **WebDAV** ([Web Distributed Authoring and Versioning](#)) koji je što se tiče funkcionalnosti negdje između FTP i gore navedenih protokola
- ...

Najosnovniji primjer upotrebe sustava za koji bi mogli reći da je neka vrsta NAS sustava bi bio klasično dijeljenje nekog direktorija preko mreže, iz Windows OS-a.

Nešto poput dijeljenja (Engl. Sharing) ovog direktorija (D:\BKP) na slici :



U slučaju upotrebe na Windows operacijskom sustavu, sve se za jednostavno mrežno dijeljenje svodi na odabir željenog direktorija te njegovog dijeljenja preko SMB/CIFS sustava.

Ako govorimo o “samostalnom” NAS sustavu odnosno uređaju pod nekom od varijanti Unix ili Linux operacijskih sustava, ova procedura se u konačnici uglavnom svodi na nekoliko koraka :

1. Kreiranje nekog RAID polja diskova, koje ćemo dalje koristiti kao jedan “logički” disk
2. Particioniranje “logičkog” diska koji ćemo koristiti za dijeljenje datoteka
3. Formatiranje kreiranih particija (Linux ext3/4 ,Linux XFS, Windows NFS ili sl. (ovisno o operacijskom sustavu NAS uređaja i našim potrebama) )
4. Mountanje formatirane particije u neki direktorij
5. Odabir mountanog direktorija te konfiguracija i aktivacija nekog od mrežnih protokola za dijeljeni pristup preko mreže - pogledajte dolje (NFS, SMB/CIFS, AFP ili sl.)

## Osnovne pretpostavke i planiranje za NAS ili SAN sustav

1. Kod odabira NAS ili SAN sustava odnosno poslužitelja, prvo moramo biti svjesni zahtjeva za softverom (operacijskim sustavom). Operacijski sustav za NAS ili SAN može biti :

- Specijalizirani open source OS poput :
  - OpenFiler
  - OpenMediaVault
  - FreeNAS
  - Nas4Free

- NexentaStor (Nexenta Community Edition)
- ili neko od komercijalnih rješenja za NAS/SAN :
  - NexentaStor (Nexenta Enterprise Edition)
  - TrueNAS
  - Open-E
  - ...
- ili OS za opću upotrebu, koji ćemo dodatno konfigurirati prema našim potrebama:
  - Neki Linux koji želimo prilagoditi našim potrebama ili
  - Windows server koji već imamo te ga želimo optimizirati ili prenamjeniti za NAS/SAN sustav
  - ili nešto drugo

Postoje i gotovi “samostojeći” uređaji koji dolaze zajedno s operacijskim sustavom. Proizvode ih: **EMC2**, **IBM**, **Dell**, **NetApp** i drugi (oni nisu tema ove priče ).

2. Nakon što smo odabrali operacijski sustav za NAS ili SAN poslužitelj, moramo biti svjesni i njegovih zahtjeva za hardverom:

- Koji CPU će zadovoljiti naše potrebe
- Koliko RAM memorije (i koji tip)
- Koje mrežne kartice: 1Gbps ili 10Gbps, koji modeli (chipovi) i koliko ih je potrebno (jedna, dvije, tri, ...)
- Koji mrežni preklopnik (Switch) odabrati, s kojom verzijom Firmware-a (OS-a) i s kojim funkcionalnostima Pogledajte članak “[Switching i routing: jučer, danas, sutra](#)”
- Koji RAID kontroler, s koliko RAM memorije, BBU i sl. odabrati
- Koje tvrde diskove odabrati

3. Nakon planiranja resursa koji će nam trebati, potrebno je revidirati točku 2.

Pošto govorimo o NAS i SAN sustavima, ovdje ćemo se fokusirati na dio o RAID kontrolerima i diskovima, pošto ćemo bez njihovog dobrog odabira kasnije u radu doći do problema ili vrlo često gubitka podataka a koji ćemo vrlo skupo platiti ( \$\$\$ ).

### **Zbog čega RAID kontroler i pažnja pri odabiru diskova**

Krenimo od diskova.

Diskovi se u grubo dijele prema namjeni. U svakom slučaju želimo diskove koji su pouzdani ali i koji su proizvedeni za Tzv. Serversku namjenu u kombinaciji s RAID kontrolerima.

Budite svjesni činjenice da postoje i “serverski” diskovi koji nisu dizajnirani odnosno optimizirani za rad s RAID kontrolerima.

Za više detalja pogledajte knjigu “[Uvod u Linux i Linux napredno](#)” , poglavlje :“[Podjela prema namjeni diskova](#)”

### **RAID kontroleri**

I ovo je priča za sebe ali svakako želimo imati poštenu hardverski RAID kontroler provjerenoga proizvođača. Dodatno, važna je i verzija Firmware-a za RAID kontroler u kombinaciji s driverom za operacijski sustav na kojemu ga koristite.

U praksi su se čak i kombinacije određenih verzija Firmware-a i drivera kao i kombinacije Firmware RAID kontrolera i Firmware-a drugih komponenti poslužitelja (pr. matične ploče),

pokazale katastrofalnim odabirom. Dakle treba si dati malo truda i proučiti što se kupuje, kao i komentare korisnika, za što približniju konfiguraciju vašoj : OS, driveri (RAID, LAN, MB, ...), Firmare-i, Softver, ... Cjenovni opseg poštenih RAID kontrolera (ovisno o broju diskova koje možete spojiti na njega), kreće se od minimalno tisuću KN na više. Sve ispod toga, kao i odabir Tzv. “Integriranih RAID” kontrolera na matičnim pločama (osim ako su u pitanju prave Serverske matične ploče : 5.000+ KN) nemojte niti pomišljati koristiti. Za više detalja pogledajte poglavlje “[hardverski RAID](#)”

### **Integrirana RAID rješenja**

U ovim “[Integriranim](#)” kombinacijama, dobivate upravljački program (driver) i pripadajući softver, koji :

- je loše dokumentiran ili
- se jako rijetko održava ili
- je pun grešaka
- a u slučaju katastrofe nema definirane korake (što i kako) ili su oni nejasni (a uglavnom i ne rade)

Kod ovih rješenja, zapravo ne postoji pravi hardverski RAID kontroler već se većina ili gotovo sve funkcionalnosti RAID kontrolera odrađuju unutar upravljačkog programa (drivera) koji se pretvara da je RAID kontroler. On nadalje operacijskom sustavu prijavljuje RAID polje diskova o kojem se brine, kao jedan fizički disk - slično kako bi to uradio i pravi RAID kontroler.

I na kraju sve je zapravo prepušteno navedenom softveru (driveru) i operacijskom sustavu, pa što bude. Nakon nekog vremena obično bude : “Nešto ne radi. A kako da vratimo svoje podatke” ... e lijepo sam vam rekao. Na kraju priče s krivim odabirom tehnologije ili uređaja uvijek završite sa slanjem diskova u neku od tvrtki specijaliziranih za povrat podataka. Ova zabava će vas obično koštati znatno više nego da ste odmah kupili možda i najskuplje komponente koje postoje na (našem) tržištu. Povrat odnosno spašavanje podataka se često naplaćuje po GB, pa cifre vrlo brzo mogu narasti na desetke tisuća KN.

### **Opcija dva : Logical Volume Manager**

Zaboravimo na “Integrirane RAID” kontrolere. Druga prihvatljivija opcija je upotreba Tzv. “Logical Volume Managera” unutar operacijskog sustava (govorimo o linuxu). U linuxu se radi o Logical Volume Manageru verzije 2 (LVM2). Za više detalja o LVM2 pogledajte poglavlje [LVM2](#). I ovdje radi o softverskom RAID-u odnosno njegovoj funkcionalnosti. Ipak ovo je puno sigurnije i stabilnije rješenje od onoga koje dobivamo s upotrebom “Integriranih RAID kontrolera” na matičnim pločama za stolna računala (tj. Ne serverskim matičnim pločama). Ovo je dokazano rješenje koje koristi velika zajednica ljudi, a koje se razvija s otvorenim kodom pa se svaka novootkrivena greška vrlo brzo popravljiva. Osim toga vrlo je dobro dokumentirano te postoje dobro definirane procedure u slučaju neke havarije odnosno što i kako napraviti u kojem slučaju.

### **Zbog čega je ipak bolji hardverski RAID kontroler i pripadajuće RAID polje**

Zbog toga što će vam na budućem NAS ili SAN sustavu biti pohranjeni važni podaci. Stoga ne želite da se sve snima na samo jedan tvrdi disk, već na više njih i to u RAID polju koje vam osigurava najbolji odnos:

- sigurnosti podataka (koliko kopija podataka želite - na dva, tri ili više tvrdih diskova istovremeno)

- brzine rada
- brze zamjene neispravnog diska i povratka u normalan rad
- lakoće proširenja kapaciteta RAID polja

Dodatno želite RAID kontroler jer ne želite prepustiti nekom traljavom programčiću (mislim na “Integrirana RAID rješenja”) da vam odraduje ovako važne zadatke pohranjivanja vama osjetljivih i za život tvrtke važnih podataka. A s druge strane želite iskoristiti hardversku snagu pravog RAID kontrolera koji ne opterećuje sustav (CPU/RAM) pošto ima svoj specijalizirani CPU i pripadajuću RAM memoriju.

**Dakle želite ozbiljan zadatak oko RAID-a prepustiti tvrtci koja RAID kontrolere proizvodi profesionalno, kao i njihov pripadajući firmware/softver.**

Neka najbolji proizvođač specijaliziranog hardvera i softvera zaradi na onome što radi najbolje.

### Odabir RAID polja

Kao što je i odabir dobrog RAID kontrolera važan, važno je i pravilno odabrati RAID polje, ovisno o vašem budžetu i potrebama.

A sada ponovno bacite pogled na RAID polja u već navedenoj knjizi, u poglavlju : [Koja su najčešća RAID polja u upotrebi i koje su im prednosti i mane](#) “”

Skraćena verzija (navesti ćemo par često korištenih RAID polja) :

RAID polje	Brzina	Min. broj diskova	Koliko diskova se može istovremeno pokvariti	Vrijeme od zamjene pokvarenog diska do normalnog rada	Jednostavnost proširenja RAID polja
1	Jednako ili brže od jednog diska	2	1	Brzo	Vrlo teško
5	Brže od RAID 1	3	1	Sporo	Teško
6	Sporije od RAID 5	4	2	Sporo	Teško
10	Najbrže	4	2	Brzo	Jednostavnije

### I na kraju zbog čega priča o diskovima, RAID kontrolerima i RAID poljima ?

Kada smo kreirali neko RAID polje (na RAID kontroleru po mogućnosti) slijedeće je na redu upotreba tog RAID polja, preko nekog od mrežnih protokola za NAS (NFS, SMB/CIFS, AFP ili dr.) ili SAN, preko kojih dijelite podatke preko mreže (vrlo jednostavno 😊).

### A što je SAN sustav

SAN (Storage Area Network) sustavi s druge strane ne nude mrežno dijeljenje datoteka, već nam osiguravaju mrežni pristup Tzv. “Block-based” mediju. Ovo u praksi znači da SAN sustavi preko mreže praktično dijele svoje diskove ili polja diskova vidljiva klijentskoj (druvoj) strani kao “običan tvrdi disk”. Nadalje takav disk se sastoji od blokova podataka kao i bilo koji lokalni ATA,SATA,SAS ili neki drugi disk. Za više detalja o diskovima pogledajte knjigu “Uvod u Linux”, [poglavlje o diskovima](#). Za ovakvo mrežno dijeljenje praktično “sirove” površine diska, potrebi su mrežni protokoli koji nam osiguravaju ovakav pristup.



Neki od SAN protokola su :

- **Fibre Channel**
- **iSCSI**
- **ATA over Ethernet (AoE)**
- **HyperSCSI.**

Nakon što se klijent preko nekog od gore navedenih protokola spoji (na površinu diska), takav disk se mora prvo particionirati i formatirati kao da se radi o lokalnom disku odnosno disku spojenom na vaše lokalno računalo.

## **Zbog čega uopće koristiti NAS ili SAN sustave**

Zašto bi uopće koristili ovakve sustave ?

- Zbog potrebe za izradom sigurnosnih kopija vaših podataka (Engl. Backup), na centralni mrežni uređaj (obično NAS):
  - koji bi morao (i obično je) biti znatno sigurniji jer u pravilo snima podatke na više diskova istovremeno (obično koristi neko RAID polje)
  - i ima sve ostale komponente kvalitetnije i pouzdanije od “običnog” računala
- Zbog centraliziranog pohranjivanja i upotrebe. Dakle želimo centralizirano mjesto za pohranu :
  - Svih podataka (datoteka) s kojima radimo
  - Virtualnih računala, a što je preduvjet za bilo koji rad u klasteru u slučaju potrebe za redundancijom (otpornosti na kvar pojedinog poslužitelja ili njegovog održavanja odnosno nadogradnje ).
- Zbog pohranjivanja sigurnosnih kopija virtualnih računala i drugih podataka (pr. dokumenti, slike, ...) i to na dnevnoj, tjednoj, mjesečnoj, polugodišnjoj ili godišnjoj bazi.

I NAS i SAN sustavi osim čiste pohrane podataka imaju (i moraju imati) i cijeli niz drugih naprednih mogućnosti koje su vrlo važne za ovu namjenu odnosno upotrebu.

### **Koje opcije bi najminimalnije morali imati NAS/SAN sustavi ?**

Praćenje:

- performansi sustava
- RAID kontrolera i svih njegovih polja
- svakog pojedinog diska (performanse ali i što je još važnije grešaka u radu)
- servisa/daemoni
- mreže i mrežnih komponenti

Napredne opcije:

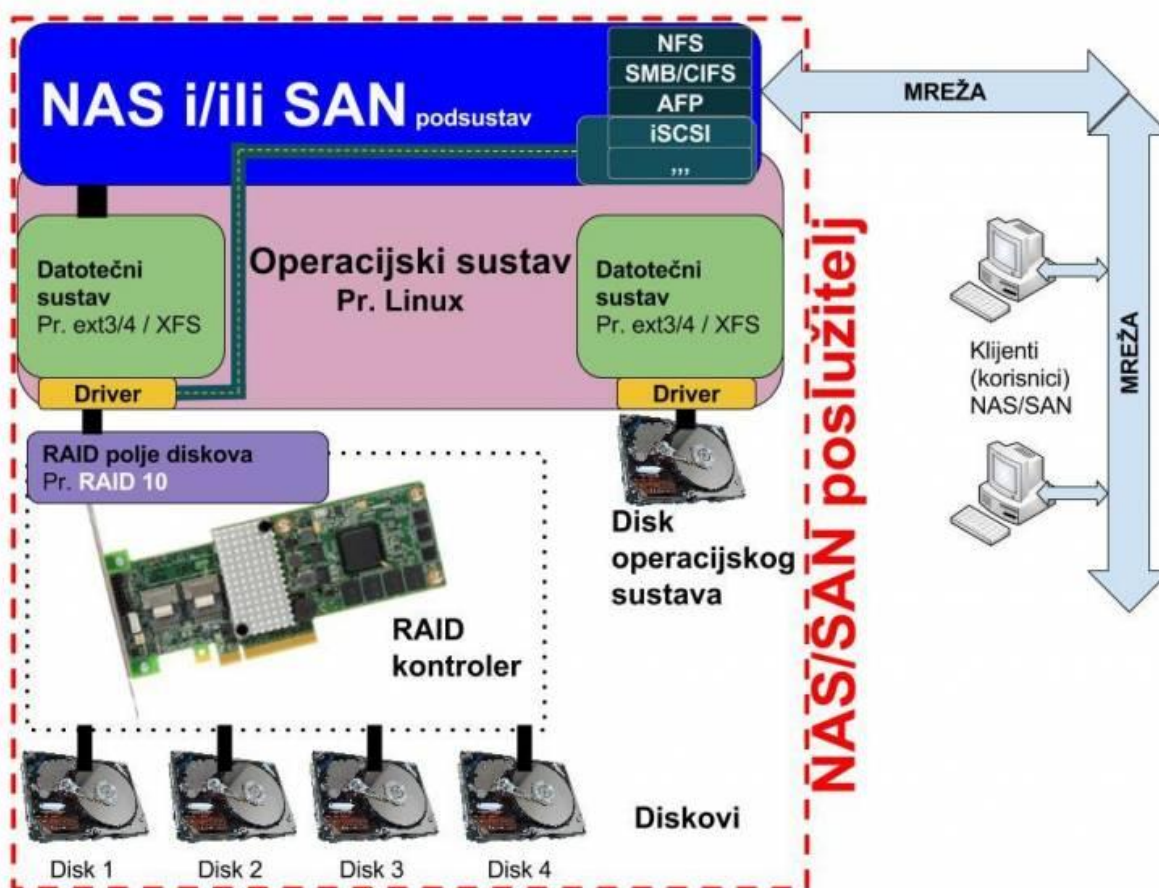
- Firewall
- naprednu konfiguraciju mreže :
  - VLAN-ovi,
  - Agregacija/Bonding - i to nekoliko mogućnosti i protokola
- mogućnost replikacije podataka na sekundarni NAS/SAN sustav
- izradu “snapshota” - u zadanim vremenskim okvirima i ručno a koja je po mogućnosti vidljiva i direktno u operacijskom sustavu klijenata (Pr. kao Windows “Previous Versions”)
- optimizacija svake važnije komponente sustava

- Prava pristupa : kreiranje korisničkih grupa kao i pojedinačnih korisnika, uz mogućnost integracije s Active Directory ili LDAP servisima.
- ...

Sve ove “napredne” opcije i parametri vrlo su važni u realnim radnim uvjetima. Naime svaka mreža i IT sustav su specifični i imaju specifične potrebe. Većina IT sustava je osim toga otvorena za mnoge optimizacije (jer ih obično nitko i nije optimizirao, barem ne na profesionalan način). Mnogi će se zapitati da li je ovo stvarno potrebno. U praksi (s kojom sam se sam susreo) reći ću **DA**. Kreće se od optimizacije:

- operacijskog sustava
- do mrežne razine (uz pretpostavku da imate [pravu mrežnu opremu](#)) koja je preduvjet za bilo kakav imalo ozbiljniji IT sustav. Ovdje se radi o mrežnim parametrima
- preko optimizacije mrežnih protokola i protokola za NAS ili SAN sustav
- do optimizacija na razini ispod NAS ili SAN sustava (prema RAID razini)

Pogledajte i logičku shemu NAS i/ili SAN sustava na slici dolje:



Sada ćemo zaokružiti sve naučeno do sada:

- RAID kontroler ima svoj centralni procesor i specijalizirani procesor (obično su obje funkcionalnosti integrirane u jednom *chipu*). Specijalizirani (dio) procesora se zove RAID [ASIC](#) i on je zadužen za RAID polja.

- RAID kontroler ima i svoju ROM i RAM memoriju kao i disk kontroler na koji se zapravo spajaju diskovi
- RAID kontroler ima svoj “minijaturni” operacijski sustav, niže razine a koji je inicijalno pohranjen u ROM memoriji i iz koje se učitava. Unutar tog “minijaturnog” operacijskog sustava se nalaze metode za pristup svim diskovima spojenim na njega kao i sve potrebno za kreiranje RAID polja unutar kojega će diskovi raditi.
- Promatrajmo diskove koji su u konačnici spojeni na RAID kontroler kao na jednu komponentu koja komunicira s ostatkom računala odnosno operacijskim sustavom i to preko upravljačkog programa (Engl. Driver) za taj RAID kontroler.
- Operacijski sustav preko upravljačkog programa za RAID kontroler vidi samo polja diskova koja su kreirana od strane RAID kontrolera, i to kao jedan jedini disk. Ako smo kao na slici, kreirali RAID 10 polje unutar kojega se nalaze četiri (4) tvrda diska, operacijski sustav (u ovom primjeru je to Linux), vidjeti će jedan jedini tvrdi disk, koji je zapravo cijelo RAID 10 polje diskova. Taj “logički” disk se s točke operacijskog sustava sastoji od svih dijelova od kojih se sastoji bilo koji “normalan” tvrdi disk (trake, cilindri, klasteri i sektori).
- Nadalje taj “logički” disk je potrebno particionirati te kasnije formatirati s nekim datotečnim sustavom.
  - Svi programi dalje koriste taj datotečni sustav za pohranjivanje datoteka, što je i slučaj s programima za dijeljenje datoteka preko mreže (NAS) : NFS, SMB/CIFS , AFP , ...
  - Što se tiče sustava koji pristupaju podacima na razini blokova (SAN sustavi), oni disku pristupaju na razini blokova podataka tj. ispod razine datotečnog sustava.

## U čemu je problem sa standardnim NAS ili SAN sustavima

Što kada se pokvari ovakav NAS ili SAN sustav odnosno poslužitelj ?

Obično ako niste imali još jedan sustav na koji ste izrađivali sigurnosne kopije - ostajete bez svih podataka.

Koliko vam je to važno - da izgubite sve podatke važne za poslovanje vaše tvrtke ?.

## Slijedeći korak: Klasterski i/ili redundantni NAS sustavi

Što nam omogućavaju ovakvi sustavi ?

Osim sigurnosti, jer se sada svi podaci mogu zapisivati na dva ili više uređaja istovremeno, dolazimo i do njihovih ograničavajućih faktora, a to su :

- Omogućavaju horizontalno skaliranje (nadogradnju) ali uz više troškove - znatno više od cijene samog drugog uređaja
- Ograničeni su na proširenje prostora tj. kapaciteta (proširenje dodavanjem diskova ili dodatnih uređaja), Pri tome proširenje dodavanjem dodatnih uređaja (ako je to uopće moguće jer za svaki model odnosno seriju redundantnih NAS uređaja postoji ograničenje do koliko se mogu proširivati) - cijene ovdje odlaze u nebo.
- U konačnici daju nam redundanciju (zalihost/sigurnost od gubitka podataka) uz ekstra cijenu.

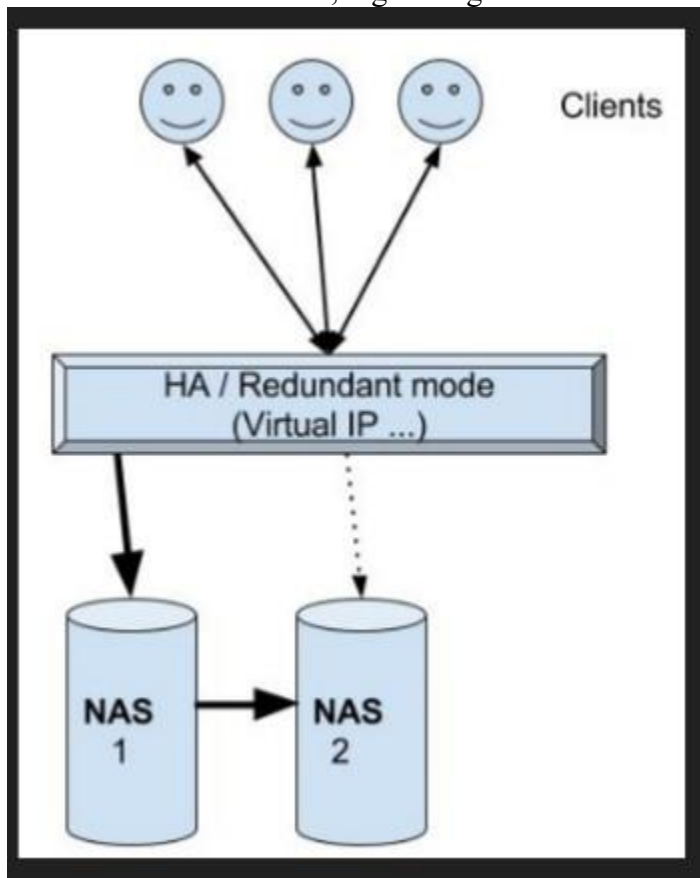
- Uz što veću cijenu, dobivamo i veću brzinu rada
- To su sve uglavnom rješenja koja su zaštićena i zatvorenog dizajna od strane proizvođača poput (EMC, IBM, ...)

## Redundantni ili Klasterizirani NAS Sustavi

### Redundantni NAS Sustavi

Redundantni sustavi su nešto jednostavnijeg dizajna jer se ispred njih logički nalazi sustav koji osigurava pristup jednoj jedinoj virtualnoj IP adresi kojoj klijenti i pristupaju (postoje i drugačije implementacije ali ova je najčešća). U pozadini se redundantni NAS sustav brine da se svi podaci uredno kopiraju s prvog (NAS-1) na drugi (NAS-2) NAS sustav. U slučaju kvara prvog (NAS-1) sustava, drugi (NAS-2) preuzima njegovu funkciju i svi podaci su sačuvani. Ovakvi sustavi su često izvedeni sa samo dva NAS sustava i rade na principu : Active-Standby (jedan je aktivan a drugi je pričuva). Kako logički izgledaju ovakvi sustavi ?

Redundantni NAS sustav, logički izgleda ovako:

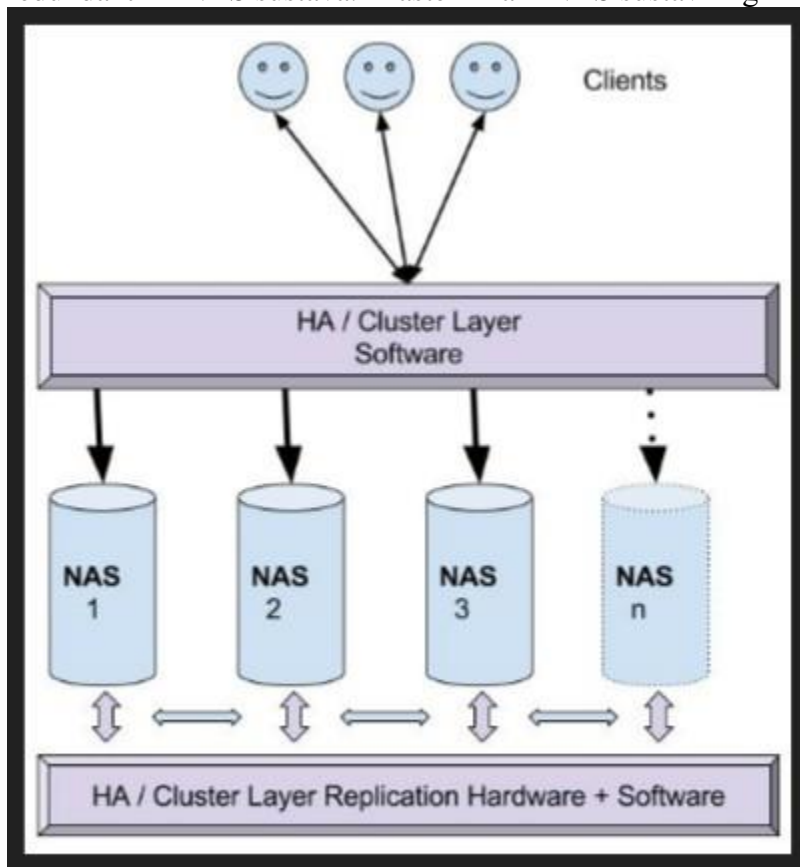


### Klasterizirani NAS Sustavi

NAS sustavi koji se nalaze u klasteru (grozd) su obično znatno kompleksnijeg dizajna koji uključuje i razne dodatne hardverske i softverske komponente.

Svi klijenti u pravilu pristupaju vršnoj klusterskoj komponenti, koja je često izvedena samo u softveru a koja se brine za raspodjelu podataka unutar klastera, na pojedine NAS uređaje.

Sama replikacija tih (odnosno svih) podataka između pojedinih NAS sustava se često izvodi i u softveru i na specijaliziranom hardveru (na slici bi to odgovaralo donjem sloju). Zbog ovakvog, složenog dizajna i potrebe za posebnim hardverom i njihova cijena je poprilično veća od redundantnih NAS sustava. Klasterizirani NAS sustavi logički izgledaju ovako:



## Softverska rješenja

Postoje i mnoga *Open Source* softverska rješenja koja nam omogućavaju osnovnu redundanciju ili klasterizirane NAS sustave. Neki od njih su :

- **GlusterFS** : Omogućava osnovne i nekoliko naprednih razina redundancije :
  - mirror - poput RAID 1 između dva NAS sustava (poslužitelja) - min. 2 poslužitelja,
  - stripe - poput RAID 0 između dva NAS sustava (poslužitelja) - min. 2 poslužitelja,
  - mirror + stripe - poput RAID 10 između dva para NAS sustava (poslužitelja) - min. 4 poslužitelja (1 i 2 u RAID 1, 3 i 4 u RAID 1 , oba para poslužitelja (1,2 + 3,4), vršno u RAID 0, što zajedno čini RAID 10)
- **pNFS** (Parallel NFS - od verzije NFS v.4.1+) :Paralelni/Distribuirani NFS - stabilna (produksijska verzija ) je još u izradi
- **OCFS2** (Oracle open source) : ima slične mogućnosti kao **GlusterFS**
- ...

Svaki od njih ima svoje prednosti i mane kao i ciljanu upotrebu (za koju je i razvijan ili se pokazao kao vrlo dobar)

## Redundantni ili Klasterizirani SAN Sustavi

Slično kao i za redundantne ili klasterizirane NAS sustave - osim sigurnosti jer se sada svi podaci mogu zapisivati na dva ili više uređaja istovremeno, ovdje imamo sljedeće mogućnosti odnosno ograničenja :

- Omogućavaju veće horizontalno skaliranje (nadogradnju) ali uz znatno više troškove - znatno više od cijena za NAS sustave.
- Ograničeni su na ekspanziju prostora tj. kapaciteta (proširenje dodavanjem diskova ili dodatnih uređaja), Pri tome proširenje dodavanjem dodatnih uređaja (ako je to uopće moguće jer za svaki model odnosno seriju redundantnih NAS uređaja postoji ograničenje do koliko se mogu proširivati) - cijene ovdje odlaze u nebo.
- U konačnici daju nam redundanciju (zalihost/sigurnost od gubitka podataka) uz ekstremno visoku cijenu i vrlo kompleksan dizajn.
- u pravilu uz što veću cijenu, dobivamo i veću brzinu rada
- To su sve uglavnom rješenja koja su zaštićena i zatvorenog dizajna od strane proizvođača poput (EMC, IBM, ...)

## Softverska rješenja za SAN i klusterske SAN sustave

I u ovoj kategoriji imamo nekoliko opensource rješenja koje možete proučiti, a koja su dosta česta u upotrebi - i to obično u kombinaciji s drugim elementima odnosno komponentama (ovisno da li se radi o SAN ili klusterskom SAN rješenju ).

- **DRBD8** (Distribuirani - replicirani “Block Device”) - “Distributed Replicated Block Device” : - praktično RAID1 (mirror) preko mreže, prema principu : Primari poslužitelj → Sekundarni poslužitelj. Potrebna su dva poslužitelja ( nije za sve primjene !)
  - **DRBD9** (u aktivnom razvoju) : omogućava rad s više poslužitelja, višestruke replikacije i sl.
- **iscsid (open-iscsi)** (iSCSI initiator) servis/daemon za Linux (sam po sebi nije redundantan već pruža osnovnu iSCSI funkcionalnost )
  - **device-mapper-multipath** - DM-Multipath (“Device Mapper Multipathing”) servis/daemon koji omogućava redundanciju (ili load balancing) prema iSCSI uređajima (SAN storage-ima) (i dalje je pitanje kako sinkronizirati dva ili više SAN storage-a) - koristi se najčešće kod active/passive SAN sustava
  - **ALUA** (“Asymmetric Logical Unit Assignment”) nudi load balancing prema SAN storage-u (ostaje isto pitanje sinkronizacije SAN storage-a) - koristi se za active/active SAN sustave

## ZFS - negdje između

### Ali prvo malo o ZFS-u i tvrtki Sun Microsystems

ZFS je razvila tvrtka [SUN Microsystems](#), danas u vlasništvu tvrtke **Oracle**. Ideja je bila riješiti gore navedene probleme, uvesti mnoga poboljšanja i mogućnosti koje su do tada bile dostupne samo kao specijalizirana rješenja ili uopće nisu postojala, te sve integrirati u jednom “proizvodu”. ZFS je prema svojoj funkcionalnosti praktično kombinacija:

- Naprednog RAID kontrolera odnosno “Logical Volume Managera” i

- Datotečnog sustava s naprednim sustavom kontrole ( ACL) odnosno “Access Listama” za prava pristupa

Izvorno je bio razvijan unutar tvrtke **SUN Microsystems** kao zatvoreni kod, unutar njihovog UNIX operacijskog sustava **Solaris** 2005. godine. Već slijedeće godine je prebačen u open source pod [CDDL](#) licencom unutar projekta “[OpenSolaris](#)” te je postao sastavni dio Solaris UNIX-a v.10, sredinom 2006. godine.

Naravno, nakon što ih je kupio **Oracle**, već nakon par mjeseci cijeli izvorni kod ZFS-a se više nije održavao od strane Oracle-a. Dakle Oracle je prestao s razvojem “OpenSolaris”-a pa je zajednica morala sav kod prebaciti u novi projekt imena “**Illumos**” ( tu se nalazio i kod ZFS-a ). Zajednica koja je stajala iza projekta “**Illumos**” preuzela je zadnju verziju dostupnog koda te ga nastavila razvijati. Nakon nekog vremena je pokrenut i projekt “**OpenZFS**” koji je prihvatila još veća zajednica programera i korisnika ili i sve veći broj tvrtki. Svi zajedno su nastavili s razvojem open source verzije ZFS-a, koja se razvija i danas.

Kao i većina programa ili sustava koji su izašli iz tvrtke **SUN Microsystems** ZFS je razvijan od strane inženjera za inženjere, na najbolji mogući način, kao nedostižni uzor svim ostalim tvrtkama (barem za 99.9999% njih).

### **Borba s open source licencama**

Pošto je ZFS razvijan pod [CDDL](#) licencom a koja nije kompatibilna s Linux GPL licencom pod kojom je razvijan Linux kernel, već od početka javnog razvoja (krajem 2005.g i početkom 2006.g.) bilo je jasno da se ZFS ne smije direktno implementirati u Linux kernel.

Za linux je osmišljeno privremeno rješenje : upotreba preko FUSE frameworka, unutar kojega su se smjeli pokretati programi s drugim licencama. Problem je bio u tome što se [FUSE](#) izvršava na višoj razini iznad kernela te je samim time znatno sporiji. Ali i ovo je bilo nešto za početak.

Istovremeno s ovom borbom krenulo se u razvoj ZFS-a od nule te je 2013.g. razvijena prva stabilna verzija ( v.0.6.1 ) - iste godine je pokrenut i projekt “**OpenZFS**”. Godine 2016 s Ubuntu Linuxom v.16.0.4, ZFS koji se razvijao u projektu “**OpenZFS**” je uključen u ovu distribuciju Linuxa. Što se tiče drugih open source UNIX operacijskih sustava poput onih koji su razvijani s BSD licencom : FreeBSD, NetBSD, OpenBSD i drugih ovdje nije bilo problema s korištenjem te je ZFS na njima zaživio vrlo brzo te se smatra (zbog godina korištenja i testiranja/popravljanja) kao jedna od najboljih implementacija u open source operacijskim sustavima.

OpenZFS projekt nudi implementaciju ZFS-a za Mac OS X.

**ZFS** je nastao u želji da se riješe problemi koje niti najnapredniji RAID kontroleri nisu mogli riješiti. Osim rješenja problema, željele su se dodati i neke napredne mogućnosti koje su većini korisnika bile poželjne i dobrodošle.

Neki od problema koji su poznati a ZFS ih rješava :

- problemi s RAID5 i RAID6 poljima (pogledajte [WIKI](#))
- problem kada želimo zamijeniti neispravni tvrdi disk s novim diskom (koji na većini RAID kontrolera **mora** biti identičan onome koji se mijenja [pr. po broju glava/cilindara/sektora] )
- problemi odnosno komplikacije kod proširenja RAID polja ovisno o RAID polju
- problem u slučaju da nam se RAID kontroler pokvario te ga moramo zamijeniti s novim (ovo je ponekad nemoguće jer možemo izgubiti sve podatke )
- problem kada nam se pokvari pr. matična ploča (MB) te moramo prebaciti sve diskove i RAID kontroler na novi hardver (ovo ponekad može poći po zlu)



- problem koji nastaje s vremenom - podaci na površini tvrdih diskova postaju nekonzistentni a RAID kontroleri nisu toga svjesni, sve dok ne naiđu na problematični dio površine diska. Ovaj problem se naziva “Data decay” ili “Data rot”. On je znan i kao degradacija podataka na površini diska, a tek najnapredniji RAID kontroleri imaju mogućnost (Engl. disk-scrubbing ) korekcije ovakvih grešaka i to samo do neke granice. Sličan problem nastaje i uslijed grešaka u firmware-u diska ili RAID kontrolera, “fantomskog” zapisivanja (ako podatak nije stvarno zapisan na površinu diska) ili grešaka kod zapisivanja ili čitanja zbog pristupa prema ili od krivih bokova na ili sa površine diska.

Dodatne Funkcionalnosti ZFS-a :

- komprimiranje podataka “u letu”, prema konfigurabilnom tipu (pr. ) i razini kompresije. S obzirom na dostupne algoritme za ovu namjenu i brzine ali i mogućnosti modernih CPU-a, komprimiranje i dekomprimiranje podataka “u letu” je gotovo neprimjetno. ZFS trenutno podržava : LZJB, LZ4, ZLE i GZIP.
- ZFS je i Tzv. “Copy On Write” i “transakcijski” datotečni sustav što znači da se operacije snimanja rade transakcijski (poput [transakcija](#) u SQL bazama podataka). To znači da se svaka operacija zapisivanja završava tek kada je potvrđeno uredno zapisivanje (kada je transakcija uredno završila). Performanse su i dalje zadržane naprednim modelom transakcija te uvođenjem posebnog [ZIL](#) log-a za operacije snimanja. Ovaj “Copy On Write” model uvodi i :
  - mogućnost izrade Tzv. “Snapshota” odnosno snimke stanja diska/podataka u vremenu te mogućnost povratka na bilo koji trenutak kada je izrađen bilo koji od “Snapshot”.
  - mogućnost izrade “klonova” odnosno verzije “snapshota” na kojoj se može i zapisivati
- mogućnost naprednih ACL-a (sigurnosnih postavki/prava na datotečni sustav ali i na NFS share direktno )
- ... te cijeli niz drugih funkcionalnosti

## ZFS u radu

Nakon što smo instalirali ZFS, na njega dodajemo diskove i kreiramo ekvivalente RAID poljima, slično kao što ih dodajemo u neki hardverski RAID kontroler (što se konfiguracije tiče). Tako je moguće kreirati ekvivalente gotovo svim RAID poljima :

- RAID 0 ( ovdje se naziva **stripe** ) ,
- RAID 1 ( ovdje se naziva **mirror** ),
- RAID 5 ( ovdje se naziva **RAID-Z** ili **RAID-Z1** ),
- RAID 6 ( ovdje se naziva **RAID-Z2** ),
- Nešto poput RAID “6” ali s tri paritetna diska umjesto dva - naziva se **RAID-Z3**
- RAID 10
- ...

S jedinom razlikom da to sve radi bez grešaka i problema koje možemo imati na bilo kojem RAID kontroleru a pogotovo ako nismo prethodno testirali sve scenarije u slučaju nekog kvara. Osim toga na današnjem hardveru to sve radi ekstremno brzo a sve je moguće i dodatno drastično ubrzati uvođenjem :

- **L2ARC**-a za ubrzavanje operacija čitanja (read) i /ili



- **ZIL log**-a za ubrzavanje operacija pisanja (write)

S time da se za obje metode (**L2ARC** i **ZIL log**) mogu koristiti zasebni SSD diskovi a koji dodatno mogu biti i u nekom ekvivalentu RAID polja, da bi se dodatno dobilo na brzini i/ili pouzdanosti.

Pošto je jasno da je ZFS povezan s NFS-om a i vrlo jednostavno sa SMB/CIFS ili nekim drugim sustavom za dijeljenje datoteka (NAS sustavom) vidljiva je njegova upotreba kao naprednog NAS sustava. Jedna od naprednih stvari oko ZFS-a je i u tome što se na bilo kojem ZFS polju diskova (ekvivalent RAID polju) može kreirati poseban “Block device” koji se može koristiti kao iSCSI logički uređaj (disk). Taj logički disk je samo potrebno proslijediti nekom od iSCSI “serverskih” servisa/daemonu. Ovime dobivamo upotrebu ZFS-a kao SAN sustava.

Potencijalna mana upotrebe ZFS-a leži u tome što nije trivijalan kao što je više manje korištenje RAID kontrolera i kreiranje nekog RAID polja. U svakom slučaju potrebna su vam neka naprednija predznanja. ZFS-ove mnogobrojne opcije i funkcionalnosti početnicima mogu izgledati komplicirane ali su profesionalcima definitivno vrlo važne.

Na kraju krajeva niti ne želite da vam NAS ili SAN sustav “složi” netko onako usput, za sat-dva, uz svoj svakodnevni posao koji obično nema veze s ovom temom, jer bi mogli zažaliti kada nešto krene po zlu. Namjerno nisam napisao “ako” već “kada” jer je uvijek pitanje vremena kada će doći do problema i dali ćete ih biti u stanju riješiti te koliko vremena i novaca će vam biti potrebno za tu “igru”.

O ZFS-u su napisane knjige i knjige te više nećemo ulaziti dublje u ovaj najbolji “Volume Manager” i datotečni sustav svih vremena. O njemu detaljnije u nekom od slijedećih postova.

## Proces učenja

Ako tek krećete s učenjem, prvo si dajte nekoliko dana da bi dobro savladali:

- osnove Storage tehnologija te napredne mogućnosti
- osnove rada RAID kontrolera te njihove mogućnosti, način rada i dodatne opcije (uz njihovo razumijevanje)

I na kraju dodajte još koji dan za proučavanje foruma koji se bave ovom tematikom, kao i foruma od strane proizvođača s pitanjima i odgovorima vezanim za pojedine (konkretno) modele RAID kontrolera koji vam je ušao u uži izbor (ako ste odlučili da ćete koristiti RAID kontroler a ne ZFS). Kada završite prvu fazu učenja i nakon što ste kupili RAID kontroler, slijedi novo učenje :

- proučite napredne parametre i testirajte ih (istovremeno testirajte i performanse sustava ovisno o promjeni parametara)
- testirajte razne scenarije havarija za barem nekoliko RAID polja (pogledajte dolje za ZFS) i povrata podataka , mjerite i vrijeme koje potrebno da se sve vrati na “staro stanje” - i vrijeme potrebno za “recovery” je ponakad ključno za konačan odabir vrste RAID polja (RAID 1, RAID 5, RAID 6, RAID 10, ...)
- sve dobro i detaljno dokumentirajte (ovo ćete najviše cijeniti kad vam se dogodi prva veća greška - višestruko će se isplatiti dani i dani testiranja i dokumentiranja)

Ako ste krenuli prema ZFS-u tada krenite s proučavanjem osnova:

- [WIKI ZFS info](#)

Proučite si i pripremite upute iz nekoliko izvora te odvojite još par tjedana za upoznavanje i isprobavanje te smišljajte razne scenarije havarija (i smislite/pronađite najbolje rješenje) :

- Kreirajte jedno ZFS polje:
  - a. Mirror (Ekvivalent RAID 1)
    - 0. Kreirajte share (NFS ili SMB/CIFS), dodjelite ovlasti i dobro ih naučite (proučite kako se dodjeljuju, nasljeđuju, ne nasljeđuju i sl.), Zapišite određene podatke i pratite performanse kod zapisivanja i čitanja (s različitim parametrima).
    - 1. Izvadite jedan disk iz ZFS polja, pa ga vratite
    - 2. Ponovite 1. korak ali prije vraćanja diska, obrišite ga
    - 3. Zamijenite mjesta diskovima (prvi na mjesto drugog i sl.)
    - 4. Izvadite sve diskove, reinstalirajte cijelo računalo pa vratite diskove i pokušajte importirati staro ZFS polje
  - b. Obrišite postojeće ZFS polje i kreirajte novo (Pr. RAID-Z)
  - Ponovite sve točke : 0 - 4
  - c. Obrišite postojeće ZFS polje i kreirajte novo (Pr. RAID-Z2)
  - Ponovite sve točke : 0 - 4
  - d. Obrišite postojeće ZFS polje i kreirajte novo (Pr. RAID-10 : 2 x “mirror” u “stripe”)
  - Ponovite sve točke : 0 - 4

Napravite što više scenarija te:

- sve isprobajte (testirajte i performanse) i dokumentirajte
- isprobavajte rad s osnovnim postavkama, pa sve probajte optimizirati (za svaki scenarij) - testirate i stabilnost i izdržljivost ovisno o scenariju i opcijama koje ste mijenjali - uz:
  - restart poslužitelja
  - restart servisa/daemon
  - namjerno stopiranje servisa/daemon (uz ručno pokretanje - naknadno)

U ovim koracima/scenarijima ćete naučiti najviše, te se približiti produkcijskoj primjeni i znanju o ovim sustavima.

## Što je slijedeće

### Problemi klusterskih NAS i SAN sustava

Kao što smo vidjeli u prethodnom poglavlju klusterski NAS i SAN sustavi imaju svoje limitirajuće faktore. Kod većine je to cijena ali i ograničenja skalabilnosti. Naime veći sustavi često trebaju sve veći i veći kapacitet pohrane podataka, koji postaje ili preskup u startu ili zahtjeva vrlo velika ulaganja kod proširenja. I na kraju krajeva svi oni opet imaju svoje limite, najviše sa strane proširenja.

Kod najvećih igrača poput “Cloud” providera pružanje usluge pohrane velike količine podataka pr. za spremanje virtualnih računala i sl. je svakodnevni posao. Proširivost ovakvih sustava je ključna.

Rani odgovor na ovu problematiku je bio razvoj (i kasnija upotreba) sustava koji uopće ne rade na način na koji rade tradicionalni klusterski NAS ili SAN sustavi.

## Object storage

I rodio se “Object storage”, koji podatke “promatra” i pohranjuje kao objekte, za razliku od tradicionalnih sustava kod kojih postoji neka struktura datoteka i direktorija (odnosno klasičan datotečni sustav) kod NAS sustava. Ovo je drugačije i od SAN sustava koji rade s blokovima podataka koji se spremaju u sektore na disku (logičkom ili fizičkom). Kao što RAID kontroler “razlama” neku datoteku na male blokove podataka koje dalje raspoređuje na diskove, ovisno o RAID polju, tako i ovi sustavi “razlamaju” podatke na tzv. objekte (uz pripadajuće metapodatke), koje onda raspoređuju na poslužitelje u klasteru. Objektni “storage” trebao bi nam nuditi, skalabilni (proširivi) sustav otporan na greške. Ovakvi sustavi su se počeli znatnije razvijati od 1995. godine iako su neki radovi i ideje nastali i znatno ranije. Prvo komercijalno rješenje je razvila tvrtka “Centra Technology” koju je odmah kupila tvrtka “EMC” te je 2002. izbacila na tržište pod tržišnim nazivom “[EMC Centera](#)”. Ova linija proizvoda se i danas razvija. Smatra se da se u razvoj ove tehnologije od strane neovisnih investitora u prvim godinama uložilo oko 300 milijuna dolara (ova cifra je rasla sve više). Ne računajući ulaganja tvrtki poput : DataDirect Networks, Centra, Atmos, HDS, EMC2, HP, IBM, NetApp, Redhat i drugih a kasnije i od strane “Cloud providera” poput : Amazon AWS, Microsoft (Microsoft Azure), Google (Google Cloud Storage) i drugih.

Pogledajmo listu nekoliko visoko skalabilnih, redundantnih “Object storage” sustava dostupnih pod nekom od “open source” licenci:

- CEPH ([info](#))
- Lustre ([info](#))
- LizardFS ([info](#))
- Hadoop Distributed File System([info](#))
- Moose File System ([info](#))
- Quantcast File System ([info](#))
- ...

Kod većih sustava, kao i kod sustava kod kojih korisnici NE žele kupovati super skupi hardver i softver za “Object Storage” sustave, jedno od open source rješenja je “**CEPH**” o kojemu ćemo govoriti dalje u tekstu.

## CEPH



Ceph je distribuirani objektni sustav za pohranu podataka (Engl. Storage) koji je dizajniran za postizanje odličnih performansi, te sustav koji je visoko dostupan i pouzdan. Osim toga on je krajnje skalabilan odnosno proširiv do razine [Exabyte-a](#).

Ovo je sustav koji je zbog svog dizajna otporan na greške i kvarove cijelih poslužitelja i/ili pojedinačnih diskova ili grupe diskova, a u većim implementacijama, cijelih ormara punih poslužitelja pa čak i cijelih podatkovnih centara a samim time i desetcima, stotinama ili tisućama diskova. Sve ovisno o konfiguraciji i raspoloživoj opremi.

Više informacije možete pronaći na : <http://ceph.com/>

## Malo o povijesti CEPH-a

Razvio ga je [Sage Weil](#) kao temu za doktorski rad na sveučilištu “University of California, Santa Cruz”. Razvoj se nastavio u tvrtki “Inktank”. Navedenu tvrtku je kupio “RedHat”, 30.04.2014 (za 175 milijuna U\$ u gotovini). Tvrtka “Red Hat” ga nastavlja razvijati do danas (kao i zajednica koja ga koristi). Projekt je i dalje, i ostati će “open source”.

## Da li postoji i podrška od strane proizvođača hardvera

Naravno, vrlo brzo nakon ućlanjenja u obitelj “Red Hat” svi važniji proizvođači hardvera počeli su nuditi sustave koji su certificirani za CEPH, pr. :

- [Supermicro](#)
- [HP](#)
- [DELL](#)
- ... i mnogi drugi

Osim navedenog hardvera, CEPH se može koristiti i na bilo kojem hardveru koji imate a na kojem se može pokretati bilo koja **RedHat** ili **Debian** bazirana distribucija Linuxa, imalo novije generacije. Dakle dostupni su RPM i Debian [paketi](#). Osim toga dostupan je i izvorni kod CEPH-a, pa je sve moguće kompajlirati i za druge distribucije Linuxa.

## Integracija

CEPH klijent se već standardno nalazi unutar Linux kernela. Server je dostupan ionako kao open source na stranici : <http://ceph.com/resources/downloads/>.

Osim navedenog CEPH je trenutno integriran s dvije platforme za virtualizaciju:

- **Open Stack** - [info](#) :
  - Integriran je sa : **Nova**, **Cinder** i **Glance** za “Block storage”
  - Integriran je sa **Keystone** i **Swift** za “Object storage”
- **Proxmox VE** - pogledajte [info](#) :
  - “Block storage” za virtualna računala i za Linux kontejnere

## Tko ga trenutno koristi

Koriste ga i najveći igrači, poput :

- Amazon AWS - prema nekim informacijama, koristi se za neke dijelove [S3 Storage](#) sustava
- Facebook - za neke dijelove sustava
- CERN - prema podacima od prošle godine - koriste ga za ukupno 1+ PB (za spremanje podataka)
- DreamHost (Web hosting provider) :
  - 2+ PB za S3
  - 3+ PB kao “Block Device” - za virtualke
- ... i mnogi drugi (mnogi i ne žele iznositi što točno koriste iz sigurnosnih razloga)

## Za što se sve može koristiti CEPH

CEPH iako radi s objektima na najnižoj razini, na vršnoj se može koristiti za tri različite “upotrebe”, i to :

- Kao “**Block Device**” i to ako se koristi kao “Rados Block Device” ( RBD ) - vidljiv dalje kao “Block Device” ili logički disk koji se koristi za opću upotrebu (pr. za spremanje diskova virtualki i sl.)
- Kao “**Object Storage**” preko “RADOSGW”-a, a koji je “S3” i “Swift” kompatibilan - najčešće se koristi za snimanje/čitanje datoteka bilo kojeg tipa preko web-a (korištenjem “put” ili “get” metoda)
- Kao “**Filesystem**” tj. direktno kao datotečni sustav, preko “CEPHFS” - može se “mountati” kao običan datotečni sustav

Pogledajte i malo više detalja :

CEPH OBJECT STORE	CEPH BLOCK DEVICE	CEPH FILESYSTEM
<ul style="list-style-type: none"> <li>• RESTful Interface</li> <li>• S3- and Swift-compliant APIs</li> <li>• S3-style subdomains</li> <li>• Unified S3/Swift namespace</li> <li>• User management</li> <li>• Usage tracking</li> <li>• Striped objects</li> <li>• Cloud solution integration</li> <li>• Multi-site deployment</li> <li>• Disaster recovery</li> </ul>	<ul style="list-style-type: none"> <li>• Thin-provisioned</li> <li>• Images up to 16 exabytes</li> <li>• Configurable striping</li> <li>• In-memory caching</li> <li>• Snapshots</li> <li>• Copy-on-write cloning</li> <li>• Kernel driver support</li> <li>• KVM/libvirt support</li> <li>• Back-end for cloud solutions</li> <li>• Incremental backup</li> </ul>	<ul style="list-style-type: none"> <li>• POSIX-compliant semantics</li> <li>• Separates metadata from data</li> <li>• Dynamic rebalancing</li> <li>• Subdirectory snapshots</li> <li>• Configurable striping</li> <li>• Kernel driver support</li> <li>• FUSE support</li> <li>• NFS/CIFS deployable</li> <li>• Use with Hadoop (replace HDFS)</li> </ul>

Odabirom pojedinog modela :

- “**CEPH Block Device**”
- “**CEPH Object Storage**”
- “**CEPH Filesystem**”

moramo koristiti i dodatne servise odnosno funkcionalnosti koje su nužne za ovakav rad. Prema tome potrebno je detaljnije se upoznati sa zahtjevima i načinom implementacije te konfiguracije svakoga od njih.

## Prednosti CEPH-a

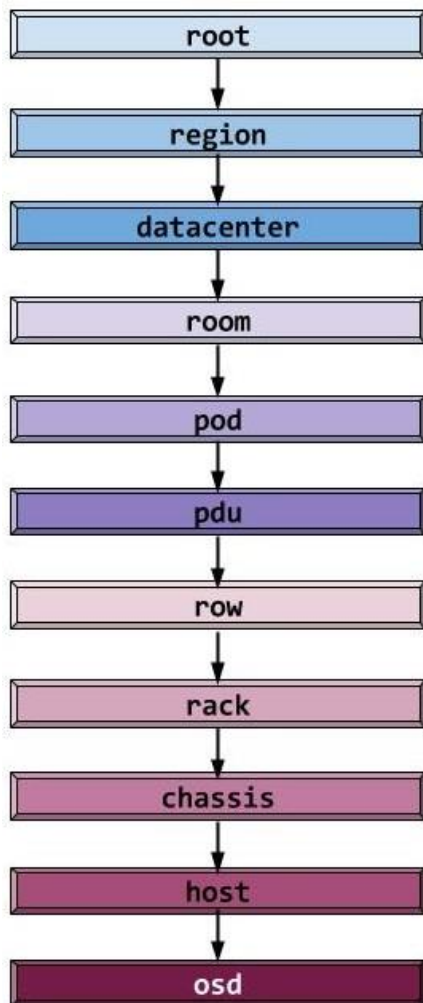
Osnovne prednosti CEPH-a ([i u kombinaciji s Proxmox VE platformom za virtualizaciju](#)) su :

- (Relativno) Jednostavan setup i management iz naredbene linije i grafičkog sučelja Proxmox VE
- “Thin provisioning” (minimalno zauzeće stvarnog diskovnog prostora s podacima)
- Izrada Snapshot-a podataka (datoteka) u letu (dok se radi na njima)
- Automatsko popravljavanje grešaka u radu (kod ispada diska, poslužitelja i sl.)
- Ne postoji niti jedna komponenta sustava koja nije redundantna (zalihost)
- Sustav je skalabilan ( proširiv ) do razine Exabyte-a
- Moguća je konfiguracija više segmenata (Engl. Ceph Pools) polja za pohranu podataka, te razina performansi/replikacije za svaki segment
- Svi podaci unutar polja su replicirani, čineći cijelo polje otpornim na kvarove
- CEPH je moguće instalirati i na pristupačan hardver
- Nema potrebe za RAID kontrolerima ( “zabranjena” je njihova upotreba - kao i kod ZFS-a (kod kojega je to izričito **ZABRANJENO**))

- CEPH je razvijan kao “open source” prema licenci [LGPL 2.1](#)

## Kako se podaci distribuiranju unutar cijelog CEPH clustera

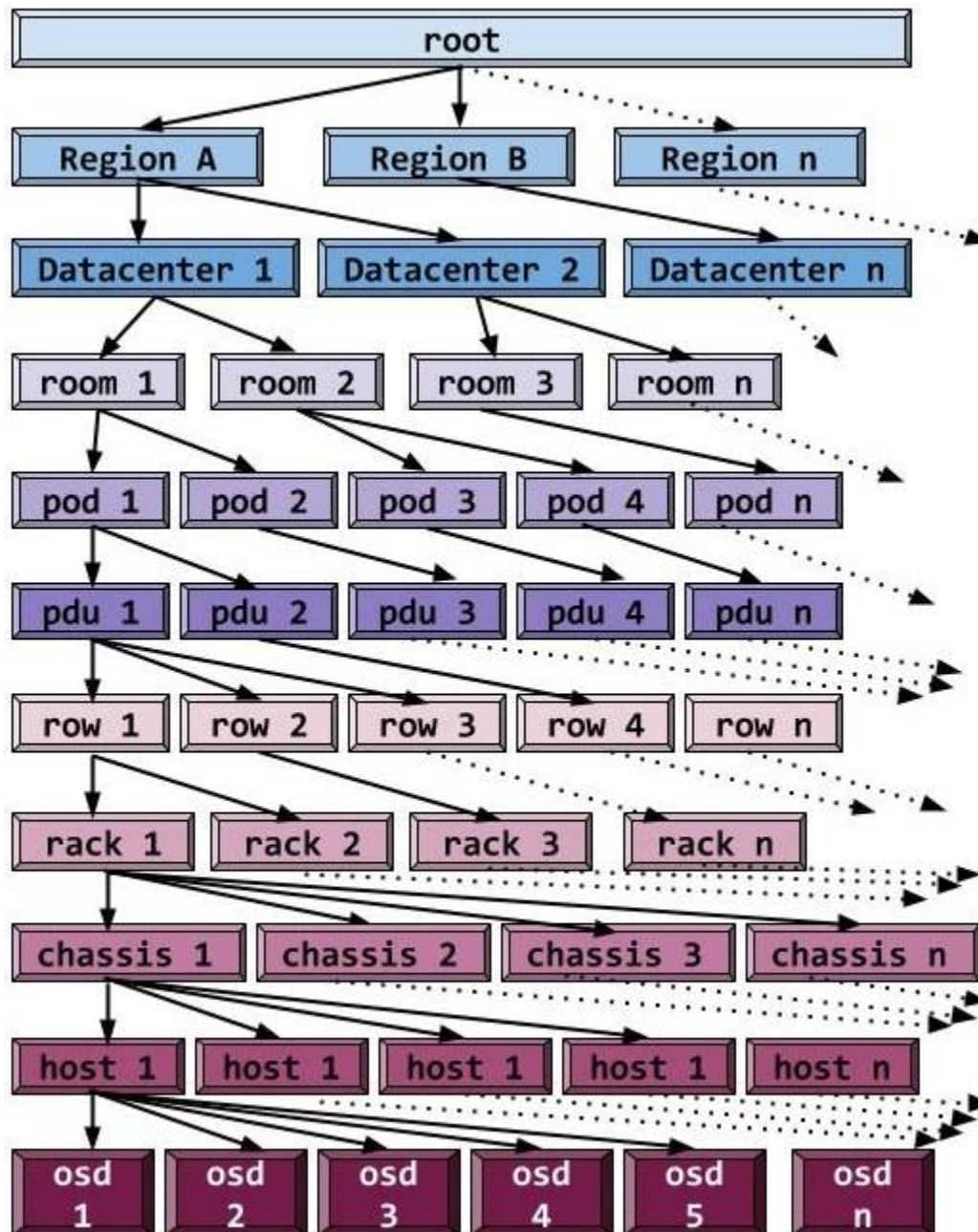
Koristi se Tzv. [CRUSH algoritam](#) i pripadajuća “CRUSH” tablica (koja je distribuirana na više poslužitelja) a koja je zadužen za distribuciju, replikaciju i redistribuciju podataka unutar CEPH clustera. CRUSH je dizajniran da omogućava raznoliku upotrebu, ovisno o veličini implementacije. Prema tome postoje “CRUSH” tipovi koji opisuju fizičku poziciju CEPH-a unutar cijelog CEPH clustera. Drugim riječima definiramo fizičku hijerarhijsku strukturu svakog elementa unutar hijerarhije:



- **root** (predstavlja vršnu komponentu cijelog CEPH-a - nazovimo ju “cijelom planetom”)
- **region** (predstavlja prvu nižu hijerarhiju - recimo kontinent)
- **datacenter** (predstavlja pojedini podatkovni centar)
- **room** (predstavlja “sobu” unutar podatkovnog centra)
- **pod** (predstavlja logičku podjelu unutar jedne “serverske” sobe) - može predstavljati i jedan dio podatkovnog centra koji može biti podjeljen na više ovakvih potencijalno nezavisnih (što se tiče mreže, napajanje, klimatizacije i sl.) cjelina.
- **pdu** “Power Distribution Unit” odnosno podjela prema izvoru napajanja (u podatkovnim centrima ih imamo više pa je ovo dobrodošla dodatna razdioba)
- **row** (predstavlja jedan red s ormarima punim poslužitelja)
- **rack** (predstavlja jedan ormar s poslužiteljima)
- **chassis** (predstavlja jedno kućište unutar kojega može biti više poslužitelja - misli se na “Blade” učilišta)
- **host** (predstavlja jedan poslužitelj)
- **osd** (predstavlja, u konačnici, pojedinačni disk)



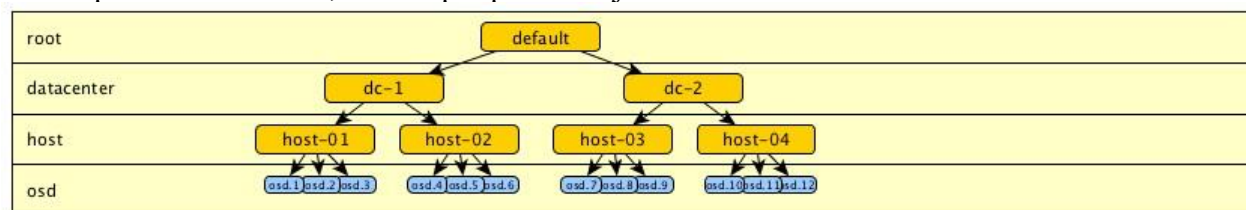
Osim toga u svakoj kategoriji u hijerarhiji, može biti i više elemenata na istoj razini - poput ovoga na slici dolje:



Ovakav hijerarhijski model nam omogućava stvarno raznolike scenarije upotrebe. Stoga CEPH može biti implementiran od najmanjih sustava - pr. s minimalno tri (3) poslužitelja s diskovima a s druge strane na sustavima koji imaju tisuće poslužitelja s diskovima, koji su raspoređeni u konačnici na veliki broj podatkovnih centara.

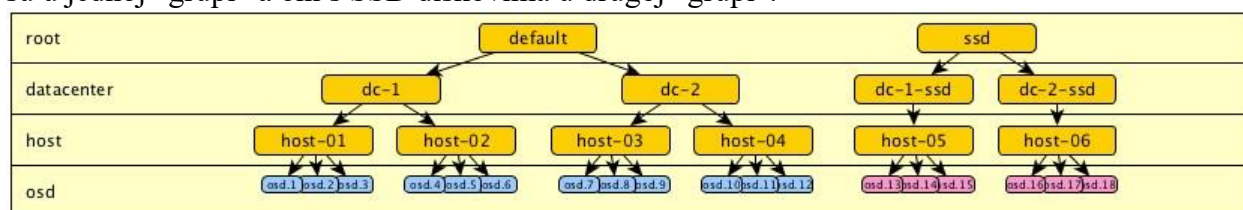
## Pogledajmo nekoliko mogućih scenarija :

### 1. Dva podatkovna centra, svaki s par poslužitelja



Vidljivo je da unutar svakog podatkovnog centra (**datacenter**) imamo dva poslužitelja (**host**) od kojih svaki ima po tri tvrda diska (**osd**).

2. Prošireni scenarij u kojemu isto imamo dva podatkovna centra ali sada imamo poslužitelje s običnim (tvrdim diskovima) i poslužitelje sa SSD diskovima. Poslužitelji s “običnim” diskovima su u jednoj “grupi” a oni s SSD diskovima u drugoj “grupi”.



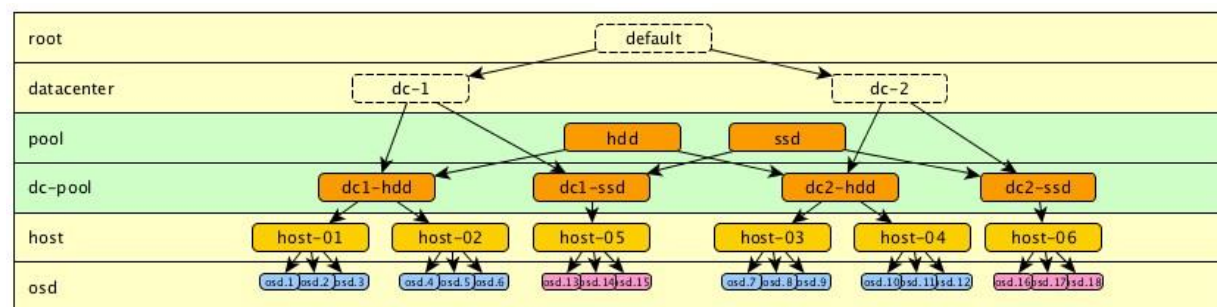
2.1. Logička shema dolje prikazuje i inicijalizaciju Tzv. “**Pool**”-a.

U CEPH terminologiji “Pool” je ono što bi u RAID-u bilo RAID polje diskova.

Moguće je imati više “Pool”-ova, svaki sa svojom konfiguracijom.

Svaki pojedini **Pool** može biti za svoju namjenu:

- brzina
- pouzdanost
- vrijeme odziva
- georeplikacija
- ...



U primjeru na slici u svakom podatkovnom centru imamo poslužitelje sa SSD i poslužitelje s običnim tvrdim diskovima.

- Vršno Pool “**hdd**” koristi sve poslužitelje koji imaju obične diskove
- Vršno Pool “**ssd**” koristi sve poslužitelje koji imaju SSD diskove

Kod kreiranja **Pool**-a (to je korak koji možete vidjeti u tekstu o radu CEPH-a) odabiremo koliko replika će imati, kao i druge parametre.

Slike su preuzete sa <http://cephnotes.ksperis.com/blog/2015/02/02/crushmap-example-of-a-hierarchical-cluster-map>



## Kako se zapisuju podaci na CEPH cluster

Nakon što je definirana hijerarhijska struktura za CEPH cluster (CRUSH) te kreiran ekvivalent RAID polja koji se prema CEPH terminologiji naziva “**Pool**” sve je spremno za rad (to je opisano negdje od koraka ”[CEPH pools](#)“). Pojednostavljeno svaka datoteka koja se zapisuje razloma se na manje blokove koji se onda u konačnici zapisuju odnosno **distribuiraju** na dostupne poslužitelje i njihove diskove. Dakle ako smo za određeni **Pool** na kojem radimo, kod kreiranja odabrali da je broj **Replika** odnosno prema CEPH terminologiji “**CEPH Pool Size**” jednak tri (3) to znači da se podaci zapisuju na određeni poslužitelj a potom na još druga dva (2) poslužitelja. Tako da ćemo u ovom slučaju isti podatak imati sveukupno na tri (3) mjesta. Veličina bloka je standardno 4 MB ali se može promijeniti do razine više MB - ovisno o vrsti podataka koje zapisujemo ili čitamo. To znači da je za neke primjene ova veličina zadovoljavajuća a za neke je ova veličina premalena jer se zapisuju ili čitaju podaci koji zahtijevaju dohvaćanje većih blokova podataka odjednom. Promjenom veličina bloka možemo poboljšati performanse i smanjiti opterećenje sustava - zbog smanjenja broja operacija dohvaćanja velikog broja malih objekata.

Ulazno/izlazne operacije prema diskovnom sustavu kod pisanja ili čitanja se zovu [IOPS-i](#). Klasični (magnetski) odnosno “mehanički” diskovi su znatnije pogođeni ovim operacijama od SSD diskova. Dakle SSD diskovi u prosjeku mogu podnijeti desetke, stotine i tisuće puta više ulazno/izlaznih operacija u sekundi, od mehaničkih/magnetskih diskova.

### Proces distribucije podataka

Podaci se distribuiraju na cijeli CEPH cluster, sve njegove poslužitelje i njima dostupne tvrde diskove, te se istovremeno radi replikacija, svakog bloka podataka na drugi poslužitelj odnosno disk na njemu. Sve prema tome kako je konfigurirana hijerarhija za CRUSH te koliko replika smo odabrali za određeno CEPH polje odnosno **Pool**.

Proces zapisivanja i dodatno replikacije, radi se transakcijski (pogledajte ZFS i transakcijski model) - zbog konzistentnosti podataka. Kod procesa čitanja se također prema klsterskoj tablici i CRUSH algoritmu zna (određuje/izračunava) koji blok podataka je završio na kojem poslužitelju, i na kojem disku na njemu, te se počinje s čitanjem blokova podataka - sa svih poslužitelja i svih diskova. U konačnici sve se svodi na to da se podaci zapisuju na sve poslužitelje te se kod čitanja također čitaju sa svih njih. Ovime se znatno povećavaju performanse : što više poslužitelja to je brže zapisivanje ili čitanje.

### Redistribucija podataka

Što u slučajevima kada se primjerice :

- poslužitelj gasi (zbog kvara, održavanje ili bilo kojeg razloga) ili se dodaje novi
- dodaje se novi poslužitelj
- dodaju se novi diskovi u postojeće poslužitelje ili se neki diskovi vade

Tada CEPH radi Tzv. redistribuciju podataka.

Pogledajte sliku upotrebe CEPH-a na **Proxmox VE** platformi za virtualizaciju:

Name	Type	Status	weight	reweight	Used	
					%	Total
226x	host					
osd.15	osd	up/in	0.809998	1	0.28	832.34GB
osd.14	osd	up/in	0.809998	1	0.30	832.34GB
osd.13	osd	up/in	0.809998	1	0.29	832.34GB
osd.12	osd	up/in	0.809998	1	0.31	832.34GB
osd.11	osd	up/in	0.809998	1	0.29	832.34GB
osd.10	osd	up/in	0.809998	1	0.27	832.34GB
osd.9	osd	up/in	0.809998	1	0.30	832.34GB
osd.8	osd	up/in	0.809998	1	0.27	832.34GB
224x	host					
osd.7	osd	up/in	0.809998	1	0.26	832.34GB
osd.6	osd	up/in	0.809998	1	0.29	832.34GB
osd.5	osd	up/in	0.809998	1	0.31	832.34GB
osd.4	osd	up/in	0.809998	1	0.30	832.34GB
osd.3	osd	up/in	0.809998	1	0.29	832.34GB
osd.2	osd	up/in	0.809998	1	0.26	832.34GB
osd.1	osd	up/in	0.809998	1	0.30	832.34GB
osd.0	osd	up/in	0.809998	1	0.31	832.34GB

Na slici su vidljiva samo dva poslužitelja **225x** i **224x** (iako su u testu bila tri (i **223x**)) od njih svaki ima po 8 tvrdih diskova :

Poslužitelj	Diskovi	Kapacitet diskova
225x	OSD.8 OSD.9 OSD.10 OSD.11 OSD.12 OSD.13 OSD.14 OSD.15	(svaki tvrdi disk ima dostupni kapacitet 832GB)
224x	OSD.0 OSD.1 OSD.2 OSD.3 OSD.4 OSD.5 OSD.6 OSD.7	(svaki tvrdi disk ima dostupni kapacitet 832GB)
223x	OSD.16 OSD.17 OSD.18 OSD.19 OSD.20 OSD.21 OSD.22 OSD.23	(svaki tvrdi disk ima dostupni kapacitet 832GB)

Pogledajte stupac “**Used**” i to postotke (kreću se od 0.27 do 0.31).

Kod dobro balansiranog sustava, postotak zauzeća (upotrebe) svih diskova mora biti podjednak. Za to su zaduženi automatizmi o kojima ćemo malo kasnije.

Dodavanjem novog diska, vađenjem jednog od njih ili dodavanjem/izbacivanjem cijelog poslužitelja sa svim diskovima CEPH kreće u redistribuciju svih podataka. To znači da ako smo recimo dodali novi poslužitelj s osam diskova (detaljnije se radi i o koeficijentu svakog diska ovisno o njegovom kapacitetu i drugim parametrima) podaci se preraspoređuju unutar cijelog klastera i svih diskova, tako da svi diskovi na svim poslužiteljima budu podjednako zauzeti.

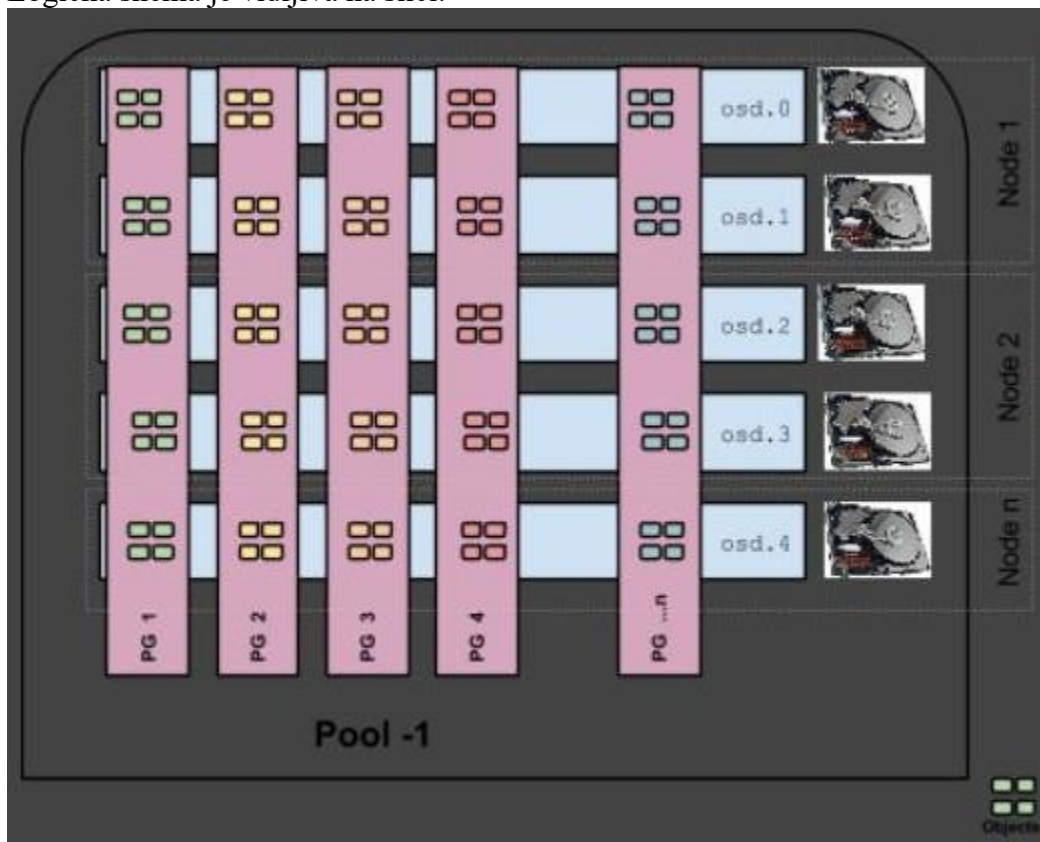
Ovo je vrlo važno jer se nakon dovršetka redistribucije podaci tada počinju zapisivati ili čitati i s tog novog poslužitelja ili novog diska, ravnomjerno koristeći sve resurse (poslužitelje i diskove) klastera. Za Redistribuciju kao i za replikaciju podataka, koristi se (preporuča) zasebna mreža - da se ne opterećuje “radna” mreža.

Prema CEPH preporukama, potrebno je imati dvije zasebne mreže :

- “Public Network” - preko nje čitamo i pišemo podatke na CEPH
- “Cluster Network” - preko nje se odrađuju sve ostale radnje poput redistribucije i replikacije podataka

## Logička shema cijelog sustava

Logička shema je vidljiva na slici:



Opis :

- Podaci se spremaju kao objekti
- Objekti se nalaze unutar **Pool**-a
  - Standardna veličina objekta je 4MB

- Objekti se grupiraju u “**Placement Grupe**” (PG). **Placement Grupe** su distribuirane preko više OSD-ova (diskova)
- OSD-ovi se koriste za stvarnu distribuciju (“read” i “write” operacija) objekata na tvrde diskove
- “CRUSH” tablica/konfiguracija se koristi za kreiranje i kasniju upotrebu i distribuciju objekata (podataka) unutar svakog pojedinog “**Pool-a**” za cijeli CEPH klaster. (Moguće je imati i više **Pool**-ova s različitim konfiguracijama)

**Pool** promatrajte kao RAID polje.

### Malo detaljnije

Iako se podaci u konačnici zapisuju kao objekti, odnosno najmanji blok podataka je jedan objekt, standardne veličine 4MB, objekti se prvo grupiraju u Tzv. “Placement” grupe. Ove “Placement” grupe prema tome povezuju niz objekata koji su dalje raspoređeni na niz OSD-ova. Pohrana objekata na OSD-ove znači pohranu na niz tvrdih diskova, raspoređenih na više poslužitelja - ovisno o Pool-u i hijerarhijskoj strukturi definiranoj u CRUSH tablici/konfiguraciji.

Prisjetimo se da “CRUSH maps” tablica/konfiguracija definira fizičku topologiju cijelog CEPH klastera koju koristi CRUSH algoritam za određivanje (izračun) točnih pozicija na koje će se podaci (u konačnici objekti) i njihove replike spremati odnosno čitati.

Sve operacije čitanja i pisanja se zapravo rade na razini svake pojedine “Placement” grupe a ne na razini svakog pojedinog objekta. U protivnom bi rad na razini svakog pojedinog objekta uz dohvaćanje metapodataka za svaki objekt drastično usporilo cijeli sustav.

“Placement” grupe rješavaju problem s performansama, jer se transakcije događaju na razini PG-a, kao i pohranjivanje ili baratanje s pripadajućim metapodacima, koje su definirani za cijelu placement grupu a n pojedini objekt u njoj. CEPH kod čitanja ili pisanja radi na razini “placement” grupa i njihovih metapodataka (koji ih opisuju), i to transakcijski.

Osim poboljšanja performansi, uvođenjem “Placement” grupa, poboljšala se i skalabilnost (proširivost) cijelog CEPH sustava. Odnos između broja objekata i broja “Placement” grupa se može okvirno izračunati ili utvrditi testiranjem. Prema preporukama, osnovna računica je :

### PG Calculation (formula):

$$PG = \frac{Nr. of OSDs \cdot 100}{Nr. of replicas}$$

Za što bolji odabir odnosno izračun broja “Placement grupa” potrebo je uzeti i druge parametre (o tome [kasnije](#)).

### Pool i PG

Možemo promatrati “Placement” grupe (PG) kao segmente unutar svakog logičkog **Pool-a** odnosno polja (objekata) na koje se logički “spaja” svaki CEPH klijent za čitanje ili pisanje na CEPH klaster. Dakle CEPH vršno gledano, prema podatke unutar **Pool-a**, koji predstavlja logičku grupu PG-ova.

Pool se brine i o tome koliko je primjerice replika potrebno izraditi kod svakog zapisivanje podataka. CEPH može raditi i “snapshot” **Pool**-a, u bilo kojem trenutku - kao “snimku stanja u vremenu”.

## **CEPH Block Device (Rados Block Device) tj. RBD**

Mi ćemo se dalje u tekstu fokusirati na upotrebu “**CEPH Block device**”-a. Prema tome druga dva modela (“CEPH Object Storage” i “CEPH Filesystem”) više nećemo spominjati.

### **Potrebne funkcionalnosti (CEPH Roles) za RBD**

Kao što smo rekli za svaki od CEPH modela, potrebne su određene funkcionalnosti na strani CEPH poslužitelja u CEPH klasteru.

Za upotrebu CEPH-a kao “Block device”-a tj. kao RBD-a, potrebne su nam dvije funkcionalnosti odnosno “uloge” poslužitelja. To prema definiciji znači da moramo imati poslužitelje od kojih je svaki zadužen samo i isključivo za jednu ulogu:

- uloga Monitor poslužitelja (Engl. Monitor Node)
- uloga OSD poslužitelja (ovo su poslužitelji na kojima se nalaze tvrdi diskove koje ćemo koristiti u CEPH klasteru).

Preporuka za najosnovniju upotrebu kao CEPH RBD, bi bila:

- minimalno 3 poslužitelja s ulogom “Monitor”
- minimalno 3 poslužitelja s ulogom “OSD”

Mi ćemo, s obzirom da imamo samo tri poslužitelja s diskovima (koje želimo koristiti kao CEPH klaster za “Block device”) te stoga što što ne tražimo ekstra/turbo brz/proširiv/... sustav, napraviti slijedeće.

Uloge poslužitelja:

- Poslužitelj 1 : **OSD** i **MON**itor
- Poslužitelj 2 : **OSD** i **MON**itor
- Poslužitelj 3 : **OSD** i **MON**itor

Dakle svaki poslužitelj će imati i OSD i MONitor ulogu. S ovime smo na malo zaobilazan način osigurali da imamo i tri OSD-a i tri MONitora.

### **Zbog čega minimalno tri (3) poslužitelja za klaster**

Većina klastera u radu rade na principu ”[Quoruma](#)“ dakle tri je najmanji broj poslužitelja u kojemu minimalna većina (dva) poslužitelja sudjeluju u dogovaranju i provjerama rada.

Ovdje se radi o sustavu “glasovanja” i izbora što znači da svaki poslužitelj ima jedan glas za glasovanje. Ako su samo dva poslužitelja u sustavu glasovanja izbori su nemogući. Prema tome za sustav glasovanja je potrebno minimalno troje.

## Quorum pojednostavljeno

U ovakvim minimalnim klasterima s tri poslužitelja, u svakom trenutku moraju biti aktivna i funkcionalna dva (2) poslužitelja. Ovo ne mora čak značiti da je jedan poslužitelj ugašen već možda ne radi kako treba, pa daje pr. krive rezultate (ili ih ne daje uopće) tada se ta zadnja dva pokušavaju sustavom “glasovanja” dogovoriti. Ovakav sustav “Quoruma” se koristi i kod klasterskih sustava za virtualizaciju pr. [Proxmox VE cluster](#).

Zamislimo tri poslužitelja koja imaju “Cluster Map” tablicu s pripadajućom verzijom tablice i njen [hash/checksum](#) koji govori o tome da li je integritet tablice narušen.

## Primjer

Prva dva poslužitelja kažu da im je zadnja verzija v.234 te HASH : A348F3609D a treći poslužitelj tvrdi da je njegova zadnja verzija v.252 te HASH : 35D56FAB5D. Dogoditi će se to da će prva dva nadglasti treći iako ima veći broj verzije (što bi značilo da je novija) te se on IZBACUJE iz klastera te se više ne uzima u obzir koje slijedeće provjere (sve dok i on ne bude imao sve iste “podatke” kao i preostala dva). Obično kod ovakvih sustava postoje Tzv. “Izbori” za klaster “Mastera”, a koji se događaju svakih nekoliko sekundi (pr. svakih 15. sekundi). Dakle u jedinici vremena unutar koje se događaju izbori (ili reizbori) za “Mastera” tj. “Primarnog” poslužitelja, svaki poslužitelj ima određeni prioritet: primjerice:

- Prvi poslužitelj - prioritet 1
- Drugi poslužitelj - prioritet 2
- Treći poslužitelj - prioritet 3

Ako se recimo onaj s najmanjim brojem prioriteta bira za “Master”-a (tj. “Primarnog”) , tada će “Prvi poslužitelj” postati “Master” ako je sve u redu s njegovim verzijama i integritetom. Ako nije tada će “Master” postati onaj s prioritetom 2 tj. “Drugi poslužitelj” itd. Dakle svakih recimo 15. sekundi se odabire novi “MAster”. “Master” je obično zadužen za vrlo važne operacije odlučivanja - koji će poslužitelj biti izbačen iz klastera te će on to i fizički napraviti (obično zapisati u datoteku u kojoj je lista aktivnih poslužitelja u klasteru). Ova funkcionalnost je ne zahtjevna prema resursima ali kao što je vidljivo, vrlo važna. “Master” osim toga radi još nekoliko resursno ne zahtjevnih zadataka - ovisno o vrsti i tipu klastera.

Ovo znači da ako primjerice restartamo cijeli klaster (recimo zbog nadogradnji sustava), da to radimo oprezno. Prvo jedan poslužitelj, pa kada je on potpuno funkcionalan nakon restarta, drugi, pa kada je drugi nakon restarta funkcionaln, tek onda treći.

## MONitor uloga u CEPH clusteru

MONitor uloga mora biti instalirana na minimalno tri poslužitelja. Ona se brine o:

- tome koji poslužitelji u CEPH klasteru su živi OSD poslužitelji i koji su sve dostupni diskovi (OSD-ovi).
- Pohranjuje i održava 5 “tablica/konfiguracija”:
  - **Monitor map** - tablica s MONitor poslužiteljima
  - **OSD map** - tablica s OSD poslužiteljima/diskovima
  - **PG map** - tablica s PG (Placement Group)- grupama za pohranu objekata
  - **CRUSH map** - “CRUSH” hijerarhijska tablica/konfiguracija
  - **MDS map** (za MDS ulogu [koristi se samo za **S3** ili **Swift** tj. za upotrebu kao “Object Storage”])



**OSD** = Object Storage Daemon. Servis (daemon) je to zadužen za rad s objektima i njihovu distribuciju te u konačnici snimanje na tvrdi disk. Jedan OSD daemon (servis) je zadužen za jedan tvrdi disk.

Dakle OSD poslužitelj koji ima osam (8) tvrdih diskova, ima i pokrenuto osam (8) OSD daemon (servisa).

### OSD uloga u CEPH clusteru

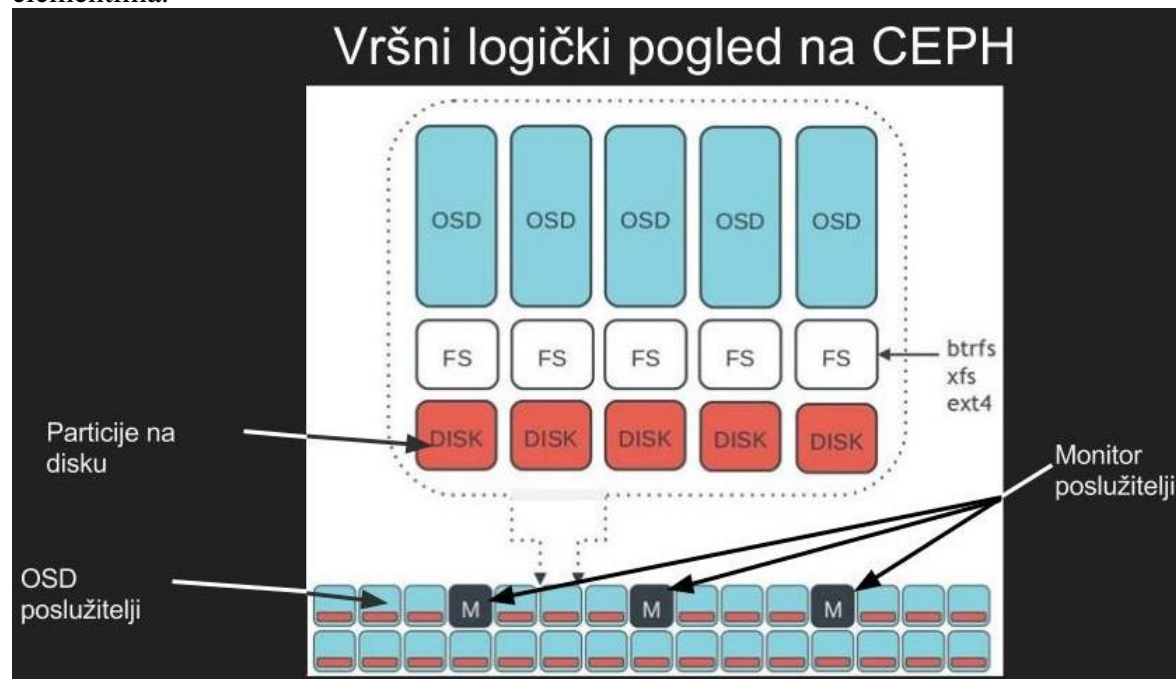
Ovu ulogu moraju imati minimalno tri (3) poslužitelja.

OSD uloga je zadužena za:

- Spremanje objekata na lokalni datotečni sustav (u konačnici na “OSD” tvrtde diskove ) i omogućavanje pristupa objektima preko mreže
- zadužena je za replikaciju objekata koji se zapisuju na konkretni OSD (Daemon/servis) odnosno tvrdi disk. Dakle radi replikaciju objekata koji završe zapisani na OSD (Tvrdi disk) prema drugom OSD (tvrdi disk) - ovisno o “Cluster Map”-i i drugim parametrima (tj. o “**Pool**”-u ili ekvivalentu RAID polja koje se rsprostire na poslužitelje i diskove u CEPH klasteru).
- korištenje [journaling](#) mehanizama kod zapisivanja podataka na OSD (disk) prema transakcijskom modelu.

### Pogled na CEPH

Pogledajmo kako logički izgleda cijeli CEPH, sada kada smo se upoznali sa svim važnijim elementima.



U gornjem dijelu slike je vidljiv izgled jednog OSD poslužitelja s pet tvrdih diskova.

Svaki tvrdi disk mora imati minimalno jednu particiju, koju možemo formatirati s nekim od predloženih datotečnih sustava:

- xfs (preporuka)
- ext4 ili
- btrfs

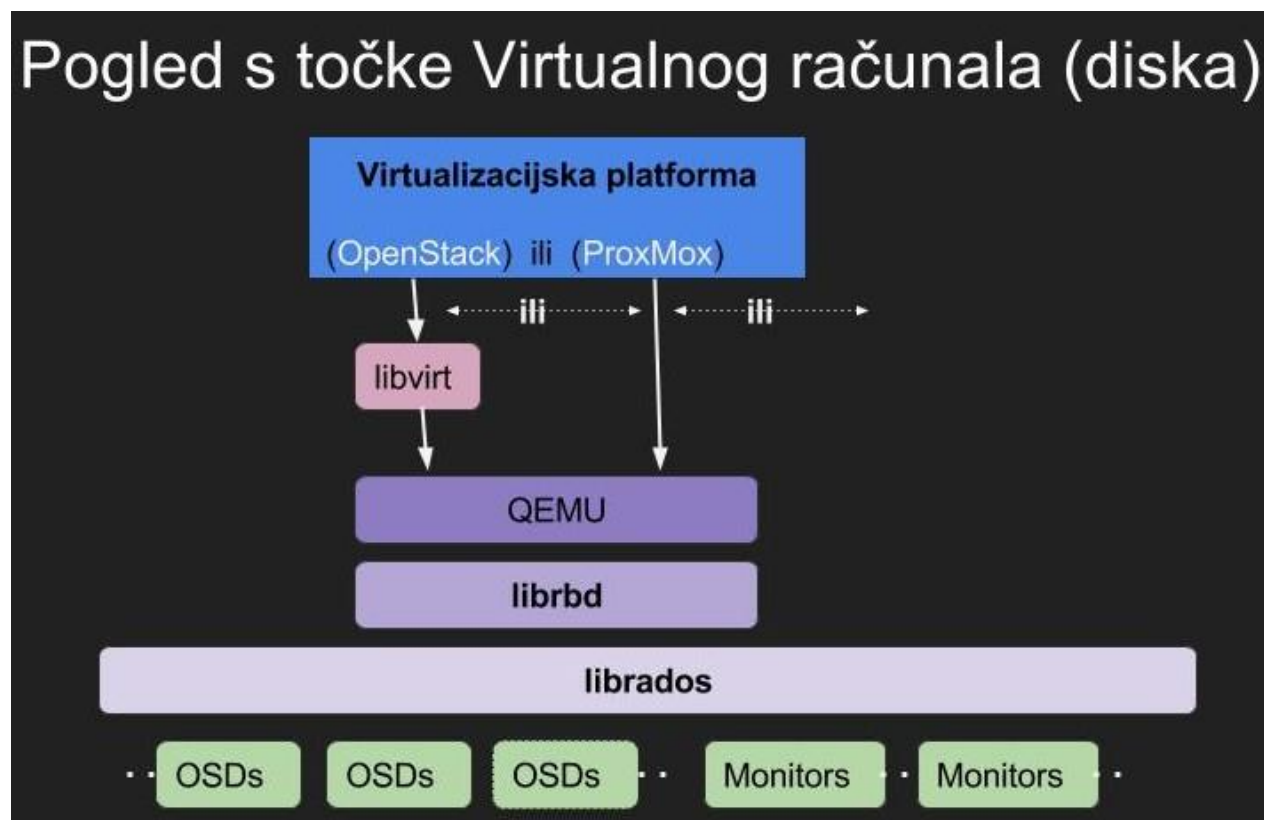
Dodatno, potrebna nam je još jedna particija (ili zaseban disk ili polje diskova s dodatnom particijom za “Journaling”)

U konačnici, na postojeću particiju koja je namijenjena za CEPH, na datotečni sustav kreira se struktura direktorija u koju se spremaju CEPH objekti kao i njihovi pripadajući metapodaci.

**U donjem dijelu slike** je vidljiva pozicija svakog pojedinog OSD poslužitelja (s svim njegovim “OSD” diskovima) te pozicije svih MONitor poslužitelja. Dakle vidljiv je CEPH sustav sa ukupno 30 poslužitelja i to :

- tri CEPH MONitor poslužitelja i
- 27 CEPH OSD poslužitelja.

Sada zamislimo upotrebu u kojoj imamo poslužitelje za virtualizaciju, koji koriste ovakav CEPH sustav (sa svih 30 poslužitelja) kao disk storage sustav, dakle za spremanje virtualnih diskova virtualki. Pogledajmo sliku kako to izgleda sa strane Virtualnog računala odnosno platforme za virtualizaciju prema CEPH sustavu (od gore do dolje)



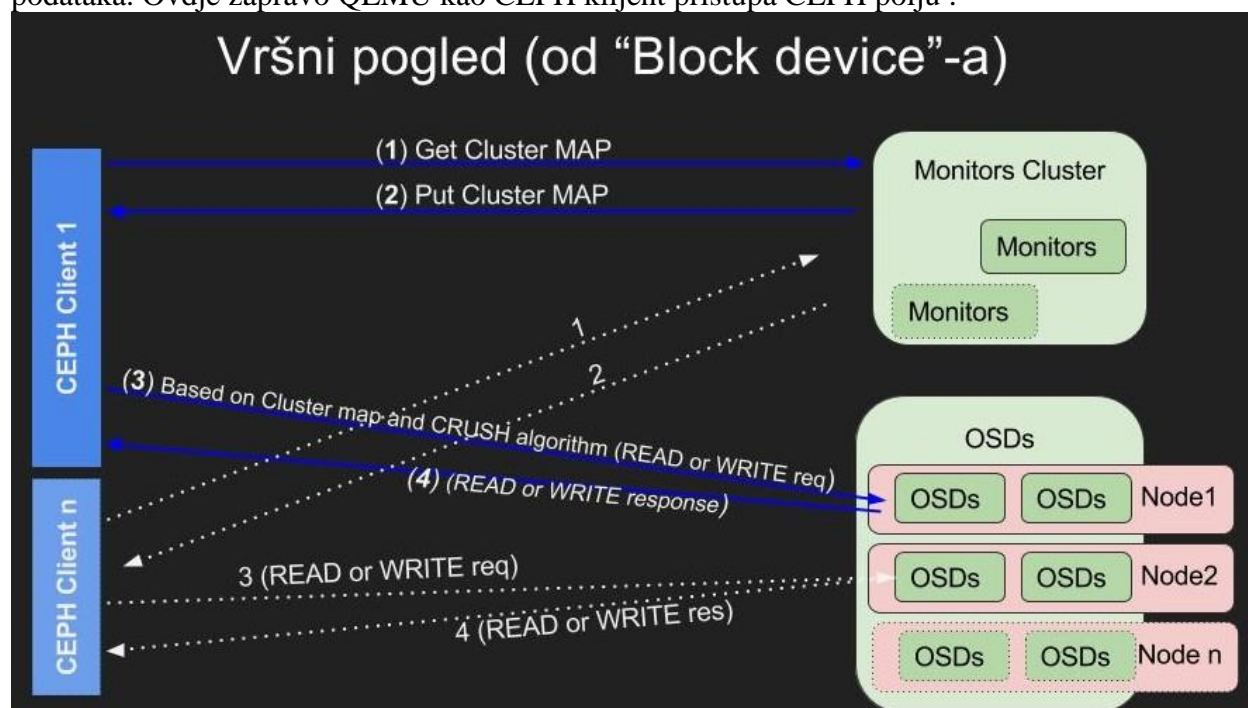
Ovdje je vidljiv način pristupa CEPH “Block device”-u tj. logičkom “blok” uređaju odnosno disku koji predstavlja cijeli CEPH cluster. Na primjeru su dvije česte platforme za virtualizaciju:

- OpenStack i
- Proxmox VE



Platforma za virtualizaciju za svako virtualno računalo koje koristi virtualni tvrdi disk (koji je zapravo “blok uređaj” tj. logički tvrdi disk od cijelog CEPH klastera ), koristi QEMU (i Linux KVM). QEMU i Linux KVM su zaduženi za sve potrebne funkcionalnosti da bi se virtualizacija uopće mogla koristiti. Dakle oni simuliraju sve virtualne komponente svakog pojedinog virtualnog računala (Matična ploča i njen BIOS, CPU, mrežna kartica i njen BIOS, disk kontroler i njen BIOS te pripadajući virtualni tvrdi disk, ...). Qemu kao Hipervizor ima nadalje metodu za korištenje svakog pojedinog virtualnog diska koji se zapravo nalazi unutar CEPH klastera ( kao “Block device” ). QEMU se tada spaja kao klijent na CEPH klaster i to na točno određeni **CEPH Pool** te njega koristi kao da je “polje diskova” na nekom SAN sustavu (jer govorimo o upotrebi CEPH-a kao “Block device-a” tj. kao RBD)

A sada pogledajmo kako to izgleda sa strane “CEPH Block Device”-a odnosno blok uređaja, kao krajnje komponente, koja na kraju stvarno pristupa CEPH klasteru za čitanje ili zapisivanje podataka. Ovdje zapravo QEMU kao CEPH klijent pristupa CEPH polju :



### Klijent 1 piše ili čita na ili sa CEPH RBD

1. Kod procesa čitanja ili pisanja na “Block device” tj. CEPH RBD ,klijent koji žali nešto zapisati ili pročitati iz CEPH clustera koji koristi kao blok uređaj (logički kao tvrdi disk), prvo kontaktira CEPH klaster i to MONitor poslužitelje i od njih traži “CLuster Map” tablicu/konfiguraciju.
2. CEPH cluster MONitor poslužitelj(i) mu šalju traženu tablicu/konfiguraciju
3. Na osnovi tablice/konfiguracije koju je dobio, klijent pomoću CRUSH algoritma traži od OSD poslužitelja i OSD diskova podatke za čitanje ili traži pisanje. Do točnih OSD poslužitelja i točno određenih OSD diskova je pomoću CRUSH algoritma izračunao koji su te od njih i traži/šalje podatke
4. S OSD-ova dobiva odgovor na traženi zahtjev (čitanje ili pisanje)

### Klijent 2 piše ili čita na ili sa CEPH RBD

Ponavlja se proces kao i za prvog klijenta

## Licenca

Ovaj dokument je pisan pod [GNU GPL licencom](#).

**U nastavku, cijeli dokument je pisan na engleskom jeziku, jer sam u vrijeme dizajna, testiranja, implementacije i kasnijih optimizacija sve i pisao na engleskom (kada uhvatim vremena krećem i polako s prijevodom).**

**Uvodni dio “Consideratoins” se sastoji od preporuka izvađenih iz više izvora i predstavlja samo okvire za moj početak razmišljanja o zahtjevima koje pred nas postavlja CEPH.**

**Dakle prenesena su mišljenja drugih koja sam uzeo u obzir te ih kao takve i prenosim.**

**Nastavak uputa je zapravo opis mog osobnog iskustva u implementaciji CEPH-a na Proxmox VE platformu. U toku procesa dokumentiranja, pisao sam upute više manje za sebe (da mogu ponoviti instalaciju i negdje drugdje) ali želja je bila i da se isti tekst podjeli s drugima koji su ušli u ovo područje, u nadi da će im skratiti vrijeme učenja.**

**Dakle promatrajte ovaj dokument kao moj blog o CEPH-u odnosno kao radnu verziju dokumenta koji vam može pomoći pri vašoj instalaciji istog.**

**Napomena :**

**Primjeri u daljem tekstu se odnose na implementaciju CEPH-a na Proxmox VE platformi za virtualizaciju **v.3.4**, ali ih je lako primijeniti i bilo gdje drugdje.**

Autor: Hrvoje Horvat

[LinkedIN](#)

# CEPH Storage with integration in Proxmox VE

In the text that follows i will try to reproduce my steps and experience of installation of CEPH inside Proxmox VE platform. In my limited time of work od this project many topics are leave open, so use this documet as initial guiding or installation manual under construction (possible with flaws).

Everyone are welcome to comment or suggest some changes or point at some errors that must be changed. Look at this as my personal internal document that i just want to share with the people who are trying to do the same thing (try to use CEPH as shared, distributed Block device storage (RBD))

## License

This document is released under [GNU GPL license](#).

**Author:** Hrvoje Horvat

[LinkedIN](#)

## Introduction

Ceph is a distributed object store and file system designed to provide excellent performance, reliability and scalability - See more at: <http://ceph.com/>. Proxmox VE supports Ceph's RADOS Block Device (RBD) to be used for VM disks. The Ceph storage services are usually hosted on external, dedicated storage nodes. Such storage clusters can sum up to several hundreds of nodes, providing petabytes of storage capacity. For smaller deployments, it is also possible to run Ceph services directly on your Proxmox VE nodes. Recent hardware has plenty of CPU power and RAM, so running storage services and VMs on same node is possible. This articles describes how to setup and run Ceph storage services directly on Proxmox VE nodes.

## Advantages

Major advantages of CEPH storage's are :

- Easy setup and management with CLI and GUI support on [Proxmox VE](#)
- Thin provisioning
- Snapshots support
- Self healing
- No single point of failure
- Scalable to the exabyte level
- Setup pools with different performance and redundancy characteristics
- Data is replicated, making it fault tolerant
- Runs on economical commodity hardware
- No need for hardware RAID controllers (not needed at ALL)
- Easy management
- Open source

What we shall do:

We will prepare ALL nodes for CEPH storage, on the same server in which we have installed [Proxmox VE](#) virtualization platform. We will install and use CEPH on local nodes inside separate **DISKS NOT IN RAID**. We have also special network cards Bonded for CEPH only (in a separate VLAN).

## What we need ?

What we'll be nice to have ?

- Working ProxMox VE Cluster with minimum 3 nodes (i was using Proxmox VE v.3.4.x):
- RAID 1 : 2 system disks (for ProxMox)
- JBOD Disks : 8 or more used as Storage for CEPH, on every node
- Network :
  - Minimum:
  - 2 or more x 1Gbps in LACP/Bond with Switch config
    - If possible :
  - 2 x 1Gbps (LACP) for CEPH Frontend LAN
  - 2 x 1Gbps (LACP) for CEPH Backend LAN
- Use it in separate LAN/ VLAN just for CEPH

⇒ General commendation:

- Do not use LACP+VLAN (bond + VLAN) it will slow things (not so much in my case - if using openvSwitch at least 2.3.2+).
- Recommendation is 10Gbps network cards with 10Gbps Switch

(minimum Netgear XS712T : 12 x 10Gbps ([tested on Proxmox VE article](#) ) ):

- 1 x 10Gbps for CEPH Frontend
- 1 x 10Gbps for CEPH Backend

More (1 or more) 1Gbps network cards for VM/Linux Container Hosts also preferred 2 x 1Gbps in LACP/Bond + VLANs with Switch

- Minimum 3 Nodes for CEPH (it can be the same nodes as ProxMox):
- Minimum 3 Nodes as Monitor CEPH Role (it can be the same nodes as ProxMox)
- Minimum 3 Nodes as OSD CEPH Role (Object Storage Devices) (it can be the same nodes as ProxMox)

[nodes/storages with disks/storages])

## What we will use in this test

Minimum environment shall be 3 Servers - All of them with all of 3 roles installed.

### 3 x Dell PowerEdge R720:

- 128GB RAM
- 32 Core CPU (2 x Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz)
  - RAID Controller with BBU : LSI Logic / Symbios Logic MegaRAID SAS 2208
- 2 x SAS 10k RPM ( SEAGATE ST9146853SS ) RAID 1 (Mirror) for System (Proxmox VE)
- 8 x 900GB SAS 10k RPM ( TOSHIBA AL13SEB900 ) in JBOD (as single disk drives for CEPH)
- 4 x 1Gbps LAN

**This entire article is based on using CEPH as block device (using RBD) inside Proxmox VE Proxmox VE and CEPH side**

- Proxmox VE v.3.4. using kernel 2.6.32-x-PVE (Proxmox VE patched kernel 2.6.32.x, with OpenVZ “stuff”)
- Ceph code name “HAMMER” : v.0.94

## **What we should be aware of using Ceph RBD (as Block Device) inside Proxmox VE**

Using current version of CEPH as Block device (RBD) mounted inside ProxMox Cluster we must be aware of :

- It can be used for storing Virtual Machines disk only :
- In one Word CEPH RBD Storage can be used only for storing ProxMox “Disk Image” Content
- Virtual Machine hard disk format can be ONLY RAW (it can not be changed)
- Virtual Machine Snapshot is working without problem (it is using CEPH snapshot in the background)
- Can not be used for storing ISO images for VMs
- Can not be used for storing Linux Containers at all :
- ⇒ in current CEPH + ProxMox combination version : Proxmox VE v.3.x (v.4.x. fix this and using LXC containers instead of OpenVZ Linux Containers)
- Can not be used for storing Linux Container Templates (in current CEPH version)
- Can not be used for storing Backup of Linux Containers or Virtual Machines (in current CEPH version)

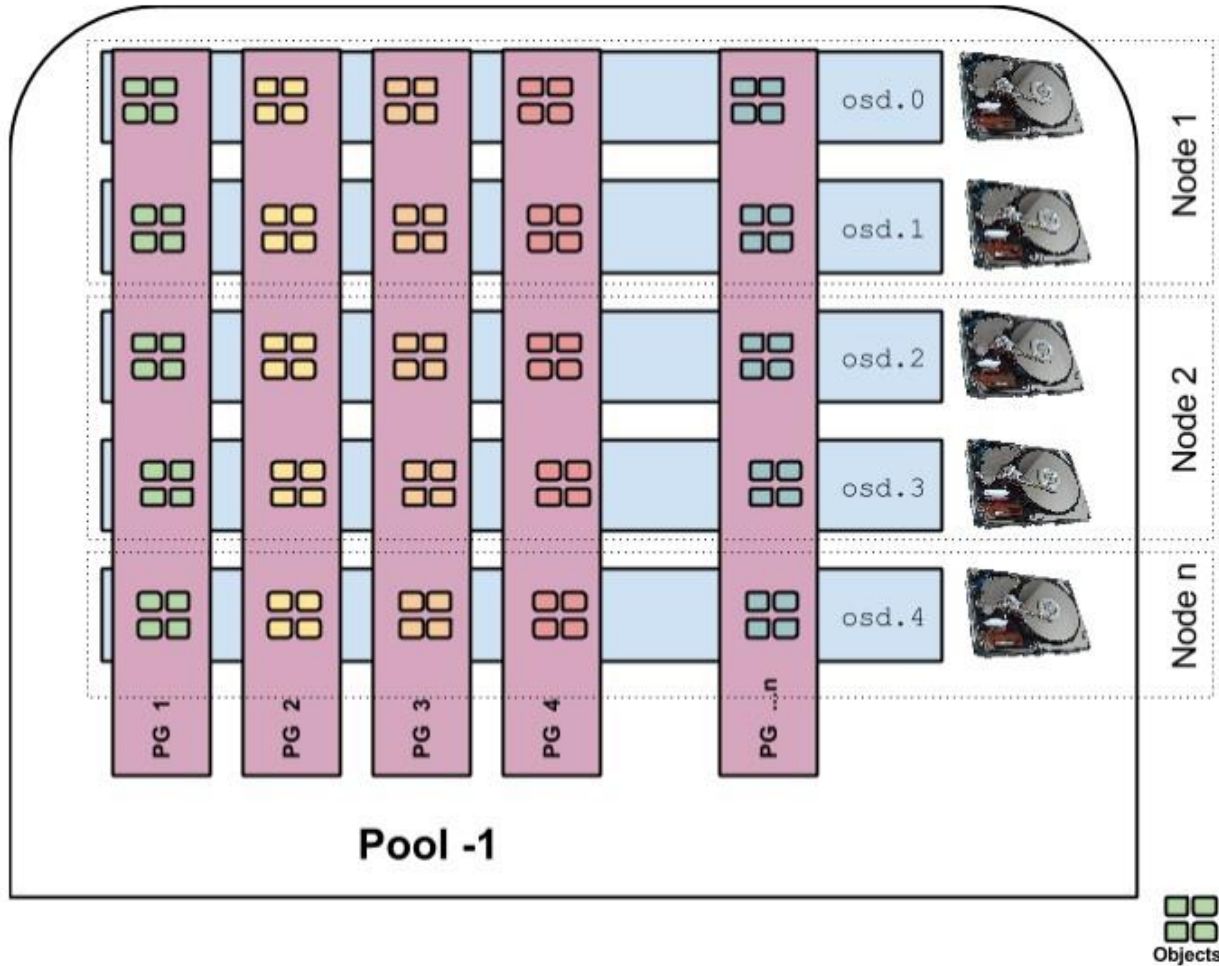
## **CEPH Logical scheme**

Ceph stores, replicates and rebalances data objects across a RADOS cluster dynamically. With many different users storing objects in different pools for different purposes on countless OSDs, Ceph operations require some data placement planning. The main data placement planning concepts in Ceph include:

- **Pools:** Ceph stores data within pools, which are logical groups for storing objects. Pools manage the number of placement groups, the number of replicas, and the ruleset for the pool. To store data in a pool, you must have an authenticated user with permissions for the pool. Ceph can snapshot pools. See Pools for additional details.
- **Placement Groups:** Ceph maps objects to placement groups (PGs). Placement groups (PGs) are shards or fragments of a logical object pool that place objects as a group into OSDs. Placement groups reduce the amount of per-object metadata when Ceph stores the data in OSDs. A larger number of placement groups (e.g., 100 per OSD) leads to better balancing. See Placement Groups for additional details.
- **OSDs :** This is a real physical disk drives
- **CRUSH Maps:** CRUSH is a big part of what allows Ceph to scale without performance bottlenecks, without limitations to scalability, and without a single point of failure. CRUSH maps provide the physical topology of the cluster to the CRUSH algorithm to determine where the data for an object and its replicas should be stored, and how to do so

across failure domains for added data safety among other things. See CRUSH Maps for additional details.

Look at the picture:

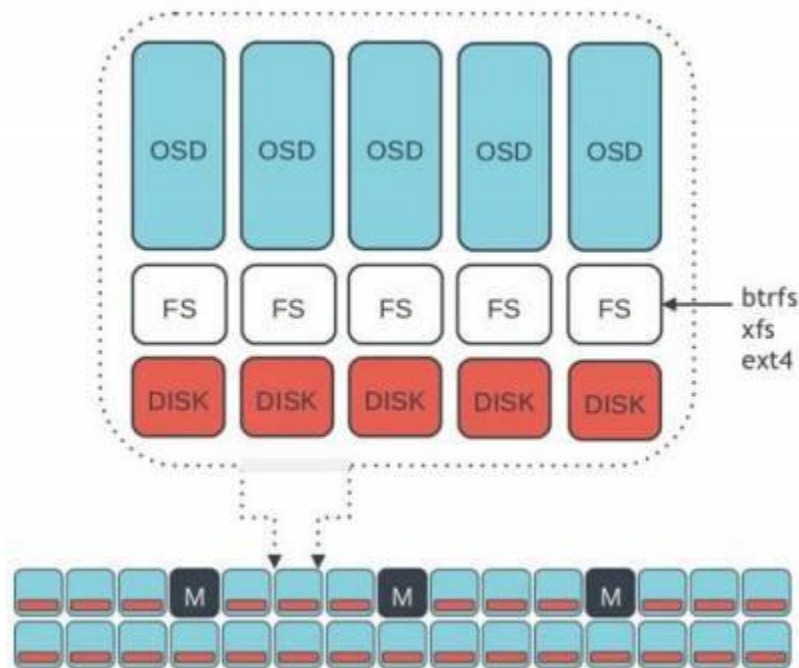


**OSD** = Actual Disk drive

**PG** = Placement Groups

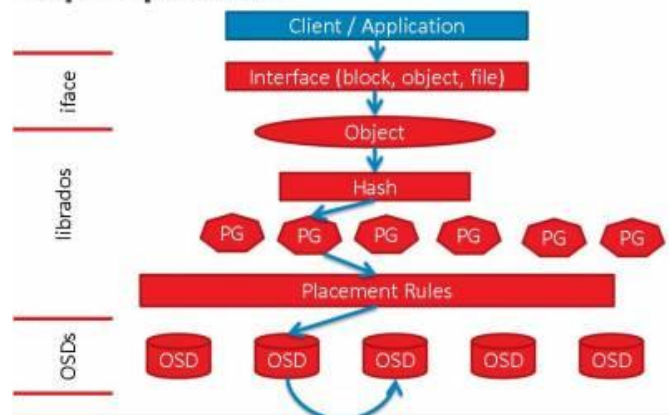
**Objects** = Smallest block that can store data

This picture represent Physical to logical mapping:



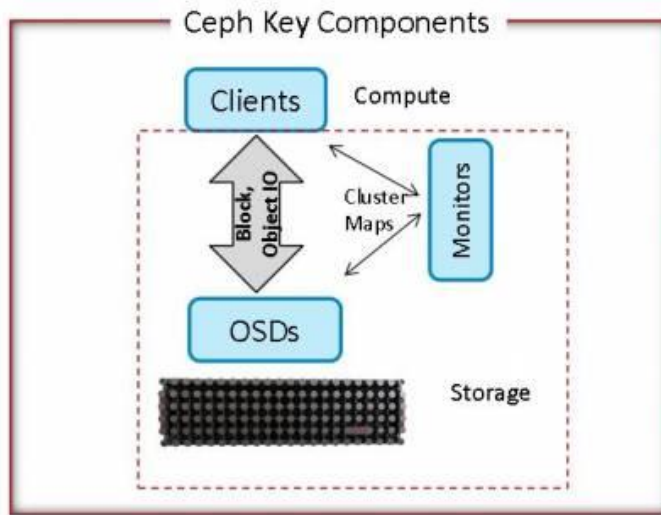
CEPH Operation is visible in picture:

### Ceph Operation





CEPH Key components:



Further Infos :

[1 Working Manual](#) [Intro](#) [More tips](#) [Profiling tools](#)

## Planning

Plan maximum number of VMs supported, plan :

- Plan IOPS Per VM - Average
- Plan IOPS Per VM - Peak

**Planing consideration that follows is extracted from few CEPH presentation, CEPH Conventions, advice's and proposals and not tested from my side. It is just used as it is as a starting point for my consideration.**

## Considerations

### Disks considerations

Given the fairly randomly distributed I/O load for object data best case average performance from spinning media is about 90 – 100 MB/sec. Real world object gateway performance is more in the 60 – 70 MB/sec average range per disk. This is also impacted by object gateways not providing a particularly deep I/O queue in observed tests. Peak disk performance can be higher, which is why a 4:1 SSD journal ratio is recommended.

- 8 - 10 SAS HDD per 1x 10Gbps
- ~12 SATA HDD per 1x 10Gbps
- 1 x SSD for Write journal per 4-6 OSDs (HDD)

### RAM planning

- 1-2 GB RAM per 1 TB per one OSD storage space

### CPU Planning

- 0.5 CPU core at 1GHz per OSD disk (in some cases (Source: Dinko Korunic) up to 1 CPU Core at 2GHz (when using Ceph FS))
- 1-2 CPU cores per SSD drive



CEPH Replica count :

- More replica slower speed
- Consider 2 replica minimum, minimum
- Replica count means multiple media writes for each object PUT

## Journal

Peak write performance of spinning media without separate journals is around half due to writes to journal and data partitions going to the same device.

A RAID1 of SSDs is not recommended. Wear leveling makes it likely SSDs will be upgraded at similar times. The doubling of SSDs per node also reduces storage density and increases price per gig. With massive storage scale,

it's better to expect drive failure and plan so failure is easily recoverable and tolerable.

[How to test right SSD for CEPH Journaling](#)

⇒ **MUST READ BEFORE BUY SSD FOR YOUR JOURNAL**

## CEPH Monitor nodes

1 CEPH monitor node per 15-20 OSD nodes (min 3 per cluster)

## Workload planning

70% / 30% read/write IOPS 4-8KB random read pattern

## CEPH + XFS Statistical Efficiency

4-8K Random Read 88% (0.88) 4-8K Random Write 64% (0.64)

If we have 100 IOPS on our Disks -CEPH will effectively use

- 88 IOPS for read
- 64 IOPS for Write.

## Sizing

**Sizing Examples from the practice (for 1000 TB of space):**

- 1PB (1.000 TB)
- 500 Virtual Machines
- 100 IOPS per VM (General storage )

= 50 000 IOPS in summary from storage cluster

## MON nodes

- CPU : 2620v3 (6 core)
- RAM : 64 GB
- HDD : 1 x 1 TB SATA
- NIC: 1 x 1Gbps , 1 x 10 Gbps

1 Node per 15-20 OSD nodes

**NOTE : This planning is for datacenter usage - we will use smaller footprint**

## OSD Server nodes

### Performance optimized

- CPU : 2620v3 (6 cores)
- RAM : 64 GB
- HDD (for OS) : 2 x SATA 500 GB : RAID 1-2
- HDD (for OSD) : 20 x SAS 1.8TB 10000 rpm (HGST C10K1800)
- SSD (write journals) : 4 x SSD 128GB Intel Server SSD : S3700
- NIC: 1 x 1Gbps , 4 x 10Gbps (2 bonds : 1 for CEPH public, 1 for CEPH replication)

How many server we need for 1000 TB (1PB) ?

1000 TB / (20 disks per server x 1.8TB disks x 0.85 (it is 85% capacity max because of XFS in the background) ) x 3 (replicas )

= 98 Servers

## Basic Calculation

$1000 / (20 \times 1.8 \times 0.85) \times 3 = 98$  Servers

Edit

### IOPS Calculations

#### READ Calculation

If one of this SAS 10k HDD Hard drive has 250 IOPS We have it 20 per server, and we have 98 servers and 88% (0.88) is CEPH read usage then :

$250 \times 20 \times 98 \times 0.88 = 431\ 200$  IOPS This is 800 IOPS per VM

#### WRITE Calculation

---

If one of this SAS 10k HDD Hard drive has 250 IOPS We have it 20 per server, and we have 98 servers and 64% (0.64) is CEPH write usage then :

$250 \times 20 \times 98 \times 0.64 = 313\ 600$  IOPS This is 600 IOPS per VM

### Higher density , Cost optimized

- Server Super Micro 6048R
- CPU : 2 x 2630v3 (16 cores)
- RAM : 192 GB
- HDD (for OS) : 2 x SATA 500 GB : RAID 1-2
- HDD (for OSD) : 28 x SATA 4TB 7200 rpm (HGST 7kK4000)
- SSD (write journals) : 4 - 6 x SSD 128GB Intel Server SSD : S3700
- NIC: 1 x 1Gbps , 4 x 10Gbps (2 bonds : 1 for CEPH public, 1 for CEPH replication)

How many server we need for 1000 TB (1PB) ?

1000 TB / (28 disks per server x 4TB disks x 0.85 (it is 85% capacity max because of XFS in the background) ) x 3 (replicas ) = 98

Basic Calculation  $1000 / (28 \times 4 \times 0.85) \times 3 = 32$  Servers

= 32 Servers

## IOPS Calculation

One Disk Drive IOPS = 150

## READ Calculation

If one of this SATA 7200 RPM HDD Hard drive has 150 IOPS We have it 28 per server, and we have 32 servers and 88% (0.88) is CEPH read usage then :

$150 \times 28 \times 32 \times 0,88 = 118\,272$  IOPS This is 200 IOPS per VM

## WRITE Calculation

If one of this SATA 7200 RPM HDD Hard drive has 150 IOPS We have it 28 per server, and we have 32 servers and 64% (0.64) is CEPH write usage then :

$150 \times 28 \times 32 \times 0,64 = 86\,016$  IOPS This is 150 IOPS per VM

## CEPH OSD Limitations

IOPS per OSD - in CEPH v.0.94.x+ : ~ 30.000+ IOPS (Not a real limit anymore)

## CEPH Recovery time

In the case of a failure (of Entire CEPH node or similar entity) if CEPH need to fully recover **TB** of data - how it will long, depending on Network Speed:

### On 1Gbps LAN

Data size in TB	Time i hours
5 TB Data	11.11 hours
10 TB Data	22.22 hours
20 TB Data	44.4 hours
200 TB data	444.4 hours

### On 10Gbps LAN

Data size in TB	Time i hours
20 TB Data	4.4 hours
200 TB data	44.4 hours

### On 40Gbps LAN

Data size in TB	Time i hours
20 TB Data	1.1. hours
200 TB data	11.1 hours

### On 100Gbps LAN

Data size in TB	Time i hours
20 TB Data	0.4. hours
200 TB data	4.4 hours

**Recovery Time in seconds = disk capacity in Gigabits / ( network speed \*(nodes-1) )**

On 4 nodes ,every with 8 x 1TB HDD= 32 TB of storage and 1Gbps LAN : Replication of 32TB storage will take approx 24.hours on 1Gbps LAN. On 10Gbps LAN this time is reduced to 2.h. Calculated Recovery time as show in picture (as example):

### Calculated Recovery time

CEPH Calculations			
Nr. Of Ceph nodes	4		
Nr. Of disk drives	32		
Each disk capacity (TB)	1 TB		
Sum of Capacity (TB)	32 TB		
Number of replica	2		
Usable Capacity (GB)	16 TB		
Capacity in Gigabit	262.144		
Network speed ( Gbps )	1 Gbps	10 Gbps	
Recovery time (sec)	87.381	8.738	
Recovery time (min)	1.456	146	
<b>Full Recovery time (hours)</b>	<b>24 h</b>	<b>2 h</b>	

## Latency

Latency on disk operations, consideration in the practice.

CEPH Latency (Delay) 1 - 15ms (depending on hardware : can be up to 10 - 40 ms)?

Look the Latency in Proxmox VE Virtualization Cluster (using CEPH):

PROXMOX Proxmox Virtual Environment Version: 3.4-9/4b51d87a									
Server View Node * 224x									
Search Summary Services Network DNS Time Syslog Bootlog Task History UBC Subscription Firewall Updates Ceph									
Reload Start Stop Out In Remove									
Name	Type	Status	weight	reweight	Used		Latency (ms)		
					%	Total	Apply	Commit	
226x	host								
osd.15	osd	up/in	0.809998	1	0.28	832.34GB	1	0	
osd.14	osd	up/in	0.809998	1	0.30	832.34GB	1	0	
osd.13	osd	up/in	0.809998	1	0.29	832.34GB	7	1	
osd.12	osd	up/in	0.809998	1	0.31	832.34GB	12	2	
osd.11	osd	up/in	0.809998	1	0.29	832.34GB	2	1	
osd.10	osd	up/in	0.809998	1	0.27	832.34GB	1	0	
osd.9	osd	up/in	0.809998	1	0.30	832.34GB	1	0	
osd.8	osd	up/in	0.809998	1	0.27	832.34GB	1	0	
224x	host								
osd.7	osd	up/in	0.809998	1	0.26	832.34GB	2	1	
osd.6	osd	up/in	0.809998	1	0.29	832.34GB	1	0	
osd.5	osd	up/in	0.809998	1	0.31	832.34GB	1	0	
osd.4	osd	up/in	0.809998	1	0.30	832.34GB	1	0	
osd.3	osd	up/in	0.809998	1	0.29	832.34GB	1	1	
osd.2	osd	up/in	0.809998	1	0.26	832.34GB	1	0	
osd.1	osd	up/in	0.809998	1	0.30	832.34GB	1	0	
osd.0	osd	up/in	0.809998	1	0.31	832.34GB	11	1	

Select Proxmox VE Node in CEPH Cluster ⇒ **CEPH** → **OSD**

⇒ Look at the Latency (in ms): in this case (2 nodes 2 replica , 1Gbps LAN) it varies : 1.ms - 12.ms.

# CEPH installation inside Proxmox VE (on Proxmox VE nodes)

## Initial steps

### Prerequisites

1. Installation of XFS filesystem support in ProxMox VE (sone we will use XFS)

```
apt-get update
apt-get install xfsprogs
```

2. Install lsb package:

```
apt-get install lsb-release
```

Do it on all CEPH nodes

Create user :

```
useradd -d /home/cephadmin -m cephadmin
passwd cephadmin
(type new password)
```

Assign sudo permissions to this user Add :

```
cephadmin ALL = (root) NOPASSWD:ALL
```

to file : /etc/sudoers.d/cephadmin

```
chmod 0440 /etc/sudoers.d/cephadmin
```

Get keys:

```
wget -q -O- --no-check-certificate
'https://ceph.com/git/?p=ceph.git;a=blob_plain;f=keys/release.asc' | apt-key
add -
```

Or go to Browser and open this file , copy the content to file /root/release.asc and paste the content to it. Now import the key:

```
apt-key add release.asc
```

Check the latest CEPH “codename” :

<http://ceph.com/docs/dumpling/install/debian/>

(in our case HAMMER is latest stable for production)

And add this content to the file

For Proxmox VE 3.x

```
deb http://ceph.com/debian-hammer/ wheezy main
```

Or for automatic addition to this file just type:

```
echo deb http://ceph.com/debian-hammer/ $(lsb_release -sc) main | tee
/etc/apt/sources.list.d/ceph.list
```

Update

```
apt-get update
```

Install ceph tools:

```
apt-get install ceph-deploy
```

Copy the auth key

```
cp /root/release.asc /usr/share/pyshared/ceph_deploy/hosts/debian/
cd /usr/share/pyshared/ceph_deploy/hosts/debian/
```

## CEPH Cluster creation

We will create CEPH Cluetr on Proxmox VE node name **226x**

For the First CEPH SERVER ONLY and create the CEPH Cluster Add the first server (using hostname resolved in network for CEPH)

```
ceph-deploy new 226x
```

```
[ceph_deploy.conf][DEBUG ] found configuration file at:
/root/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (1.5.22): /usr/bin/ceph-deploy new 226x
[ceph_deploy.new][DEBUG ] Creating new cluster named ceph
[ceph_deploy.new][INFO  ] making sure passwordless SSH succeeds
[osir226x][DEBUG ] connected to host: 226x
[osir226x][DEBUG ] detect platform information from remote host
[osir226x][DEBUG ] detect machine type
[osir226x][DEBUG ] find the location of an executable
[osir226x][INFO  ] Running command: /bin/ip link show
[osir226x][INFO  ] Running command: /bin/ip addr show
[osir226x][DEBUG ] IP addresses found: ['192.168.200.226',
'192.168.222.226', '192.168.254.226']
[ceph_deploy.new][DEBUG ] Resolving host 226x
[ceph_deploy.new][DEBUG ] Monitor 226x at 192.168.200.226
[ceph_deploy.new][DEBUG ] Monitor initial members are ['226x']
[ceph_deploy.new][DEBUG ] Monitor addrs are ['192.168.200.226']
[ceph_deploy.new][DEBUG ] Creating a random mon key...
[ceph_deploy.new][DEBUG ] Writing monitor keyring to ceph.mon.keyring...
[ceph_deploy.new][DEBUG ] Writing initial config to ceph.conf...
```

### Install ca-cert packages

```
apt-get install openssl ca-certificates
```

If using PROXY (10.10.10.10) then:

```
vi ~/.wgetrc
```

Add this lines to the file and save it (change with your Proxy IP:port):

```
use_proxy=yes
http_proxy=10.10.10.10:80
https_proxy=10.10.10.10:80
```

Update packages again (to test if CEPH repo keys are properly imported)

```
apt-get update
```

If you receive error like this in the end , then proceed

An error occurred during the signature verification. The repository is not updated and the previous index files will be used.

GPG error: http://ceph.com wheezy Release:

The following signatures couldn't be verified because the public key

is not available: NO\_PUBKEY E85AC2C0460F3994

Copy the key:

```
gpg --keyserver pgpkeys.mit.edu --recv-key E85AC2C0460F3994
```

```
gpg: requesting key 460F3994 from hkp server pgpkeys.mit.edu
gpg: key 460F3994: public key "Ceph.com (release key) <security@ceph.com>"
imported
gpg: Total number processed: 1
gpg:             imported: 1 (RSA: 1)
```

Now we can import it

```
gpg -a --export E85AC2C0460F3994 | apt-key add -
```

Now try to update again

```
apt-get update
```

⇒ There must be no more errors

**Install CEPH packages on first server (in this example on 226x ) ⇒ Repeat this step on all other CEPH servers/nodes**

```
ceph-deploy install 226x
```

```
[ceph_deploy.conf][DEBUG ] found configuration file at:
/root/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (1.5.22): /usr/bin/ceph-deploy install |
226x
[ceph_deploy.install][DEBUG ] Installing stable version giant on cluster ceph
hosts 226x
[ceph_deploy.install][DEBUG ] Detecting platform for host 226x ...
[226x][DEBUG ] connected to host: 226x
[226x][DEBUG ] detect platform information from remote host
[226x][DEBUG ] detect machine type
[ceph_deploy.install][INFO  ] Distro info: debian 7.8 wheezy
[226x][INFO  ] installing ceph on 226x
[226x][INFO  ] Running command: env DEBIAN_FRONTEND=noninteractive apt-get -q
install --assume-yes ca-certificates
[226x][DEBUG ] Reading package lists...
[226x][DEBUG ] Building dependency tree...
[226x][DEBUG ] Reading state information...
[226x][DEBUG ] ca-certificates is already the newest version.
[226x][DEBUG ] 0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
[226x][INFO  ] Running command: wget -O release.asc
https://ceph.com/git/?p=ceph.git;a=blob_plain;f=keys/release.asc --no-check-
certificate
[226x][WARNIN] --2015-08-24 17:17:22--
https://ceph.com/git/?p=ceph.git;a=blob_plain;f=keys/release.asc
[226x][WARNIN] Connecting to 10.10.10.10:80... connected.
[226x][WARNIN] WARNING: The certificate of `ceph.com' is not trusted.
[226x][WARNIN] WARNING: The certificate of `ceph.com' hasn't got a known
issuer.
[226x][WARNIN] Proxy request sent, awaiting response... 301 Moved Permanently
[226x][WARNIN] Location:
https://git.ceph.com/?p=ceph.git;a=blob_plain;f=keys/release.asc [following]
[226x][WARNIN] --2015-08-24 17:17:24--
https://git.ceph.com/?p=ceph.git;a=blob_plain;f=keys/release.asc
[226x][WARNIN] Connecting to 10.10.10.10:80... connected.
[226x][WARNIN] WARNING: The certificate of
`git.ceph.com[ceph_deploy.conf][DEBUG ] found configuration file at:
/root/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (1.5.22): /usr/bin/ceph-deploy install 226x
[ceph_deploy.install][DEBUG ] Installing stable version giant on cluster ceph
host 226x
[ceph_deploy.install][DEBUG ] Detecting platform for host 226x ...
[226x][DEBUG ] connected to host: 226x
[226x][DEBUG ] detect platform information from remote host
[226x][DEBUG ] detect machine type
[ceph_deploy.install][INFO  ] Distro info: debian 7.8 wheezy
[226x][INFO  ] installing ceph on 226x
```



```

[226x][INFO ] Running command: env DEBIAN_FRONTEND=noninteractive apt-get -q
install --assume-yes ca-certificates
[226x][DEBUG ] Reading package lists...
[226x][DEBUG ] Building dependency tree...
[226x][DEBUG ] Reading state information...
[226x][DEBUG ] ca-certificates is already the newest version.
[226x][DEBUG ] 0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
[226x][INFO ] Running command: wget -O release.asc
https://ceph.com/git/?p=ceph.git;a=blob_plain;f=keys/release.asc --no-check-
certificate
[226x][WARNIN] --2015-08-24 17:17:22--
https://ceph.com/git/?p=ceph.git;a=blob_plain;f=keys/release.asc
[226x][WARNIN] Connecting to 10.10.10.10:80... connected.
[226x][WARNIN] WARNING: The certificate of `ceph.com' is not trusted.
[226x][WARNIN] WARNING: The certificate of `ceph.com' hasn't got a known
issuer.
[226x][WARNIN] Proxy request sent, awaiting response... 301 Moved Permanently
[226x][WARNIN] Location:
https://git.ceph.com/?p=ceph.git;a=blob_plain;f=keys/release.asc [following]
[226x][WARNIN] --2015-08-24 17:17:24--
https://git.ceph.com/?p=ceph.git;a=blob_plain;f=keys/release.asc
[226x][WARNIN] Connecting to 10.10.10.10:80... connected.
[226x][WARNIN] WARNING: The certificate of `git.ceph.com
! DO IT ON ONLY FIRST CEPH NODE (SERVER) !
Define Network for CEPH, in our case we are using separate network interfaces in range :
192.168.200.0 / 24 (255.255.255.0)
pveceph init --network 192.168.200.0/24
! DO IT ON ONLY FIRST CEPH NODE (SERVER) !

```

## Monitor Servers

Create CEPH MONITOR SERVER (in our case all [at least 3 servers] servers will have MONITOR role): (Do it on all 3 nodes)

```

pveceph createmon

creating /etc/pve/priv/ceph.client.admin.keyring
monmaptool: monmap file /tmp/monmap
monmaptool: generated fsid 1a52a990-0e39-40b5-bf5c-40a99b314354
epoch 0
fsid 1a50a990-0e39-40b5-bf5c-40a99b314354
last_changed 2015-08-25 12:19:38.848959
created 2015-08-25 12:19:38.848959
0: 192.168.200.226:6789/0 mon.0
monmaptool: writing epoch 0 to /tmp/monmap (1 monitors)
ceph-mon: set fsid to 12d7e250-f08f-4633-9994-e50830248359
ceph-mon: created monfs at /var/lib/ceph/mon/ceph-0 for mon.0
=== mon.0 ===
Starting Ceph mon.0 on 226x...
Starting ceph-create-keys on 226x...
(do it on at least 3 nodes)
Since we wish to have 3 Monitor nodes

```

Check do we have a working monitors Go to Proxmox Web GUI , select first node ⇒ CEPH → Go down and select MONITOR:



Look at the : Quorum (must be Yes for all mon nodes)

We must see the first monitor node (after adding all nodes as monitor role, we will see them all).

## Proxmox VE and CEPH Auth

AFTER ALL NODES (SERVERS) are added to CEPH Cluster go to the next step.

Authentication CEPH ← → Proxmox

Create CEPH directory on Proxmox cluster storage :

```
mkdir /etc/pve/priv/ceph
```

Copy keys to it (name of the CEPH Pool that we will later create will be CEPH01):

```
cp /etc/ceph/ceph.client.admin.keyring /etc/pve/priv/ceph/CEPH01.keyring
```

Edit /etc/pve/ceph.conf

Edit line :

```
keyring = /etc/pve/priv/c...
```

and change it to :

```
keyring = /etc/pve/priv/ceph/CEPH01.keyring
```

try :

```
ceph -s
```

if getting error like this

```
2015-09-01 13:59:57.617827 7f46d9158700 0 librados: client.admin
```

```
authentication error (1) Operation not permitted
```

```
Error connecting to cluster: PermissionError
```

Then

You can just not use auth using keys (we are on separate network on CEPH after all and this is test anyway) Edit file : **/etc/pve/ceph.conf**

Change line :

```
auth supported = cephx
```

to

```
auth supported = none
```

This is not recommended in real production use (configure Authentication)

Restart all nodes

## Checking status

Shell commands: To see the status of ceph

```
ceph -s
```

```
cluster 12d7e250-f08f-4633-9994-e50830248359
health HEALTH_WARN
    too many PGs per OSD (400 > max 300)
monmap e3: 3 mons at
{0=192.168.200.224:6789/0,1=192.168.200.226:6789/0,2=192.168.200.223:6789/0}
    election epoch 20, quorum 0,1,2 2,0,1
osdmap e167: 16 osds: 16 up, 16 in
pgmap v427893: 3200 pgs, 1 pools, 62491 MB data, 15778 objects
    122 GB used, 13194 GB / 13317 GB avail
    3200 active+clean
client io 10163 B/s wr, 3 op/s
```

⇒ we must have all CEPH mon nodes here (for now we have it 3, but will add the third one now)

Our CEPH configuration is looking like this (at this moment):

Our CEPH configuration: (/etc/pve/ceph.conf) → link to /etc/ceph/ceph.conf

```
[global]
    auth client required = cephx
    auth cluster required = cephx
    auth service required = cephx
    auth supported = none
    cluster network = 192.168.200.0/24
    filestore xattr use omap = true
    fsid = 12d7e250-f08f-4633-9994-e50830248359
    keyring = /etc/pve/ceph.client.admin.keyring
    osd journal size = 5120
    osd pool default min size = 1
    public network = 192.168.200.0/24

[osd]
    keyring = /var/lib/ceph/osd/ceph-$id/keyring

[mon.0]
    host = 224x
    mon addr = 192.168.200.224:6789

[mon.1]
    host = 226x
    mon addr = 192.168.200.226:6789

[mon.2]
    host = 223x
    mon addr = 192.168.200.223:6789
```

## Reverting installation if something goes wrong

Revert installation There are useful commands to purge the Ceph installation and configuration from every node so that one can start over again from a clean state. This will remove Ceph configuration and keys

```
ceph-deploy purgedata {ceph-node} [{ceph-node}]
```

```
ceph-deploy forgetkeys
```

This will also remove Ceph packages

```
ceph-deploy purge {ceph-node} [{ceph-node}]
```

## Preparing DISK drives for CEPH

Be aware that CEPH can use ONLY ENTIRE Disk drive, without any partition or filesystem on it. CEPH internally use replication, default is 3 replicas of every object ( on different nodes ).

WE DO NOT NEED ANY RAID, just JBOD disks

Prepare one Disk in one JBOD ( Inside RAID controller BIOS) One disk will become one OSD on every node: 1 Hard disk = 1 OSD

We will have 8 disks per server:

/dev/sdb

/dev/sdc

/dev/sdd

/dev/sde

/dev/sdf

/dev/sdg

/dev/sdh

/dev/sdi

For performance reasons Journal SHOULD be used on separate SSD disk → One SSD can be used for entire Server as Journal disk if needed (proposal is to use it more than one)

SSD disk must be Enterprise grade Reliable and Durable version of SSD (For Example Intel DC S3700)

### First step

Login to Proxmox web GUI

Go to the first node (in our case on Proxmox VE node **224X**)

Go to CEPH ⇒ Disks , Select disk /dev/sdb (this is our first disk used for CEPH):

Create: OSD				
Device	Usage	Size	Vendor	Model
/dev/sda	LVM	136.13GB	DELL	PERC H710
/dev/sdb	No	837.75GB	DELL	PERC H710
/dev/sdc	No	837.75GB	DELL	PERC H710
/dev/sdd	No	837.75GB	DELL	PERC H710
/dev/sde	No	837.75GB	DELL	PERC H710
/dev/sdf	No	837.75GB	DELL	PERC H710
/dev/sdg	No	837.75GB	DELL	PERC H710
/dev/sdh	No	837.75GB	DELL	PERC H710
/dev/sdi	No	837.75GB	DELL	PERC H710

Be aware that in our case disk /dev/sda is system disk used for Proxmox VE

Go to → **Create OSD**



Selected disk must be **/dev/sdb**

Under Journal Disk leave as it is (Use OSD disk), since we have no SSD disks for that purpose. Select **CREATE**.

### What is happening in the background ?

In the background system will create 2 partitions :

- One for CEPH (partition Nr. 1 [sdb1 , sdc1 , ...])
- One for journal (partition Nr. 2 [sdb2 , sdc2 , ...])

Like this :

```
...
sdb                8:16    0    1.8T    0 disk
├─sdb1              8:17    0    1.8T    0 part /var/lib/ceph/osd/ceph-16
└─sdb2              8:18    0         5G    0 part
...
```

Repeat this step for all disks that we have prepared for CEPH ()

And after some time we will see them all

Use command

```
lsblk
```

```
...  ...  ...
sdb                8:16    0    1.8T    0 disk
├─sdb1              8:17    0    1.8T    0 part /var/lib/ceph/osd/ceph-16
└─sdb2              8:18    0         5G    0 part
sdc                8:32    0    1.8T    0 disk
├─sdc1              8:33    0    1.8T    0 part /var/lib/ceph/osd/ceph-17
└─sdc2              8:34    0         5G    0 part
sdd                8:48    0    1.8T    0 disk
├─sdd1              8:49    0    1.8T    0 part /var/lib/ceph/osd/ceph-18
└─sdd2              8:50    0         5G    0 part
...  ...  ...
```

After all disks is visible we will see them mounted like this:

```
mount | grep ceph
```

```
/dev/sdb1 on /var/lib/ceph/osd/ceph-16 type xfs
(rw,noatime,attr2,delaylog,inode64,noquota)
/dev/sdc1 on /var/lib/ceph/osd/ceph-17 type xfs
(rw,noatime,attr2,delaylog,inode64,noquota)
/dev/sdd1 on /var/lib/ceph/osd/ceph-18 type xfs
(rw,noatime,attr2,delaylog,inode64,noquota)
```

```
/dev/sde1 on /var/lib/ceph/osd/ceph-19 type xfs
(rw,noatime,attr2,delaylog,inode64,noquota)
```

... ..

After some time CEPH will create all partitions and filesystem on this disk, and mount it.

Then after few moment this new disks will be visible as OSDs in Proxmox VE GUI, in CEPH , OSD Section, like in picture:

Device	Usage	Size	Vendor	Model	Serial
/dev/sda	LVM	136.13GB	DELL	PERC H710	6c81f660dc1547001a3dea861554a01e
/dev/sdb	osd.0	837.75GB	DELL	PERC H710	6c81f660dc1547001d785d680c1e959
/dev/sdc	osd.1	837.75GB	DELL	PERC H710	6c81f660dc1547001d785d860a9e7c42
/dev/sdd	osd.2	837.75GB	DELL	PERC H710	6c81f660dc1547001d785d8f0b161b40
/dev/sde	osd.3	837.75GB	DELL	PERC H710	6c81f660dc1547001d785d970b92400e
/dev/sdf	osd.4	837.75GB	DELL	PERC H710	6c81f660dc1547001d785d9e0bfabe0a
/dev/sdg	osd.5	837.75GB	DELL	PERC H710	6c81f660dc1547001d785da60c77dd9e
/dev/sdh	osd.6	837.75GB	DELL	PERC H710	6c81f660dc1547001d785db60d7373a5
/dev/sdi	osd.7	837.75GB	DELL	PERC H710	6c81f660dc1547001d785dc40e448ec3

Repeat this steps for all other CEPH disks (/dev/sdb through the ... .. /dev/sdi) on all other Proxmox VE (CEPH) nodes.

Look to second CEPH server (Proxmox VE node **226x**) after creating OSDs from disks:

Device	Usage	Size	Vendor	Model	Serial
/dev/sda	LVM	136.13GB	DELL	PERC H710	6c81f660d91525001a3ee5b9163f717f
/dev/sdb	osd.8	837.75GB	DELL	PERC H710	6c81f660d91525001d79706d0bb71e56
/dev/sdc	osd.9	837.75GB	DELL	PERC H710	6c81f660d91525001d7970770c5440c1
/dev/sdd	osd.10	837.75GB	DELL	PERC H710	6c81f660d91525001d7970ed1350970d
/dev/sde	osd.11	837.75GB	DELL	PERC H710	6c81f660d91525001d7970f713e7d5e4
/dev/sdf	osd.12	837.75GB	DELL	PERC H710	6c81f660d91525001d7970ff146faee8
/dev/sdg	osd.13	837.75GB	DELL	PERC H710	6c81f660d91525001d79710714e9610e
/dev/sdh	osd.14	837.75GB	DELL	PERC H710	6c81f660d91525001d79710f15646d34
/dev/sdi	osd.15	837.75GB	DELL	PERC H710	6c81f660d91525001d79711915f33e89

Be aware that OSD numbers are unique : on first node OSD 0 - 7 , on second node OSD 8 - 15 ,

...

Lets check this on all nodes (we must have the same info)

Go to our first node (**224x**) → CEPH ⇒ Now select in **CEPH ⇒ OSD** (down menu) and look at the first node (**224x**) and second node (**226x**):

**PROXMOX** Proxmox Virtual Environment  
Version: 3.4-9/4b51d87a

Server View Node: 226x

Search Summary Services Network DNS Time Syslog Bootlog Task History UBC Subscription Firewall Updates Ceph

Reload Start Stop Out In Remove

Name	Type	Status	weight	reweight	Used		Latency (ms)	
					%	Total	Apply	Commit
226x	host							
osd.15	osd	up/in	0.809998	1	0.00	832.34GB	2	0
osd.14	osd	up/in	0.809998	1	0.00	832.34GB	4	0
osd.13	osd	up/in	0.809998	1	0.00	832.34GB	3	0
osd.12	osd	up/in	0.809998	1	0.00	832.34GB	2	0
osd.11	osd	up/in	0.809998	1	0.00	832.34GB	4	0
osd.10	osd	up/in	0.809998	1	0.00	832.34GB	4	0
osd.9	osd	up/in	0.809998	1	0.00	832.34GB	6	1
osd.8	osd	up/in	0.809998	1	0.00	832.34GB	7	2
224x	host							
osd.7	osd	up/in	0.809998	1	0.00	832.34GB	1	0
osd.6	osd	up/in	0.809998	1	0.00	832.34GB	0	0
osd.5	osd	up/in	0.809998	1	0.00	832.34GB	0	0
osd.4	osd	up/in	0.809998	1	0.00	832.34GB	0	0
osd.3	osd	up/in	0.809998	1	0.00	832.34GB	0	0
osd.2	osd	up/in	0.809998	1	0.00	832.34GB	0	0
osd.1	osd	up/in	0.809998	1	0.00	832.34GB	2	1
osd.0	osd	up/in	0.809998	1	0.00	832.34GB	0	0

## CLI Commands

We can also see OSDs with cli command : `ceph osd tree`

```

ID WEIGHT  TYPE NAME                UP/DOWN REWEIGHT PRIMARY-AFFINITY
-1 20.19995 root default
-2  6.47998  host  224x
   0 0.81000    osd.0      up  1.00000      1.00000
   1 0.81000    osd.1      up  1.00000      1.00000
   2 0.81000    osd.2      up  1.00000      1.00000
   3 0.81000    osd.3      up  1.00000      1.00000
   4 0.81000    osd.4      up  1.00000      1.00000
   5 0.81000    osd.5      up  1.00000      1.00000
   6 0.81000    osd.6      up  1.00000      1.00000
   7 0.81000    osd.7      up  1.00000      1.00000
-3  6.47998  host  226x
   8 0.81000    osd.8      up  1.00000      1.00000
   9 0.81000    osd.9      up  1.00000      1.00000
  10 0.81000    osd.10     up  1.00000      1.00000
  11 0.81000    osd.11     up  1.00000      1.00000
  12 0.81000    osd.12     up  1.00000      1.00000
  13 0.81000    osd.13     up  1.00000      1.00000
  14 0.81000    osd.14     up  1.00000      1.00000
  15 0.81000    osd.15     up  1.00000      1.00000
-4  7.23999  host  223x
  16 1.81000    osd.16     up  1.00000      1.00000
  17 1.81000    osd.17     up  1.00000      1.00000
  18 1.81000    osd.18     up  1.00000      1.00000
  19 1.81000    osd.19     up  1.00000      1.00000

```



## Log files

Log files are as follows:

- OSD log files are located inside : `/var/log/ceph/` directory.
- OSD log has file name `ceph-osd.x.log` (x is OSD number).

## CEPH Pools

Ceph Pools Now we need to create a Pool for our CEPH Storage.

We will create test pool with 1(one) replica in the Cluster. This means that there is no guaranty if one OSD is down 😞 but we will use it just for the test

Go to one node in the **CEPH Cluster** ⇒ **CEPH** → **Pool** → **Create**

Here you can see the PG calculation basic formula:

### PG Calculation (formula):

$$PG = \frac{Nr.of\ OSDs \cdot 100}{Nr.of\ replicas}$$

For more details about PG calculation read

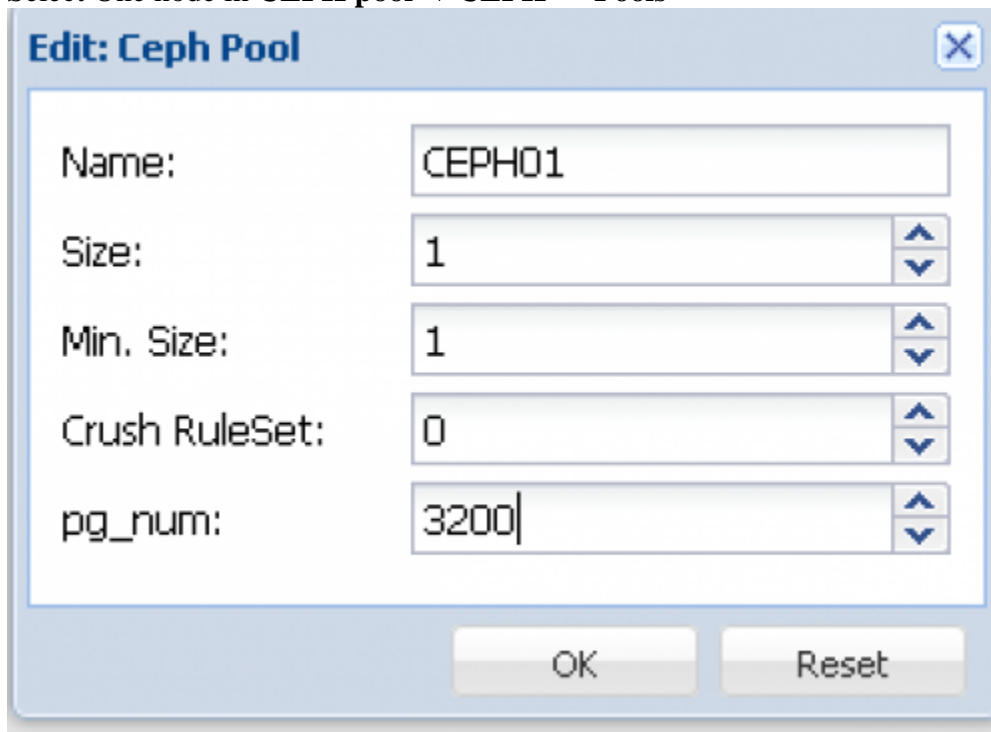
<http://docs.ceph.com/docs/master/rados/operations/placement-groups/>

Since we are planning very soon to have 4 CEPH servers we will calculate PG as it is :

- We will have 8 disks (OSDs) per server in CEPH cluster, 4 servers which is 32 OSDs:

$$PG = \frac{32 \cdot 100}{2} = 1600$$

Select One node in **CEPH pool** ⇒ **CEPH** → **Pools**



DO NOT USE Size =1 for Production (1= NO REPLICA)

We will later use Size = 2 which is minimum for Production use (Means 2 replicas).

So change pg\_num to **1600**.

For now we will use Pool Size =1 , just for playing with it, and will online change it in next steps  
My proposal is to use AT LEAST 3 (for CEPH POOL SIZE)

Basic CEPH performace formula, regarding number of replicas:

$$\text{CEPH performance} = \frac{1}{\text{Nr. of Replica}}$$

## Mounting CEPH RBD to Proxmox VE

On proxmox VE GUI , Go to **DATACENTER** ⇒ **STORAGE** → **Add** → **RBD**

Under:

- ID: Select Your NAME for the CEPH Storage (use different from Pool name)
- Pool : This is the name of CEPH POOL created on previous page (CEPH01)
- Monitor hosts : Enter IP addresses of all three nodes , separated by space :  
192.168.200.224 192.168.200.226 192.168.200.227
- Nodes : For which nodes you wish to enable Our CEPH Storage : Select one by one with holding on CTRL key

(For now we have only 2 nodes : 224x and 226x) - in production add them all

**Add: RBD**

ID:  Nodes:

Pool:  Enable: ☐

Monitor Host:

User name:

Node	Memory usage	CPU usage
224x	1.2%	0.3% of 32CPUs
226x	1.5%	0.5% of 32CPUs
227x	40.6%	4.4% of 40CPUs

Now go to any Node on which this Storage is enabled ⇒ Select our new Storage :**ceph-storage**:

**PROXMOX** Proxmox Virtual Environment  
Version: 3.4-9/4b51d87a

Server View

Storage 'ceph-storage' on node ' 224x'

**Summary** Content Permissions

Status	
Enabled	Yes
Active	Yes
Content	Disk image
Type	RBD
Shared	Yes

So it is here and ACTIVE. But we can NOT access it yet because **keyring** must be imported for it.

Connect with SSH to any node in CEPH Cluster.

We need to copy the keyring to ProxMox Cluster storage (pmxcfs) (used for ProxMox Clustering).

This is actually directory visible as : /etc/pve/

```
cd /etc/pve/priv/
```

```
mkdir ceph
```

Check the new pool from CLI (we have created Pool named “CEPH01”):

```
ceph osd lspools
```

```
1 CEPH01,
```

So we have this pool : “CEPH01” → This is OK

Check our Keyring file for that pool is named : as pool name in the syntax:

**POOLNAME.keyring** inside directory : /etc/pve/priv/ceph/

In our case : /etc/pve/priv/ceph/CEPH01.keyring

This file is a copy of master keyring file :

```
/etc/pve/ceph.client.admin.keyring
```

**How to check the POOL size (Pool Size=Number of Replicas): Syntax:**

```
ceph osd pool get POOLNAME size
ceph osd pool get CEPH01 size
size : 1
```

**So YES, Pool size is 1 (We will change this later - this is just for testing)**

**Disk usage of Pool(s): with command `rados df` :**

```
rados df
```

pool name	KB	objects	clones	degraded	unfound	rd	rd KB	wr
	wr KB							
CEPH01	9988929	2449	0	0	0	27692	509292	57132
	15452951							

`rados` is a utility for interacting with a Ceph object storage cluster (RADOS), part of the Ceph distributed storage system

Be sure that Original CEPH configuration files on ALL NODES must be Symbolic links to file inside ProxMox Cluster File System mounted as : `/etc/pve` :

```
/etc/ceph/ceph.client.admin.keyring - symlink to →
/etc/pve/ceph.client.admin.keyring
/etc/ceph/ceph.conf - symlink to → /etc/pve/ceph.conf
/etc/ceph/ceph.mon.keyring - symlink to → /etc/pve/ceph.mon.keyring
```

In that way all important CEPH files will be shared on all NODES in CEPH (and ProxMox ) CLUSTER.

### How to change Pool size

We will change Our Pool (CEPH01) to have 2 Replicas instead of 1 that is initially created. That means that one entire node (server) can be down at any time , including all OSD in that one server.

```
ceph osd pool set CEPH01 size 2
```

```
set pool 1 size to 2
```

Wait few moment Now check that our pool replica is set to 2

```
ceph osd pool get CEPH01 size
size: 2
```

It is 2 OK

After we change the size (number of replicas to 2) of this pool we have a half of capacity available.

**Disk usage of CEPH with command `ceph df` :**

```
ceph df
GLOBAL:
      SIZE      AVAIL      RAW USED      %RAW USED
13317G  13278G           39649M           0.29
POOLS:
      NAME      ID      USED      %USED      MAX AVAIL      OBJECTS
CEPH01  1      19509M      0.14      6637G      4896
```

We can see this in GUI.

Go to any CEPH Server in the CEPH Cluster ⇒ CEPH → Select POOLS:



Name	Size/min	pg_num	ruleset	Used	
				%	Total
CEPH01	2/1	3200	0	0.00	9.53GB
				0.00	9.53GB

Under : **Size/min** , we now have **2/1** (2 is in use , 1 is minimum)

Now the pool will start replication inside the Cluster OSDs/PGs. This can take some time, depending on size.

## Adding a new nodes/disks/OSDs to CEPH Cluster

Adding a new nodes/disks/OSDs to CEPH Cluster procedure

Procedure is standard as adding nodes in the beginning:

1. Add a new node to Proxmox Cluster
2. Install all CEPH packages
3. Add a new CEPH node to CEPH Cluster
4. Create OSDs from DISKS
5. When finished new OSDs will expand the capacity of Entire CEPH Cluster
6. Redistribution of data across new CEPH cluster will automatically be started !!
7. When redistribution is finished status of CEPH cluster will be OK:

GO to Any node in CEPH cluster, Select CEPH tab , go down to STATUS (From Proxmox VE GUI):

Proxmox Virtual Environment  
Version: 3.4-11/6502936f

Server View

Node \* 223x

Datacenter

223x

MyBookDuo ( 223x)
Synology ( 223x)
ceph-storage ( 223x)
local ( 223x)

224x

100 (Test-Centos)
148 (Test-PMX01)
MyBookDuo ( 224x)
Synology ( 224x)
ceph-storage ( 224x)
local ( 224x)

226x

145 (Test-Centos2)
MyBookDuo ( 226x)
Synology ( 226x)
ceph-storage ( 226x)
local ( 226x)

227x

Pool01

Search
Summary
Services
Network
DNS
Time
Syslog
Bootlog
Task History
UBC
Subscription
Firewall
Updates
Ceph

health
quorum
cluster
monmap
osdmap
pgmap

HEALTH\_OK
Yes {2 0 1}
12d7e250-f08f-4633-9994-e50830248359
e3: 3 mons at 2=192.168.200.223:6789/0,0=192.168.200.224:6789/0,1=192.168.200.226:6789/0,
e295: 20 osds: 20 up, 20 in
v429107: 3200 pgs: 3200 active+clean; 61.03GB data, 123.34GB used, 20.14TB avail

8. Expanded Capacity will be visible in graph on CEPH mounted Storage :

Proxmox Virtual Environment  
Version: 3.4-11/6502936f

Server View

Storage 'ceph-storage' on node \* 224x

Summary
Content
Permissions

Status

Enabled	Yes
Active	Yes
Content	Disk image
Type	RBD
Shared	Yes
Size	20.26TB
Used	123.34GB
Avail	20.14TB

Usage

60

## Optimizing CEPH

### CEPH Cache

Configuration regarding CEPH is placed inside CEPH config file under Section **[client]**

We will enable Writeback cache on CEPH Level ( `rd cache = true` ). Safety parameter ( `rd cache writethrough until flush = true` ) and tells to start out in writethrough mode, and switch to writeback after the first flush request is received. I.e. if the client behaves properly (sends flush) it will be switched to writeback mode.

Add this lines :

```
rd cache = true
rd cache writethrough until flush = true
```

In the first line we are enabling CACHE I the second set it to safe option

Read more at: <http://docs.ceph.com/docs/hammer/rd/rd-config-ref/>

### Adding/Removing disks/OSDs

When you add or remove Ceph OSD Daemons to a cluster, the CRUSH algorithm will want to rebalance the cluster by moving placement groups to or from Ceph OSD Daemons to restore the balance. The process of migrating placement groups and the objects they contain can reduce the cluster's operational performance considerably. To maintain operational performance, Ceph performs this migration with 'backfilling', which allows Ceph to set backfill operations to a lower priority than requests to read or write data.

```
osd max backfills
```

Description: The maximum number of backfills allowed to or from a single OSD Default value: 10

We will change this value to 1 which will be good enough for the Adding or Removing OSD/Disk. This value will add more priority to load on CEPH storage (Virtual Machines) and less priority on rebalancing, which can consume huge amount of resources (especially NETWORK).

### Recovery process

If a Ceph OSD Daemon crashes and comes back online, usually it will be out of sync with other Ceph OSD Daemons containing more recent versions of objects in the placement groups. When this happens, the Ceph OSD Daemon goes into recovery mode and seeks to get the latest copy of the data and bring its map back up to date. Depending upon how long the Ceph OSD Daemon was down, the OSD's objects and placement groups may be significantly out of date. Also, if a failure domain went down (e.g., a rack), more than one Ceph OSD Daemon may come back online at the same time. This can make the recovery process time consuming and resource intensive. To maintain operational performance, Ceph performs recovery with limitations on the number recovery requests, threads and object chunk sizes which allows Ceph perform well in a degraded state. We will tune only one parameter:

```
osd recovery max active
```



Description: The number of active recovery requests per OSD at one time. More requests will accelerate recovery, but the requests place an increased load on the cluster. Default value : 15  
We will change this value to 1 which will be good enough for the Recovery process. This value will add more priority to load on CEPH storage (Virtual Machines) and less priority on recovery+rebalancing, which can consume huge amount of resources (especially NETWORK).

## Recovery Tuning

Recovery tuning :

We will limit Recovery process to only one thread

```
osd recovery threads = 1
```

We will limit Recovery process to low priority

```
osd recovery op priority = 1
```

## Configuration part (what we will change)

Edit file : /etc/pve/ceph.conf

Go to the section [osd]

For now we have only this options :

```
[osd]
    keyring = /var/lib/ceph/osd/ceph-$id/keyring
```

We will add our parameters, so the new configuration of this section will look like :

```
[client]
    rbd cache = true
    rbd cache writethrough until flush = true
```

```
[osd]
    keyring = /var/lib/ceph/osd/ceph-$id/keyring
    osd max backfills = 1
    osd recovery max active = 1
    osd recovery threads = 1
    osd recovery op priority = 1
```

Just save the config file.

Now we can inject this rules to OSDs.

Go to Shell (on any CEPH node) and type:

```
ceph tell osd.* injectargs '--osd-recovery-max-active 1'
ceph tell osd.* injectargs '--osd-max-backfills 1'
ceph tell osd.* injectargs '--osd-recovery-threads 1'
ceph tell osd.* injectargs '--osd-recovery-op-priority 1'
ceph tell osd.* injectargs '--osd-client-op-priority 63'
```

At this point we have injected this configuration options to all working OSDs/Disks

OSD Config guide:

<http://docs.ceph.com/docs/master/rados/configuration/osd-config-ref/>

## Monitoring and configuring CEPH

### Monitoring disk inside OSD - for error status

It is very important to monitor disk drives inside every OSD for errors (at least for SMART errors) and to be sure that OSD is properly started (in case of every Server reboot).

### Monitoring I/O

It can be done of disk level with normal commands like

`iostat` or `iostat -x` commands

Also we can monitor things at CEPH level, in realtime with command `ceph -w` :

`ceph -w`

```
cluster 12d7e250-f08f-4633-9994-e50830248359
health HEALTH_OK
monmap e2: 2 mons at
{0=192.168.200.224:6789/0,1=192.168.200.226:6789/0}
election epoch 10, quorum 0,1 0,1
osdmap e110: 16 osds: 16 up, 16 in
pgmap v823: 3200 pgs, 1 pools, 9753 MB data, 2449 objects
14100 MB used, 13303 GB / 13317 GB avail
3200 active+clean
client io 14429 B/s rd, 271 MB/s wr, 347 op/s

2015-09-03 13:57:54.893536 mon.0 [INF] pgmap v822: 3200 pgs: 3200
active+clean; 9753 MB data, 14214 MB used, 13303 GB / 13317 GB avail; 136
MB/s wr, 169 op/s
2015-09-03 13:57:55.932455 mon.0 [INF] pgmap v823: 3200 pgs: 3200
active+clean; 9753 MB data, 14100 MB used, 13303 GB / 13317 GB avail; 14429
B/s rd, 271 MB/s wr, 347 op/s
2015-09-03 13:57:56.941932 mon.0 [INF] pgmap v824: 3200 pgs: 3200
active+clean; 9753 MB data, 14018 MB used, 13303 GB / 13317 GB avail; 15783
B/s rd, 252 MB/s wr, 322 op/s
2015-09-03 13:57:57.952952 mon.0 [INF] pgmap v825: 3200 pgs: 3200
active+clean; 9753 MB data, 13916 MB used, 13303 GB / 13317 GB avail; 12002
B/s rd, 175 MB/s wr, 243 op/s
...
...
```

Here is visible how is CEPH utilized in every second of time (real time) in MB/s or op/s

### Authentication subsystem

To list the cluster's keys and their capabilities, execute the following:

`ceph auth list`

For example :

`ceph auth list`

installed auth entries:

```
osd.0
    key: AQBIr+ZVg1/SFxAAKoNbBN+Xart0iWqb+RdWkA==
    caps: [mon] allow profile osd
    caps: [osd] allow *
osd.1
    key: AQBor+ZV+o7bOhAAEdTLJOOBowJX+P6amA+MmQ==
    caps: [mon] allow profile osd
    caps: [osd] allow *
osd.10
    key: AQDht+ZVhPO6GxAAAj3ln71KbzOAICHoFugEQQ==
    caps: [mon] allow profile osd
    caps: [osd] allow *
osd.11
    key: AQATuOZV1v72ERAA+UmdF1koCk8l4Ju6zkzHuQ==
    caps: [mon] allow profile osd
    caps: [osd] allow *
osd.12
    key: AQAkuOZVbBshNRAAiCIGSt77YIwWOa0n+zQtfw==
    caps: [mon] allow profile osd
    caps: [osd] allow *
osd.13
    key: AQBauOZVCLkQEBAaz8uFbhXBZlkyF6nwgQmouA==
    caps: [mon] allow profile osd
    caps: [osd] allow *
osd.14
    key: AQBQuOZV+MrMMhAA0IT6VBpYgtBGH zrZwBj8xg==
    caps: [mon] allow profile osd
    caps: [osd] allow *
... ..
```

## Monitoring MON nodes

We can monitor the status of MON nodes with command : `ceph -m HOSTNAME mon_status`

Lets see the first 2 MON nodes:

```
ceph -m 224x mon_status
```

```
{"name":"0","rank":0,"state":"leader","election_epoch":8,"quorum":[0,1],"outside_quorum":[],"extra_probe_peers":[],"sync_provider":[],"monmap":{"epoch":2,"fsid":"12d7e250-f08f-4633-9994-e50830248359","modified":"2015-09-02 09:35:30.961677","created":"2015-09-02 09:22:44.253110","mons":[{"rank":0,"name":"0","addr":"192.168.200.224:6789\0"}, {"rank":1,"name":"1","addr":"192.168.200.226:6789\0"}]}}
```

```
ceph -m osir226x mon_status
```

```
{"name":"1","rank":1,"state":"peon","election_epoch":8,"quorum":[0,1],"outside_quorum":[],"extra_probe_peers":[],"sync_provider":[],"monmap":{"epoch":2,"fsid":"12d7e250-f08f-4633-9994-e50830248359","modified":"2015-09-02 09:35:30.961677","created":"2015-09-02 09:22:44.253110","mons":[{"rank":0,"name":"0","addr":"192.168.200.224:6789\0"}, {"rank":1,"name":"1","addr":"192.168.200.226:6789\0"}]}}
```

## Monitoring CEPH status

Monitor CEPH with : `ceph health` or `ceph health detail` commands.

## Monitoring PG problems

It is normal for placement groups to enter states like “degraded” or “peering” following a failure. Normally these states indicate the normal progression through the failure recovery process. However, if a placement group stays in one of these states for a long time this may be an indication of a larger problem. For this reason, the monitor will warn when placement groups get “stuck” in a non-optimal state. Specifically, we check for:

- **inactive** - The placement group has not been active for too long (i.e., it hasn’t been able to service read/write requests).
- **unclean** - The placement group has not been clean for too long (i.e., it hasn’t been able to completely recover from a previous failure).
- **stale** - The placement group status has not been updated by a ceph-osd, indicating that all nodes storing this placement group may be down.

With the next commands you can watch PG (Placement Groups) problems:

```
ceph pg dump_stuck stale
ceph pg dump_stuck inactive
ceph pg dump_stuck unclean
```

For stuck stale placement groups, it is normally a matter of getting the right ceph-osd daemons running again. Check that we have all running daemons using OSD (1 Daemon = 1 OSD) on all nodes: Lets look at the first node which holds OSD from 0 to 7:

```
ps -ef |grep -i osd
```

```
root 6217      1  0 Sep02 ?      00:03:38 /usr/bin/ceph-osd -i 5 --pid-file
/var/run/ceph/osd.5.pid -c /etc/ceph/ceph.conf --cluster ceph
root 6505      1  0 Sep02 ?      00:04:31 /usr/bin/ceph-osd -i 6 --pid-file
/var/run/ceph/osd.6.pid -c /etc/ceph/ceph.conf --cluster ceph
root 6866      1  0 Sep02 ?      00:04:16 /usr/bin/ceph-osd -i 0 --pid-file
/var/run/ceph/osd.0.pid -c /etc/ceph/ceph.conf --cluster ceph
root 7213      1  0 Sep02 ?      00:03:38 /usr/bin/ceph-osd -i 3 --pid-file
/var/run/ceph/osd.3.pid -c /etc/ceph/ceph.conf --cluster ceph
root 7546      1  0 Sep02 ?      00:03:20 /usr/bin/ceph-osd -i 4 --pid-file
/var/run/ceph/osd.4.pid -c /etc/ceph/ceph.conf --cluster ceph
root 7852      1  0 Sep02 ?      00:02:41 /usr/bin/ceph-osd -i 2 --pid-file
/var/run/ceph/osd.2.pid -c /etc/ceph/ceph.conf --cluster ceph
root 8269      1  0 Sep02 ?      00:02:48 /usr/bin/ceph-osd -i 7 --pid-file
/var/run/ceph/osd.7.pid -c /etc/ceph/ceph.conf --cluster ceph
```

**⇒ There is a missing OSD.1 in this listing !**

Lets see how this is visible in CEPH health :

```
ceph -s
```

```
cluster 12d7e250-f08f-4633-9994-e50830248359
health HEALTH_WARN
    220 pgs stale
    220 pgs stuck stale
monmap e2: 2 mons at
{0=192.168.200.224:6789/0,1=192.168.200.226:6789/0}
```

```
election epoch 8, quorum 0,1 0,1
osdmap e96: 16 osds: 15 up, 15 in
pgmap v188: 3200 pgs, 1 pools, 0 bytes data, 0 objects
547 MB used, 12484 GB / 12485 GB avail
2980 active+clean
220 stale+active+clean
```

**We have 16 OSD but 15 is UP – 1 is down**

We can see the same thing looking at the OSD:

```
ceph osd stat
```

```
osdmap e96: 16 osds: 15 up, 15 in
```

Now to be sure lets look at health:

```
ceph health detail
```

```
HEALTH_WARN 220 pgs stale; 220 pgs stuck stale
pg 1.271 is stuck stale for 59541.042678, current state stale+active+clean,
last acting [1]
pg 1.73b is stuck stale for 59541.043850, current state stale+active+clean,
last acting [1]
pg 1.8d3 is stuck stale for 59541.044268, current state stale+active+clean,
last acting [1]
pg 1.734 is stuck stale for 59541.043859, current state stale+active+clean,
last acting [1]
pg 1.402 is stuck stale for 59541.043077, current state stale+active+clean,
last acting [1]
...
...
...
```

Yes we have a warning (**HEALTH\_WARN 220 pgs stale; 220 pgs stuck stale**) and we can see

We can query PG problems by every PG:

Lets look at the first **PG (1.271)** problems

```
ceph pg 1.271 query
```

```
Error ENOENT: i don't have pgid 1.271
```

If this PG have another errors we will see them but in our case OSD.1 is DOWN so this PGs counted here is not accessible.

We must fix **OSD.1** problems first.

Find out what is the problem.

Lets go to server on which this **OSD.1** should be running and look at the log file of **OSD.1**:

```
tail /var/log/ceph/ceph-osd.1.log
2015-09-02 16:39:57.689607 7fcelca59700 0 -- 192.168.200.224:6805/550293 >>
192.168.200.226:6825/18843 pipe(0x40c1000 sd=59 :60461 s=2 pgs=17 cs=3 l=0
c=0x346c420).fault with nothing to send, going to standby
2015-09-02 16:39:57.689758 7fcele06f700 0 -- 192.168.200.224:6805/550293 >>
192.168.200.226:6813/15239 pipe(0x3ae4000 sd=77 :6805 s=2 pgs=21 cs=3 l=0
c=0x3f2d600).fault with nothing to send, going to standby
2015-09-02 16:39:57.692547 7fce221b0700 0 -- 192.168.200.224:6805/550293 >>
192.168.200.226:6829/20139 pipe(0x4119000 sd=65 :47479 s=2 pgs=18 cs=3 l=0
c=0x3f2d080).fault with nothing to send, going to standby
2015-09-02 16:39:57.700167 7fce2109f700 0 -- 192.168.200.224:6805/550293 >>
192.168.200.226:6821/17739 pipe(0x4032000 sd=68 :46951 s=2 pgs=21 cs=3 l=0
c=0x3f2d1e0).fault with nothing to send, going to standby
```

```

2015-09-02 16:39:57.704510 7fce1de6d700 0 -- 192.168.200.224:6805/550293 >>
192.168.200.226:6817/16350 pipe(0x4037000 sd=57 :46199 s=2 pgs=18 cs=3 l=0
c=0x3f2d340).fault with nothing to send, going to standby
2015-09-02 16:39:57.706497 7fce20998700 0 -- 192.168.200.224:6805/550293 >>
192.168.200.226:6809/13382 pipe(0x3dd8000 sd=80 :6805 s=2 pgs=20 cs=3 l=0
c=0x36f6840).fault with nothing to send, going to standby
2015-09-02 16:39:57.871077 7fce21dac700 0 -- 192.168.200.224:6805/550293 >>
192.168.200.226:6805/12247 pipe(0x3978000 sd=51 :6805 s=2 pgs=20 cs=3 l=0
c=0x36f69a0).fault with nothing to send, going to standby
2015-09-02 16:57:44.281265 7fce3ccf3700 0 monclient: hunting for new mon
2015-09-02 16:57:45.368553 7fce245cb700 -1 osd.1 87 *** Got signal Terminated
***
2015-09-02 16:57:45.368622 7fce245cb700 0 osd.1 87 prepare_to_stop telling
mon we are shutting down

```

Look the messages :

**(going to standby)**

and

**(prepare\_to\_stop telling mon we are shutting down)**

It some time , we have rebooted this node (on which is OSD.1) and this node is also Monitor, so they got some time out because of MON node reboot and new election to second monitor in the same time when host holding this OSDs are rebooted.

(Now we yet do not have 3rd monitor NODE !! IT IS NECESSARY TO HAVE THEM 3 “”)

Lets see do we have some other problems :

Do we have inactives ?

```

ceph pg dump_stuck inactive
ok

```

This is OK

Do we have unclean ?

```

ceph pg dump_stuck unclean
ok

```

This is OK now

So we have a stale-s:

```

ceph pg dump_stuck stale

```

```

ok
pg_stat state up up_primary acting acting_primary
1.271 stale+active+clean [1] 1 [1] 1
1.73b stale+active+clean [1] 1 [1] 1
1.8d3 stale+active+clean [1] 1 [1] 1
...
...
...

```

**220 of them are stale**

⇐ This is a problem

Since we find out the cause of the problem lets start OSD.1 from Proxmox GUI,

Go to Server which hosts OSD.1 (224x)

CEPH ⇒ OSD: Select OSD.1 and go to START

Wait few moments and it is started :

# Latency considerations

## Consider SSD Journal device

Latency explained:

- Commit latency is how long it takes for an operation to be applied to disk — generally speaking, how long it takes the journal to write an entry.
- Apply latency is how long it takes to get applied to the backing filesystem (which can be throttled by various things to prevent us getting arbitrarily large amounts of dirty data).

## Disk performance calculation

Disk performance calculation 1. Run tests on one Disk drive, before installing CEPH. 2. (Just to see the performance on CEPH) Run tests on CEPH storage after installing CEPH

Test tool “fio” 1. Test on “raw” disk Go inside mounted storage directory and start fio test For example

```
mkdir /SHARED-STORAGE
cd /SHARED-STORAGE
fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1 --name=test -
-filename=test.4GB --bs=4k --iodepth=64 --size=4G --readwrite=randrw --
rwmixread=75
```

Go inside mounted storage directory (/SHARED-STORAGE) and start **fio** test Delete test file (test.4GB)

Write results in the table !

2. Test on CEPH , inside VM. Create one Virtual Machine which is using CEPH Storage and test with the same parameters as above.

Delete test file (test.4GB) Write results in the table and compare them.