

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 1 / 42
--	---	---

# **CEPH KURULUMU VE İŞLETİMİ EL KİTABI**

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 2 / 42
--	---	---

## Güncelleme Tarihçesi

Sürüm	Tarih	Tanım	Yazar
v1.0	29/4/2014	Belge Yaratılması İlk Sürüm	Onur Bektaş

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 3 / 42
--	---	---

## İÇERİK

1. AMAÇ .....	8
2. Ceph GİRİŞ.....	8
2.1. Ceph Okuma Notları.....	8
2.1.1. Genel .....	8
2.1.2. Depolama Sunucusu.....	8
2.1.3. Monitör Sunucusu.....	9
2.1.4. Bağlantı .....	9
2.1.5. Bellek.....	9
3. CEPH KURULUMU .....	9
3.1. Test Sistemi Donanım Bilgileri .....	9
3.2. Test Sistemi Bağlantı Şeması.....	10
3.3. İşletim Sistemi Yapılandırması .....	11
3.3.1. İşletim Sistemi Sürümü .....	11
3.3.2. etc/hosts Dosyasının Düzenlenmesi .....	11
3.3.3. NTP ile Saat Senkronizasyonu .....	11
3.3.4. etc/security/limits.conf Dosyası Dosya Açma Limitlerinin Arttırılması.....	11
3.3.5. SSD Diskler İçin I/O Scheduler ve Rotation Değerlerinin Değiştirilmesi .....	11
3.3.6. Disk read_ahead değerinin Arttırılması .....	12
3.3.7. Apparmor Yazılımının Kapatılması .....	12
3.4. XFS Dosya Sistemi Yapılandırması.....	12
3.4.1. Ceph XFS Dosya Sistemi Yapılandırması.....	12
3.4.2. Dosya Sistemi Yaratılması .....	12
3.4.3. Dosya Sistemi Mount Opsiyonları.....	13
3.5. Ceph Yazılımı Kurulumu .....	14
3.6. Ceph Calamari GUI Kurulumu .....	16
3.6.1. Calamari Kurulum için Gerekli Paketler .....	16
3.6.2. Calamari Server Kurulumu Paketi Hazırlanması.....	17
3.6.3. Calamari Client Kurulumu Paketi Hazırlanması.....	17

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 4 / 42
--	---	---

3.6.4.	Diamond Kurulumu Paketi Hazırlanması .....	18
3.6.5.	CALAMARİ Sunucusu Kurulumu .....	18
3.6.6.	CALAMARİ İstemcileri Kurulumu.....	20
4.	Ceph İŞLETİMİ .....	21
4.1.	Ceph Performans Optimizasyonu için Ayarlanabilecek Değişken Listesi.....	21
4.1.1.	Genel .....	21
4.1.2.	Filestore Kuyruğu .....	21
4.1.3.	OSD.....	21
4.1.4.	OSD Journal .....	21
4.1.5.	OSD Kurtarma Operastonarlı (Recovery) .....	22
4.1.6.	İstemci.....	22
4.2.	Ceph PG Durumları .....	22
4.2.1.	Peering .....	22
4.2.2.	Active .....	22
4.2.3.	Clean .....	23
4.2.4.	Degraded.....	23
4.2.5.	Recovering.....	23
4.2.6.	Backfilling .....	23
4.2.7.	Remapped .....	23
4.2.8.	Stale.....	23
4.3.	Ceph Scrub Yapılandırması.....	23
5.	CEPH YÖNETİMİ.....	25
5.1.	Cluster İşlemleri .....	25
5.1.1.	Ceph Değişkenlerinin Görüntülenmesi/Değiştirilmesi.....	25
5.2.	Cluster Bakım Modları.....	26
5.2.1.	NOIN.....	26
5.2.2.	NOOUT .....	26
5.2.3.	NOUP.....	26
5.2.4.	NODOWN .....	26

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 5 / 42
--	---	---

5.2.5.	PAUSE.....	26
5.2.6.	NOSCRUB .....	26
5.2.7.	NODEEP-SCRUB.....	27
5.2.8.	NOBACKFILL .....	27
5.2.9.	NORECOVER.....	27
5.3.	Cluster Sağlığı.....	27
5.3.1.	Cluster Sağlığı Görüntüleme .....	27
5.3.2.	Cluster Üzerinde Yapılan I/O Görüntüleme .....	27
5.3.3.	Cluster Kapasite Kullanımı Görüntüleme .....	28
5.3.4.	Depolama Birimi Üzerindeki Disk Yapılandırmasını Görüntüleme .....	28
5.3.5.	Depolama Birimi Üzerindeki Journal Yapılandırmasını Görüntüleme .....	28
5.4.	POOL İŞLEMLERİ.....	28
5.4.1.	Pool Yaratma.....	28
5.4.2.	Poll'ları Listeleme .....	28
5.4.3.	Pool Silme.....	29
5.4.4.	Pool İçindeki Objelerin Listelenmesi .....	29
5.4.5.	Pool İçindeki Objelerin Silinmesi.....	29
5.4.6.	Pool Replica Sayısının Değiştirilmesi .....	30
5.4.7.	Pool Min_size Değerinin Değiştirilmesi.....	30
5.5.	OSD İŞLEMLERİ .....	30
5.5.1.	OSD Görüntüleme .....	30
	OSD AĞACINI GÖRÜNTÜLEME .....	30
	OSD DUMP .....	31
5.5.2.	OSD Başlatma/Durdurma.....	31
5.5.3.	OSD Silme .....	31
5.5.4.	OSD Config Bilgisini Görüntüleme.....	31
5.5.5.	OSD Benchmark .....	32
5.5.6.	Tek Bir OSD'ye ait Değerleri Değiştirme .....	32
5.6.	Monitör Sunucusu.....	32

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 6 / 42
--	---	---

5.6.1.	Monitör Sunucusu Restart Etme .....	32
5.6.2.	Monitör İstatistikleri .....	32
5.6.3.	Quarum Bilgisini Öğrenme .....	32
5.7.	RBD (Rados BLOCK DEVİCE) İşlemleri.....	33
5.7.1.	RBD Yaratma .....	33
5.7.2.	RBD Bilgi Alma (RBD info).....	33
5.7.3.	RBD İstemcilere MAP Edilmesi .....	33
5.7.4.	İstemcilere MAP edilen RBD'lerin Görüntülenmesi .....	33
5.7.5.	İstemcilere MAP edilen RBD'lerin Unmap Edilmesi .....	33
5.7.6.	RBD Silinmesi.....	33
5.8.	CEPH Yazılımının Güncellenmesi.....	34
5.8.1.	Giant (0.87.1 -> 0.87.2) Güncellemsi.....	34
	Mevcut Sürüm Bilgisinin Öğrenilmesi .....	34
	İşletim Sistemi Güncelleştirmelerinin Yapılması .....	34
	Monitör Sunucusu Güncellenmesi .....	34
	Depolama Sunucuları Güncellenmesi .....	34
5.8.2.	Giant (0.87.2) -> Hammer (0.94.1) Güncellemesi .....	35
	Monitör Sunucusu.....	35
	Depolama Sunucuları .....	35
	Alınan Hatalar .....	35
	İstemcilerin Güncellenmesi.....	36
6.	CEPH TESTLERİ.....	36
6.1.	Test İçin Kullanılan Programlar .....	36
6.1.1.	Fio.....	36
	Fio Programı Argümanları .....	36
	Örnek Kullanım.....	37
	Genfio ile İş Dosyası Oluşturma .....	38
	Fio Sonuçlarının Grafik Haline Getirilmesi .....	39
6.1.2.	Rados Bench.....	39

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 7 / 42
--	---	---

Rados Bench ile Ceph Benchmark.....	39
7. CEPH Betikleri.....	40
7.1. Rados Benchmark Betiği .....	40
7.2. Fio Test Betiği.....	40
7.2.1. Raw Device Test .....	40
7.2.2. Fio rbd Engine ile Test.....	40
7.2.3. Dosya Sistemi Testi .....	41
7.3. Ceph Data Node Yaratma Betiği .....	41
7.4. Ceph OSD Silme Betiği.....	41

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 8 / 42
--	---	---

## 1. AMAÇ

Bu belge TUBİTAK-ULAKBİM bulut altyapısında ceph kullanılabilirliğinin araştırılması/test edilmesi sırasında elde edilen bilginin kaybolmaması amacı ile yazılmıştır.

## 2. Ceph GİRİŞ

### 2.1. Ceph Okuma Notları

#### 2.1.1. Genel

- Sunucu isimlendirme önemli, tercihen rack numarasını içermeli, farklı donanım disk sayısı var ise isimlendirme kısaltma olarak belirtilmeli.
- Intank tarafından organize edilen "[Ceph Day](#)" organizasyonlarının sunumları oldukça bilgilendirici, 12 Mart 2015 tarihinde San Francisco'da yapılan son organizasyonun sunumları [slideshare](#)'de mevcut.
- Cluster kapasitesinin %80'den fazla doldurulmaması öneriliyor.
- Çok fazla PG kullanımı özellikle recovery durumunda CPU yükünü çok arttırması nedeni ile önerilmiyor.
- Atanmış Obje Büyüklüğü 4MByte.
- CERN tarafından test edilip kullanılan [puppet modülü](#) var.
- CERN tarafından kullanılan [Ceph monitör](#) betikleri
- Monitör için kullanılan GUI'ler ve özellikleri hakkında güzel bir [sunum](#).
- "Intank Hardware Configuration Guide" 2013 yılında yazılmasına rağmen donanım gereksinimleri konusunda fikir veriyor.
- [CephNotes](#) sitesinde ceph belgelerinde bulamayacağınızı pratik kullanıma yönelik oldukça faydalı bilgiler bulunmakta.
- PG sayısını hesaplamak için [ceph.com](#) bir sayfa oluşturmuş.
- Ceph performansının farklı değişkenlerden nasıl etkilendiğinin analiz eden güzel bir [çalışma](#).
- Ceph cluster tasarlanırken SSD,PG,kapasite kullanımı hesabı için güzel bir [sunum](#).
- Red Hat [Ceph Belgesi](#).

#### 2.1.2. Depolama Sunucusu

- Cluster'ın bir disk/node arızasında hızlı olarak toparlaması için SSD performansı önemli.
- [CERN Sunumunda](#) işletim sisteminde disklerde işlem yapan updatedb benzeri servislerin ceph dizinleri için kapatılması önerilmiş.
- I/O OSD'ye aktarıldıktan sonra 1 kez journal'a 1 kez diske yazılıyor. filestore max sync interval değeri (atanmış değer 5sn) dolduğunda veriler journal'dan diske yazılıyor.
- OSD başında Intel CPU'lar için core başına 1Ghz ile planlama yapılması öneriliyor. (AMD ise core değerini 0.7 ile çarpılmalı)
- Journal için SSD/HDD oranı 1/4, 1/5 olarak tavsiye edilmiş.



	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 9 / 42
--	---	---

### 2.1.3. Monitör Sunucusu

- Production kurulumlarında node sayısı 20 olan örnek uygulamalarda 3 adet monitör sunucusu kullanımı yeterli görülmüş.
- Dell Sunumlarında 200 Node'a kadar 3 monitör sunucusu yeterli görmüş.
- [CERN Sunumunda](#) monitör sunucusunda LevelDB'de yaşadıkları problem nedeni ile disklerinin SSD kullanılmasını öneriyor

### 2.1.4. Bağlantı

- Bağlantı tipi olarak 10Gbps ethernet kullanılmış, OSD'lerin SSD üzerinde koştığı sistemlerde LACP ile 20Gbps olarak çalıştıranlarda mevcut
- Anahtarlama cihazlarında jumbo frame desteği açılmalı
- Anahtarlama cihazlarının birbirine bağlantısı 40Gbps olması öneriliyor.

### 2.1.5. Bellek

- Bellekler ECC korumalı olmalı.
- Sunucu üzerindeki tüm slotlar bant genişliğinin maksimum kullanımı için doldurulmalı.
- 64GByte kullanılması öneriliyor, 8 slot var ise 8GByte'lık bellekler kullanmak gerekiyor.
- OSD başına asgari 2GByte hafıza ayrılması öneriliyor.

## 3. CEPH KURULUMU

Sunucu olarak Dell R815 marka sunucular bulut kullanımı için provizyonlanmadan önce test amaçlı olarak kullanılmıştır.

### 3.1. Test Sistemi Donanım Bilgileri

6 Adet Dell R815 Sunucu, 5 Adedi data düğümü bir adedi monitör olarak kullanıldı. Data sunucularının özellikleri:

- 4 x AMD Opteron(tm) Processor 6174 İşlemci
- 128GByte Bellek
- 5x500GByte 7200RPM SAS Disk, 1 Adet 240G SSD Disk

```
root@ceph1:/home/ubuntu# lshw -short -C disk
H/W path Device Class Description
=====
/0/100/4/0/0/0/2.0.0 /dev/sda disk 499GB PERC H700
/0/100/4/0/0/0/2.1.0 /dev/sdb disk 499GB PERC H700
/0/100/4/0/0/0/2.2.0 /dev/sdc disk 499GB PERC H700
/0/100/4/0/0/0/2.3.0 /dev/sdd disk 499GB PERC H700
/0/100/4/0/0/0/2.4.0 /dev/sde disk 239GB PERC H700
/0/100/4/0/0/0/2.5.0 /dev/sdf disk 499GB PERC H700
```

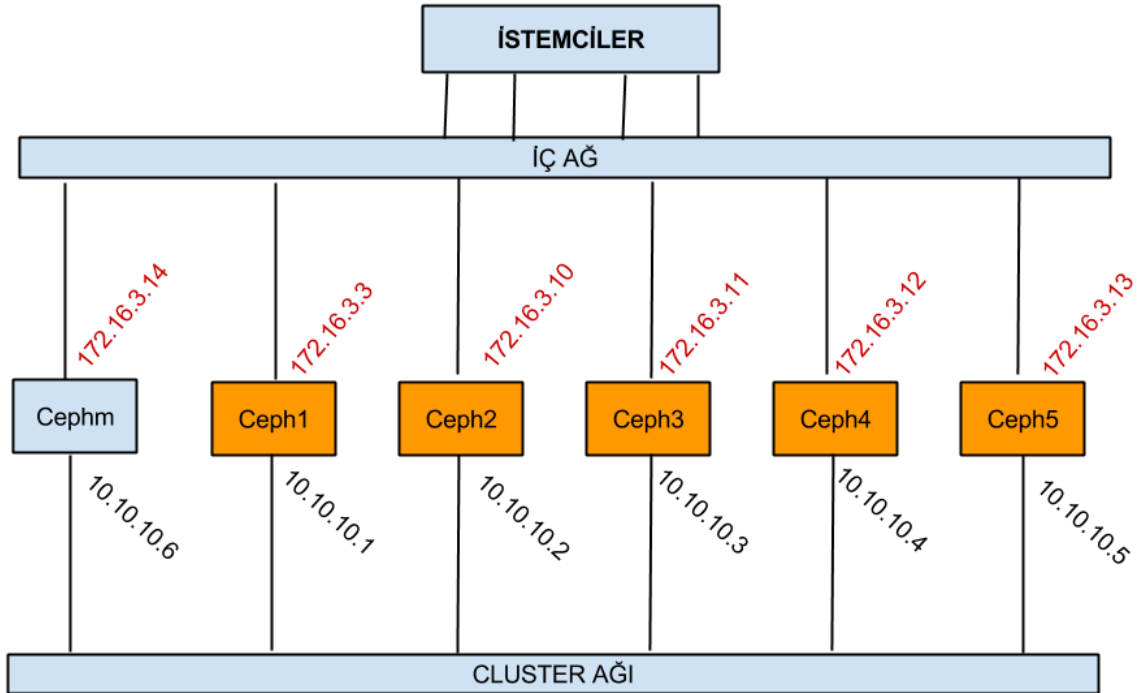
	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 10 / 42
--	---	--

- 4 Adet 1Gbps Ethernet kartı

```
root@ceph1:/home/ubuntu# lshw -short -C network
H/W path Device Class Description
=====
/0/100/2/0 em1 network NetXtreme II BCM5709 Gigabit Ethernet
/0/100/2/0.1 em2 network NetXtreme II BCM5709 Gigabit Ethernet
/0/100/3/0 em3 network NetXtreme II BCM5709 Gigabit Ethernet
/0/100/3/0.1 em4 network NetXtreme II BCM5709 Gigabit Ethernet
```

Sunucu bağlantıları için ilk olarak 1Gbps arayüz ile Juniper EX4300 Anaharalama cihaz ile yapıldı. Daha sonra Intel 82599 chiptetli 10Gbps ethernet kartları takılarak Cluster ağı 10Gbps olarak bağlandı.

### 3.2. Test Sistemi Bağlantı Şeması



	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 11 / 42
--	---	--

### 3.3. İşletim Sistemi Yapılandırması

#### 3.3.1. İşletim Sistemi Sürümü

İşletim sistemi olarak ubuntu MAAS aracılığı ile ubuntu-14.04 kuruldu. Ubuntu 14.04 ile gelen 3.13.0-48 çekirdeği 3.15 ve üzeri çekirdekler ile gelen xfs dosya sistemindeki değişiklikleri (boş olan inode'ların takibi için btree desteği) içermediğinden işletim sistemi çekirdeği

```
apt-get -y install linux-image-generic-lts-utopic
```

kumutu ile 3.16.0-33 olarak güncellendi.

#### 3.3.2. /etc/hosts Dosyasının Düzenlenmesi

Sunucuların birbirinden haberdar olması için /etc/hosts dosyasına sunucu bilgileri eklendi.

```
172.16.3.3 ceph1
172.16.3.10 ceph2
172.16.3.11 ceph3
172.16.3.12 ceph4
172.16.3.13 ceph5
172.16.3.14 cephm
```

#### 3.3.3. NTP ile Saat Senkronizasyonu

Sunucuların saatlerinin senkronize edilmesi için sisteme ntp sunucu kurup gerekli yapılandırma gerçekleştirildi.

#### 3.3.4. /etc/security/limits.conf Dosyası Dosya Açma Limitlerinin Arttırılması

Kullanıcıların açabileceği maksimum dosya tanımlayıcısı (file descriptor) sayısını /etc/security/limits.conf dosyasına aşağıdaki satırları ekleyerek arttırıyoruz.

```
* soft nofile 65000
* hard nofile 65000
```

#### 3.3.5. SSD Diskler İçin I/O Scheduler ve Rotation Değerlerinin Değiştirilmesi

SSD disklerin scheduler'ı atanmış değerinden SSD diskler için daha uygun olan noop'a değiştirmek için: (SSD disk sde)

```
echo noop > /sys/block/sde/queue/scheduler
```

yazmak yeterli. SSD diskler klasik diskler gibi dönen hareketli parçalardan oluşmadığı için SSD diskleri kullanan işletim sistemlerinin I/O operasyonlarını bu durumu göz önüne alarak gerçekleştirmeleri gerekiyor. Bu nedenle SSD disklerin rotation değeri 0 olmalı. Mevcut durumu görmek için:

```
root@ceph1:/home/ceph# cat /sys/block/sde/queue/rotational
1
```

komutunu kullandığımızda değerin 1 olduğunu görüyoruz. Normalde işletim sisteminin diskin SSD olduğunu anlayıp bu değer 0 yapmasını bekliyoruz. Manüel olarak 0 yapmak için:

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 12 / 42
--	---	--

```
echo "1" > /sys/block/sde/queue/rotational
```

yazmak yeterli. Açılıştaki geçerli olması için echo ile başlayan komutları satırı /etc/rc.local'a ekleyebilirsiniz.

### 3.3.6. Disk read\_ahead değerinin Arttırılması

Disklerin atanmış değerleri olan 128, 2048 olarak değiştirildi. Mevcut değeri görmek için:

```
root@ceph1:/home/ceph# cat /sys/block/sde/queue/read_ahead_kb
128
```

Yeni değerin atanması:

```
root@ceph1:/home/ceph# echo 2084 > /sys/block/sde/queue/read_ahead_kb
root@ceph1:/home/ceph# cat /sys/block/sde/queue/read_ahead_kb
2084
```

OSD diskleri için değerleri açılıştaki geçerli kılmak için aşağıdaki satırları /etc/rc.local dosyasına ekleyebilirsiniz.

```
echo 2084 > /sys/block/sdb/queue/read_ahead_kb
echo 2084 > /sys/block/sdc/queue/read_ahead_kb
echo 2084 > /sys/block/sdd/queue/read_ahead_kb
echo 2084 > /sys/block/sde/queue/read_ahead_kb
echo 2084 > /sys/block/sdf/queue/read_ahead_kb
```

read\_ahead\_kb değerinin arttırılması sıralı okuma (sequential read) performansınızı arttıracaktır. I/O örüntünüze göre bu değeri daha arttırarak performans kazanımı sağlayabilirsiniz. Bu değerin arttırılması hafıza kullanımının artmasına neden olacağı için bellek kullanımı üzerinde bir baskı oluşturabilir.

### 3.3.7. Apparmor Yazılımının Kapatılması

```
invoke-rc.d apparmor kill
update-rc.d -f apparmor remove
```

## 3.4. XFS Dosya Sistemi Yapılandırması

### 3.4.1. Ceph XFS Dosya Sistemi Yapılandırması

Ceph'te ext4,btrfs ve xfs dosya sistemi olarak kullanılabilir. Btrfs'in henüz production için kullanımı önerilmiyor. Ceph belgelerinde production sistemler için xfs kullanımı önerildiği için kurulumu xfs üzerinden gerçekleştiriyoruz.

Dosya sisteminin ne olacağı, mkfs ve mount opsiyonları ceph.conf dosyasında tanımlanıyor.

### 3.4.2. Dosya Sistemi Yaratılması

Disklerin dosya sistemini yaratırken mkfs.xfs programında ilk olarak size=2048 argümanını kullanıyoruz. Bunun nedeni ceph tarafından kullanılan Extended Attributes (XATTRs) daha fazla alan açmak. Size

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 13 / 42
--	---	--

argümanı inode büyüklüğü tanımlamak için kullanılıyor. Atanmış değeri (default) 256 ki buda 256Byte'a karşılık geliyor. 512 Byte inode kullanımında ortalama 350 Byte'lık bir alan xattr kullanımına kalıyor. Bu değeri 2048 yaptığımızda yaklaşık1900 Byte alanı xattr için ayırmış oluyoruz.

mkfs.xfs programına argüman olarak kullandığımız opsiyonlardan bir diğeri finobt=1. Bu argüman xfs'in boş inode'ların takibini btree algoritması ile yapılmasını sağlıyor. Bu özelliği kullanmak için çekirdek sürümünüzün 3.15 xfsprogs sürümünüzün 3.2 ve üzeri olması gerekiyor.

Ubuntu 14.04 ile xfsprogs sürümü 3.1.9 olarak geliyor. xfsprogs 3.2 ve üzerinde gelen finobt argümanının kullanabilmek için xfsprogs paketini aşağıdaki komutları kullanarak güncelliyoruz.

```
wget -c http://mirrors.kernel.org/ubuntu/pool/main/x/xfsprogs/xfsprogs_3.2.1ubuntu1_amd64.deb
dpkg -i xfsprogs_3.2.1ubuntu1_amd64.deb
```

mkfs.xfs programında kullandığımız üçüncü argüman crc=1. Bu argüman xfs'in metadatadaki bozulmaları karcı CRC32 algoritmasını kullanarak denetlemesini sağlıyor. Hepsi bir arada komut sdb diski için dosya sistemi yaratmak istersek:

```
mkfs.xfs -m crc=1,finobt=1 -f -i size=2048 /dev/sdb
```

halini alıyor.

### 3.4.3. Dosya Sistemi Mount Opsiyonları

Örnek olarak verdiğimiz sdb diskinin fstab'a eklemek için:

```
/dev/sdb /disk1 xfs rw,noatime,inode64,logbsize=256k,delaylog 0 1
```

Kullandığımız mount opsiyonları açıklamaları:

- **noatime:** Dosya sisteminin en son erişim zamanı bilgisini tutmaması için
- **inode64:** 1TByte üzeri disklerde kullanılması önerilen ve inode'ların 64 bit olarak tanımlanması
- **logbsize=256k :** Çok fazla okuma yazma yapan dosya sistemleri için artırılması öneriliyor. Atanmış değeri 32k
- **delaylog:** Metadata bilgisinin bellekte mümkün olduğunca tutulup yapılan I/O miktarını azaltan bir opsiyon.

Ceph'te dosya sistemini yaratılırken hangi opsiyonların kullanılacağını ceph.conf'ta [osd] başlığı altında tanımlıyoruz. Elle mkfs yapmanıza yada mount etmenize gerek yok.

```
[osd]
osd mkfs type = xfs
osd mkfs options xfs = -m crc=1,finobt=1 -f -i size=2048
osd mount options xfs = rw,noatime,inode64,logbsize=256k,delaylog
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 14 / 42
--	---	--

### 3.5. Ceph Yazılımı Kurulumu

İlk olarak tüm sunucularda ceph kullanıcısını açıyoruz.

```
root# useradd -d /home/ceph -m ceph
root# apt-get install sudo
root# echo "ceph ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/ceph
root# chmod 0440 /etc/sudoers.d/ceph
```

Ceph monitor sunucusunda (cephm) ssh anahtarlarını yaratıyoruz:

```
ceph@cephm: ssh-keygen -t dsa
```

Yaratığımız ssh anahtarlarını ceph depolama sunucularına (ceph1,ceph2,ceph3,ceph4,ceph5) kopyalıyoruz:

```
ceph@cephm: ssh-copy-id -i .ssh/id_dsa.pub ceph@ceph1
ceph@cephm: ssh-copy-id -i .ssh/id_dsa.pub ceph@ceph2
ceph@cephm: ssh-copy-id -i .ssh/id_dsa.pub ceph@ceph3
ceph@cephm: ssh-copy-id -i .ssh/id_dsa.pub ceph@ceph4
ceph@cephm: ssh-copy-id -i .ssh/id_dsa.pub ceph@ceph5
```

Tüm sunucularda Ceph reposunu ekleyip ceph'i kuruyoruz:

```
root# wget -q -O- 'https://ceph.com/git/?p=ceph.git;a=blob_plain;f=keys/release.asc' | sudo apt-key add -
root# echo deb http://ceph.com/debian-giant/trusty main | sudo tee /etc/apt/sources.list.d/ceph.list
root# apt-get update
root# apt-get install ceph-deploy
```

Cephm sunucusunda ceph kullanıcısı ile aşağıdaki komutları çalıştırıyoruz. İlk olarak config isimli bir dizin yaratıp yaptığımız yapılandırmalarının bu dizin altında yer almasını sağlayalım.

```
mkdir config
cd config
```

ceph-deploy ile ilk sunucumuz olan monitor sunucusunu tanımlayalım.

```
ceph-deploy new cephm
```

Bu komuttan sonra bulunduğunuz dizinin içinde ceph.mon.keyring dosyası ile ceph.conf dosyasının oluşmuş olması gerekiyor. ceph.conf dosyasına [global] tanımı altına aşağıdaki satırları ekliyoruz:

```
#20 disk x 100 / 3 replica = 666
osd pool default pg num = 666
osd pool default pgp num = 666
osd pool default size = 3
public network = 172.16.3.0/16
cluster network = 10.10.10.0/24
```

[osd] kısmı altında xfs dosya sistemi yaratılırken ve mount edilirken geçerli olacak argümanları tanımlıyoruz.

```
osd mkfs type = xfs
osd mkfs options xfs = -m crc=1,finobt=1 -f -i size=2048
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 15 / 42
--	---	--

```
osd mount options xfs = rw,noatime,inode64,logbsize=256k,delaylog
```

ceph.conf dosyamızın son hali aşağıdaki şekilde:

```
[global]
fsid = 12bee812-a843-4103-9321-febe8c5108e6
mon initial members = cephm
mon host = 172.16.3.14
auth cluster required = cephx
auth service required = cephx
auth client required = cephx
filestore xattr use omap = true
#20 disk x 100 / 3 replica = 666
osd pool default pg num = 666
osd pool default ppg num = 666
osd pool default size = 3
public network = 172.16.3.0/16
cluster network = 10.10.10.0/24
[osd]
osd mkfs type = xfs
osd mkfs options xfs = -m crc=1,finobt=1 -f -i size=2048
osd mount options xfs = rw,noatime,inode64,logbsize=256k,delaylog
```

Bu aşamadan sonra ceph-deploy ile sunucuların kurulumlarını ve monitör sunucusu tanımlarını yapıyoruz:

```
ceph-deploy install cephm ceph1 ceph2 ceph3 ceph4 ceph5
ceph-deploy mon create-initial
```

Bundan sonraki aşamada sunucular üzerinde osd için kullanacağımız diskleri hazırlıyoruz. Zap kumutu disklerde bulunan partition vs bilgisini siliyor. osd prepare ile disklerin ceph için hazırlanma işlemleri gerçekleştiriliyor. /dev/sde diski journal için kullanılacak olan SSD diski. Bu nedenle osd prepare komutunda hazırlanacak diskten sonra araya : koyarak ekliyoruz.

```
ceph-deploy disk zap ceph1:sdb ceph1:sdc ceph1:sdd ceph1:sde ceph1:sdf
ceph-deploy osd prepare ceph1:sdb:/dev/sde
ceph-deploy osd prepare ceph1:sdc:/dev/sde
ceph-deploy osd prepare ceph1:sdd:/dev/sde
ceph-deploy osd prepare ceph1:sdf:/dev/sde

ceph-deploy disk zap ceph2:sdb ceph2:sdc ceph2:sdd ceph2:sde ceph2:sdf
ceph-deploy osd prepare ceph2:sdb:/dev/sde
ceph-deploy osd prepare ceph2:sdc:/dev/sde
ceph-deploy osd prepare ceph2:sdd:/dev/sde
ceph-deploy osd prepare ceph2:sdf:/dev/sde

ceph-deploy disk zap ceph3:sdb ceph3:sdc ceph3:sdd ceph3:sde ceph3:sdf
ceph-deploy osd prepare ceph3:sdb:/dev/sde
ceph-deploy osd prepare ceph3:sdc:/dev/sde
ceph-deploy osd prepare ceph3:sdd:/dev/sde
ceph-deploy osd prepare ceph3:sdf:/dev/sde

ceph-deploy disk zap ceph4:sdb ceph4:sdc ceph4:sdd ceph4:sde ceph4:sdf
ceph-deploy osd prepare ceph4:sdb:/dev/sde
ceph-deploy osd prepare ceph4:sdc:/dev/sde
ceph-deploy osd prepare ceph4:sdd:/dev/sde
ceph-deploy osd prepare ceph4:sdf:/dev/sde

ceph-deploy disk zap ceph5:sdb ceph5:sdc ceph5:sdd ceph5:sde ceph5:sdf
ceph-deploy osd prepare ceph5:sdb:/dev/sde
ceph-deploy osd prepare ceph5:sdc:/dev/sde
ceph-deploy osd prepare ceph5:sdd:/dev/sde
ceph-deploy osd prepare ceph5:sdf:/dev/sde
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 16 / 42
--	---	--

Son olarak admin nodu' u tanımlı:

```
ceph-deploy admin cephm ceph1 ceph2 ceph3 ceph4 ceph5
sudo chmod +r /etc/ceph/ceph.client.admin.keyring
```

Her şey yolunda giderse:

```
ceph health
```

komutun çıktısı aşağıdaki şekilde olmalı:

```
HEALTH OK
```

Aşağıdaki hatayı almanız pgs sayısını arttırmanız gerektiği anlamına geliyor:

```
HEALTH WARN too few pgs per osd (3 < min 20)
```

Sayıyı arttırmak için aşağıdaki komutu kullanabilirsiniz.

```
ceph osd pool set rbd pgp num 666
ceph osd pool set rbd pg_num 666
```

### 3.6. Ceph Calamari GUI Kurulumu

Calamari Intank tarafınan ceph'in izlenmesi için geliştirilen bir gui uygulaması. [Orjinal kurulum belgesi](#) oldukça karışık, İnternette arama yaparken [bulduğum](#) belge üzerinde Ubuntu 14.04 üzerinden calamari kurulumu gerçekleştirdim. Yazıldığı şekilde çalışmadığı için bir kaç ekleme yapmak gerekti. Kurulum için ilk olarak gerekli olan deb paketlerini hazırlamanız gerekiyor.

#### 3.6.1. Calamari Kurulum için Gerekli Paketler

```
echo "deb http://ppa.launchpad.net/saltstack/salt/ubuntu trusty main"
> /etc/apt/sources.list.d/saltstack.list

wget -q -O- "http://keyserver.ubuntu.com:11371/pks/lookup?op=get&
search=0x4759FA960E27C0A6" | sudo apt-key add -

apt-get update
apt-get install salt-master salt-minion salt-syndic

echo "deb http://apt.postgresql.org/pub/repos/apt/ trusty-pgdg main"
> /etc/apt/sources.list.d/pgdg.list

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc
| sudo apt-key add -

apt-get update && sudo apt-get upgrade
apt-get install -y curl build-essential openssl libssl-dev apache2
libapache2-mod-wsgi libcairo2 supervisor python-cairo libpq5
```



	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 17 / 42
--	---	--

```

postgresql python-m2crypto python-virtualenv git python-dev swig
libzmq-dev g++ postgresql-9.1 postgresql-server-dev-9.1 libcairo2-dev
python-pip libpq-dev ruby debhelper python-mock python-configobj
cdbs gem ruby1.9.1 ruby1.9.1-dev make devscripts
software-properties-common python-support

cd ~
git clone https://github.com/joyent/node.git
cd node
git checkout v0.10.10
./configure
make -j4
make install

mkdir ~/npm
cd ~/npm
wget --no-check-certificate https://npmjs.org/install.sh
sh install.sh

npm install -g bower
npm install -g coffee-script
npm install -g grunt-cli
npm install -g grunt-legacy-util
npm install -g grunt-legacy-log
npm install -g dateformat
npm install -g eventemitter2
npm install -g glob
npm install -g minimatch
npm install -g findup-sync
npm install -g js-yaml
npm install -g rimraf
npm install -g iconv-lite
npm install -g nopt
npm install -g grunt-lib-contrib
npm install -g uglify-js
npm install -g filesize
npm install -g optipng-bin
npm install -g jpegtran-bin
gem install compass
gem install sass

```

### 3.6.2. Calamari Server Kurulumu Paketi Hazırlanması

```

cd ~
git clone https://github.com/ceph/calamari.git
cd calamari
cd debian
mv source source.old
cd ..
dpkg-buildpackage

```

Bu işlemlerden sonra bir alt dizinde aşağıdaki dosyalar oluşacak:

```
calamari-server 1.0.0-1 amd64.deb calamari 1.0.0-1.tar.gz
```

### 3.6.3. Calamari Client Kurulumu Paketi Hazırlanması

(Bu aşamadan sonraki komutları root kullancısı dışında bir kullanıcı ile çalıştırın!!)

```

cd ~
git clone https://github.com/ceph/calamari-clients.git
cd calamari-clients
export NODE_PATH=$NODE_PATH:/usr/local/lib/node_modules"
make build-real

```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 18 / 42
--	---	--

build-real komutundan sonra hata alırsanız baştan başlamak için:

```
make clean-real
```

dist dizini altında yaratılan dosyaları kurulum için bir araya toplayalım:

```
cd calamari-clients/admin/dist
tar -czpvf calamari-admin.tar.gz *
cd ../../login/dist
tar -czpvf calamari-login.tar.gz *
cd ../../manage/dist
tar -czpvf calamari-manage.tar.gz *
cd ../../dashboard/dist
tar -czpvf calamari-dashboard.tar.gz *
cd ../../
find . -name "calamari*.tar.gz" -exec cp {} . \;
```

Bulduğunuz dizinde aşağıdaki dosyaları görüyor olmanız gerekiyor:

```
ls *.gz
calamari-admin.tar.gz calamari-dashboard.tar.gz calamari-login.tar.gz
calamari-manage.tar.gz
```

### 3.6.4. Diamond Kurulumu Paketi Hazırlanması

```
git clone https://github.com/ceph/Diamond.git
cd Diamond
git checkout calamari
dpkg-buildpackage
cd ..
ls diam*
diamond 3.1.0 all.deb diamond 3.1.0 amd64.changes diamond 3.1.0.dsc
diamond_3.1.0.tar.gz
```

Tüm işlemler bitinde yaratılan tüm dosyaları bir dizin altına toplayın. Buraya kadar olan derleme kısmı sonunda elinizde bulunması gereken dosyalar:

```
ceph@cephm:~/CEPH$ ls -al
total 40176
drwxrwxr-x 2 ceph ceph 4096 Apr 3 07:06 .
drwxr-xr-x 15 ceph ceph 4096 Apr 6 09:20 ..
-rwxr-xr-x 1 root root 23263443 Apr 2 09:14 calamari 1.0.0-1.tar.gz
-rwxr-xr-x 1 root root 319104 Apr 2 14:45 calamari-admin.tar.gz
-rwxr-xr-x 1 root root 563767 Apr 2 14:45 calamari-dashboard.tar.gz
-rwxr-xr-x 1 root root 338871 Apr 2 14:45 calamari-login.tar.gz
-rwxr-xr-x 1 root root 491400 Apr 2 14:45 calamari-manage.tar.gz
-rwxr-xr-x 1 root root 11027912 Apr 2 09:21 calamari-server 1.0.0-1 amd64.deb
-rwxr-xr-x 1 ceph ceph 231674 Apr 2 14:46 diamond 3.1.0 all.deb
-rwxr-xr-x 1 ceph ceph 1123 Apr 2 14:46 diamond 3.1.0 amd64.changes
-rwxr-xr-x 1 ceph ceph 711 Apr 2 14:46 diamond 3.1.0.dsc
-rwxr-xr-x 1 ceph ceph 4874357 Apr 2 14:46 diamond_3.1.0.tar.gz
```

### 3.6.5. CALAMARİ Sunucusu Kurulumu

Kurulumu cephm sunucusunda yapım. Kurulum için gerekli olan dosyaları /home/ceph/CEPH dizini altında.

```
mkdir -p /opt/calamari/webapp/content
cd /opt/calamari/webapp/content/
mkdir dashboard
cd dashboard/
tar -xvzf /home/ceph/CEPH/calamari-dashboard.tar.gz
cd ..
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 19 / 42
--	---	--

```
mkdir login
mkdir manage
mkdir admin
cd login
tar -xvzf /home/ceph/CEPH/calamari-login.tar.gz
cd ..
cd manage
tar -xvzf /home/ceph/CEPH/calamari-manage.tar.gz
cd ..
cd admin
tar -xvzf /home/ceph/CEPH/calamari-admin.tar.gz
cd ..
cd /home/ceph/CEPH

aptitude install python-twisted python-txamqp python-greenlet
python-sqlalchemy python-gevent

dpkg -i ./calamari-server_1.0.0-1_amd64.deb
```

En son komutun çıktısı:

```
root@cephm:/home/ceph/CEPH# dpkg -i ./calamari-server 1.0.0-1 amd64.deb
(Reading database ... 100135 files and directories currently installed.)
Preparing to unpack .../calamari-server 1.0.0-1 amd64.deb ...
Unpacking calamari-server (1.0.0-1) over (1.0.0-1) ...
Setting up calamari-server (1.0.0-1) ...
ERROR: Module version does not exist!
salt-master stop/waiting
salt-master: no process found
salt-master start/running, process 52801
Stopping supervisor: supervisord.
Starting supervisor: supervisord.
* Stopping web server apache2 *
* Starting web server apache2 *
Thank you for installing Calamari.

Please run 'sudo calamari-ctl initialize' to complete the installation.
```

Diamond paketini de kuruyoruz:

```
dpkg -i ./diamond_3.1.0_all.deb
```

Salt Minion paketini kurup gerekli ayarları yapıyoruz:

```
apt-get install salt-minion
```

/etc/salt/minion dosyasında master ile başlayan tanıma sunucu kurulumunun yaptığımız sunucunun ismini yazıyoruz:

```
master: cephm
```

diamond ve salt-minion servislerini restart ediyoruz.

```
service diamond restart
service salt-minion restart
```

En son olarak sunucu tarafından calamari programının başlatılması için gerekli komutu yazıyoruz. Bu aşamada size web arayüzü için kullanıcı adı ve parola soracak.

```
calamari-ctl initialize
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 20 / 42
--	---	--

### 3.6.6. CALAMARI İstemcileri Kurulumu

ceph1,ceph2,ceph3,ceph4,ceph5 sunucularına istemci kurulumlarını yapıyoruz.

```
aptitude install python-support
dpkg -i ./diamond 3.1.0 all.deb
cd /etc/diamond/
cp diamond.conf.example diamond.conf
apt-get -y install salt-minion
```

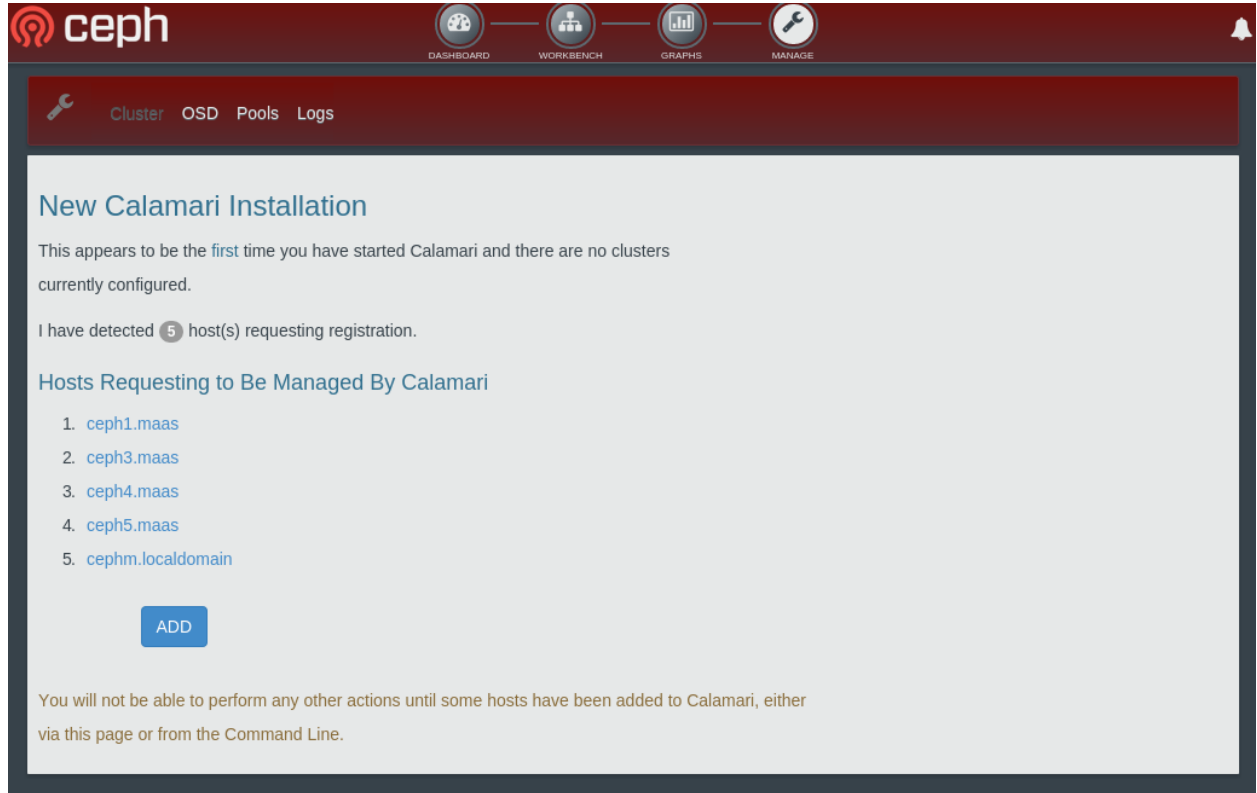
/etc/salt/minion dosyasında master ile başlayan tanıma sunucu kurulumunun yaptığımız sunucunun ismini yazıyoruz.

```
master: cephm
```

diamond ve salt-minion servislerini restart ediyoruz.

```
service diamond restart
service salt-minion restart
```

İstemcilerde salt-minion'u restart ettikten 30sn-1dk içinde sunucu web arayüzünde istemci adı ile birlikte calamari için register edilmeye hazır olarak gözüküyorlar. Ekran görüntüsünü aşağıda:



	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 21 / 42
--	---	--

## 4. Ceph İŞLETİMİ

### 4.1. Ceph Performans Optimizasyonu için Ayarlanabilecek Değişken Listesi

#### 4.1.1. Genel

- **Kernel pid max:** Linux çekirdeği tarafından kullanılacak olan maksimum işlemci numarası (Process id). OSD sayısının fazla olduğu sistemlerde thread sayısında göz önüne alındığının da bu değerin artırılması gerekebilir. Bu değeri yükseltmek için /proc/sys/kernel/pid\_max değeri artırılabilir.
- **max open files:** Ceph tarafından kullanacak olan dosyaların maksimum sayısı.
- **filestore min sync interval, filestore max sync interval:** Bu değerlerin verinin journal'dan hangi sıklıkla diske aktarılacağını belirliyor (saniye olarak). SSD diskiniz büyükse ve değerleri az tutarsanız SSD diskinizi yeterince kullanmamış olursunuz.
- **Jumbo Frames:** anahtarlama cihazı ve işletim sisteminde mtu değeri 9000 olarak ayarlanması performansı arttıracaktır.
- **Disk read\_ahead:**  
Disklerden okuma sırasında okunan verilerin tampon belleğe aktarılması istemcilerin(prefetch) disk erişimini hızlandıracaktır. Diske ait mevcut değer (sde diski için)  
cat /sys/block/sde/vda/queue/read\_ahead\_kb  
komutu ile görülebilir. Değer echo ile değiştirilebilir.

#### 4.1.2. Filestore Kuyruğu

- **filestore queue max ops:** filestore'un kuyruğa yeni bir I/O operasyonu için maksimum sayısı
- **filestore queue max bytes:** I/O operasyonlarının maksimum byte değeri
- **filestore queue committing max ops:** Bir seferde yapılacak olan I/O operasyonların maksimum sayısı
- **filestore queue committing max bytes:** Bir seferde yapılacak olan I/O operasyonların maksimum byte değeri
- **filestore op threads:** Dosya sisteminde yapılacak olan operasyonları için paralel olarak çalıştırılacak thread değeri

#### 4.1.3. OSD

- **osd max write size:** OSD tarafından bir seferde yazılabilecek olan veri miktarı (MByte)
- **osd client message size cap :** İstemciye ait verinin bellek tutulacak kısmına ait olan maksimum değeri (MByte)
- **osd deep scrub stride:** Scrub operasyonları sırasında okunacak veri miktarı (Byte)
- **osd op threads:** Ceph OSD programı tarafından kullanılan thread sayısı
- **osd disk threads:** Ceph OSD programı tarafından disk işlemlerinde kullanılacak olan thread sayısı

#### 4.1.4. OSD Journal

- **journal max write bytes:** Journal'a bir seferde yazılabilecek maksimum byte miktarı
- **journal max write entries:** Journal'a bir seferde yazılabilecek yazma sayısı

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 22 / 42
--	---	--

- **journal queue max ops:** Journal kuyruğunda tek seferde yapılabilecek olan maksimum operasyon sayısı
- **journal queue max bytes:** Journal kuyruğunda tek seferde yapılabilecek olan maksimum byte sayısı

#### 4.1.5. OSD Kurtarma Operasyonları (Recovery)

- **osd recovery op priority:** OSD kurtarma işleminin önceliği, rakam azaldıkça önceliği artıyor.
- **osd recovery op priority:** Aktif kurtarma işlemlerinin maksimum sayısı, sayı arttıkça kurtarma süresi azalıyor, sistemin genel performansı azalıyor.
- **osd max backfills:** Backfill operasyonlarının maksimum sayısı.

#### 4.1.6. İstemci

- **rbd cache:** İstemci tarafında rbd cache özelliğini açmak için (true)
- **rbd cache size:** İstemci tarafında rbd cache değeri (Byte)
- **rbd cache max dirty:** İstemci tarafında tampon bellekte tutulan maksimum veri miktarı. Bu değere erişildiğinde tampon bellekteki veriler flush ediliyor. Bu değer 0 olarak ayarlanır ise tampon bellek write-through olarak yapılandırılır. Atanmış değer write-back.
- **rbd cache max dirty age:** Saniye olarak bellekteki bilgilerin disklere flush edilme süresi

## 4.2. Ceph PG Durumları

Kurduğunuz ceph cluster'ın sağlıklı PG gruplarının durumları ile oldukça ilişkili olduğundan ceph pg stat veya ceph -s

```
root@cephm:/home/ceph# ceph pg stat
v34874: 1332 pgs: 1332 active+clean; 280 GB data, 844 GB used, 8455 GB / 9300 GB avail

root@cephm:/home/ceph# ceph -s
cluster c33eca63-f5b5-4689-9fc5-636782f66f5c
health HEALTH OK
monmap e1: 1 mons at {cephm=172.16.3.14:6789/0}, election epoch 1, quorum 0 cephm
osdmap e1673: 20 osds: 20 up, 20 in
pgmap v34874: 1332 pgs, 2 pools, 280 GB data, 81114 objects
844 GB used, 8455 GB / 9300 GB avail
1332 active+clean
```

komutu ile durumlarını göreceğiniz PG'lerinin durumlarının ne anlama geldiğini bilmek oldukça önemli.

#### 4.2.1. Peering

PG'ler kendi aralarında tutulan objelerin ve metadata bilgilerinin durumu hakkında haberleşiyorlar. Bu durum bittikten sonra (obje ve metadata'nın durumu konusunda karara varıldıktan sonra) Active, Clean, Degraded vb duruma geçiş yapılıyor.

#### 4.2.2. Active

Peerin sonrası herşeyin yolunda gittiği kararı alındı ise PG active duruma geçer, bu durumda veri ana PG'de ve replicalarda I/O operasyonuna yapmak için hazır durumdadır.

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 23 / 42
--	---	--

#### 4.2.3. Clean

Birincil (primary) ve ikincil (secondary) OSD'lerin peeringinde bir sorun yok ise, PG'lerin konumunda herhangi bir değişikliğe gidilmeyecekse (crush map'te değişim sonrası örneğin),objelerin replikasyon sayılarında bir sorun yok ise (örneğin replica count 3 ise 3 kopyada da düzgün şekilde yazılmış ise ) clean durumuna geçilir.

#### 4.2.4. Degraded

Herhangi bir OSD'ye erişilemediğinde, OSD'de bulunan PG'lerin durumu degraded olarak değiştirilir. OSD'ye erişilenememe durumu 300sn boyunca sürerse Ceph recovery işlemini başlatır ve diğer replikalardan obje başka bir OSD'de tekrar oluşturulur. 300sn'den önce OSD geri gelir ise diğer OSD'ler ile peering işlemini başlatır.

#### 4.2.5. Recovering

OSD erişilemez duruma düştüğünde tekrar erişilebilir hale gelirse peering işleminden sonra objelerin güncel halinin diğer OSD'lerden alınması için recovery durumuna geçilir.

#### 4.2.6. Backfilling

Ceph küme'sine yeni bir OSD eklendiğinde, Ceph var diğer OSD'lerde var olan verinin bir kısmını yeni OSD'ye taşıma işlemini başlatır. Bu işleme backfilling denir. Backfiling operasyonu tamamlandığında OSD I/O operasyonlarını yapmak için hazır hale gelmiş olur. Ceph kümesinde yeni bir OSD eklendiğinde ağırlık değerinin (weigh) düşük tutularak azar azar artırılarak kümeye backfilling operasyonları ile dahil edilmesi önerilmektedir.

#### 4.2.7. Remapped

Belirli bir PG'nin replicalarının bulunması gerken OSD'lere acting set ismi verilmektedir. Acting set'te bir değişikli olduğu zaman eski OSD'lerde ki verinin yeni OSD'lere taşınması gerekmektedir. Bu süre boyunca I/O isteklerine eski acting set'teki OSD'ler cevap verir. Verinin taşınması tamamlanınca, I/O operasyonlarına cevap verme görevini yeni OSD'ler üstlenir. Bu süreç içindeki PG'ler remapped durumda görülür.

#### 4.2.8. Stale

Ceph OSD istatistiklerini her 0.5 saniyede monitör sunucusuna iletir. Acting set'teki birincil OSD durumunu monitöre belirmediği anda monitör PG'nin durumuna stale olarak değiştirir.

### 4.3. Ceph Scrub Yapılandırması

Ceph objeler halinde tuttuğu verinin bütünlüğünün korunması amacı ile belirli zamanlarda scrub işlemini gerçekleştiriyor. Bu işlem sırasından objelerdeki bozulan verilerin onarılması amaçlanıyor. Tüm objelerde bu işlemin yapılmasına "deep scrub" deniyor. Atanmış değeri osd deep scrub interval değişkeni ile tanımlanıyor ve süresi haftada bir.Tanımlanabilecek olan scrub değişkenleri:

**osd max scrubs:** Ceph OSD tarafından yapılacak olan maksimum operasyon sayısı (1)

**osd scrub thread timeout:** Scrub timeout (Saniye,60)

**osd scrub load threshold:** load avarage değeri bu değerden büyük olursa scrub yapılmıyor (0.5)

**osd scrub min interval:** Scrub işlemini yapmak için maksimum saniye (Günde bir kez, 60\*60\*24)

**osd scrub max interval:** Yük durumu ne olursa olsun scrub çalıştırmak için gerken maksimum süre.

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 24 / 42
--	---	--

(Haftada bir, 7\*60\*60\*24)

**osd deep scrub interval:** deep scrub yapmak için gerekli olan maksimum süre (Haftada bir, 60\*60\*24\*7)

**osd deep scrub stride:** Scrub yaparken kullanılacak okuma büyüklüğü (524288, 512KB)

Peki elle scrub çalıştırmamız gerektiğinde ne yapacağız. Bunun için:

```
ceph osd scrub {osd-numarası}
```

yazmak yeterli. Osd numarasını görmek için

```
ceph osd dump
```

Komutunu kullanabilirsiniz. Scrub işleminin mevcut cluter'da hangi tarihlerde yapıldığını görmek için:([Referans](#))

```
root@ceph1:/home/ceph# for date in `ceph pg dump | grep active | awk '{print $20}'`; do date +%A
-d $date; done | sort | uniq -c
dumped all in format plain
665 Monday
667 Tuesday
```

Hangi saatlerde yapılacağını görmek için:

```
root@ceph1:/home/ceph# for date in `ceph pg dump | grep active | awk '{print $21}'`; do date +%H
-d $date; done | sort | uniq -c
dumped all in format plain
1 11
666 12
665 13
```

Bu değerlere baktığımızda yoğunluk olması muhtamel Pazartesi/ Salı günleri 12-13 aralığına denk geldiğini görüyoruz. Bu değerleri hafta sonuna çekmek için hafta sonları elle scrub göndermek veya günleri hafta sonuna gelecek şekilde osd deep scrub stride değerini kaydırmak sonra tekrar bir haftaya çekmek çözüm olabilir. İlk olarak scrub ile değişkenler ve atanmış değerleri listeleyelim.

```
ceph@cephm:/var/run/ceph$ sudo ceph --admin-daemon /var/run/ceph/ceph-mon.cephm.asok config show
| grep scrub
"osd scrub thread timeout": "60",
"osd scrub finalize thread timeout": "600",
"osd scrub invalid stats": "true",
"osd max scrubs": "1",
"osd scrub load threshold": "0.5",
"osd scrub min interval": "86400",
"osd scrub max interval": "604800",
"osd scrub chunk min": "5",
"osd scrub chunk max": "25",
"osd scrub sleep": "0",
"osd deep scrub interval": "604800",
"osd deep scrub stride": "524288",
```

Bugün (salı) itibarı ile osd\_deep\_scrub\_interval değerini değiştirdiğimizde günü salı'ya almış oluyoruz. Cumartesi günü yaparsak sorun çözülür

```
ceph@cephm:/var/run/ceph$ sudo ceph --admin-daemon /var/run/ceph/ceph-mon.cephm.asok config set
osd_deep_scrub_interval 604800
```



	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 25 / 42
--	---	--

```
{ "success": "osd deep scrub interval = '604800' "}  
ceph@cephm:/var/run/ceph$ for date in `ceph pg dump | grep active | awk '{print $20}'`; do date  
+%A -d $date; done | sort | uniq -c  
dumped all in format plain  
1332 Tuesday
```

## 5. CEPH YÖNETİMİ

### 5.1. Cluster İşlemleri

#### 5.1.1. Ceph Değişkenlerinin Görüntülenmesi/Değiştirilmesi

Ceph soket arayüzü ile değişkenleri görüntüleyebilir/değiştirebilirsiniz.

```
ceph@cephm:/var/run/ceph$ sudo ceph --admin-daemon /var/run/ceph/ceph-mon.cephm.asok help  
{ "add bootstrap peer hint": "add peer address as potential bootstrap peer for cluster bringup",  
"config diff": "dump diff of current config and default config",  
"config get": "config get <field>: get the config value",  
"config set": "config set <field> <val> [<val> ...]: set a config variable",  
"config show": "dump current config settings",  
"get command descriptions": "list available commands",  
"git version": "get git sha1",  
"help": "list available commands",  
"log dump": "dump recent log entries to log file",  
"log flush": "flush log entries to log file",  
"log reopen": "reopen log file",  
"mon status": "show current monitor status",  
"perf dump": "dump perfcounters value",  
"perf schema": "dump perfcounters schema",  
"quorum enter": "force monitor back into quorum",  
"quorum exit": "force monitor out of the quorum",  
"quorum status": "show current quorum status",  
"sync force": "force sync of and clear monitor store",  
"version": "get ceph version"}
```

Örneğin journal\_aio değişkeninin mevcut durumunu görüntüleyelim:

```
root@cephm:/home/ceph# sudo ceph --admin-daemon /var/run/ceph/ceph-mon.cephm.asok config show |  
grep -i journal aio  
"journal aio": "true",
```

Değeri false olarak değiştirmek için:

```
root@cephm:/home/ceph# sudo ceph --admin-daemon /var/run/ceph/ceph-mon.cephm.asok config set  
journal aio false  
{ "success": "journal aio = 'false' "}
```

Ceph değişkenlerini görüntülemek için alternatif olarak ceph --show-config komutunu kullanabilirsiniz.

```
root@ceph5:/home/ceph# ceph --show-config | more  
name = client.admin  
cluster = ceph  
debug none = 0/5  
debug lockdep = 0/1  
debug context = 0/1  
debug_crush = 1/1
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 26 / 42
--	---	--

```
debug_mds = 1/5
debug_mds_balancer = 1/5
```

## 5.2. Cluster Bakım Modları

Cluster bakım modları listesi:

```
pause|noup|nodown|noout|noin|nobackfill|norecover|noscrub|nodeep-scrub|notieragent
```

ceph clusterınızı bu modlardan herhangi birine sokmak için:

```
ceph osd set [mod ismi]
```

yazmanız yeterli, moddan geri çıkarmak için:

```
ceph osd unset [mod ismi]
```

yazmalısınız.

### 5.2.1. NOIN

Açılıp normal olarak ceph cluster'ına dahil olacak (in) OSD'lerin açıldıkları halde dahil olmamalarını sağlar. Bu modda dikkat edilmesi gereken nokta OSD'leri in olarak işaretlendikten sonra sisteme girmeleri için ayrıca restart edilmelerinin unutulmamasıdır.

```
root@cephm:/home/ceph# ceph osd set noin
root@cephm:/home/ceph# ceph health
HEALTH_WARN noin flag(s) set
cephm:/home/ceph# ceph osd unset noin
```

### 5.2.2. NOOUT

OSD'lerin herhangi bir nedenle (erişilememeleri, kapanmaları) durumunda cluster dışına atılmalarını (out) engeller.

### 5.2.3. NOUP

OSD'lerin monitörler tarafından up olarak işaretlenmesini ekler. Sürekli up/down olan bir OSD'niz var ise durumunu incelemek için kullanabilirsiniz.

### 5.2.4. NODOWN

OSD'lerin monitörler tarafından down olarak işaretlenmesini ekler. Sürekli up/down olan bir OSD'niz var ise durumunu incelemek için kullanabilirsiniz

### 5.2.5. PAUSE

Cluster üzerine yapılan I/O işlemlerinin durdurur

### 5.2.6. NOSCRUB

Cluster üzerindeki OSD'lerde otomatik olarak yapılacak olan scrub işleminin yapılmasının engeller.

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 27 / 42
--	---	--

### 5.2.7. NODEEP-SCRUB

Cluster üzerindeki OSD’lerde otomatik olarak yapılacak olan deep-scrub işleminin yapılmasının engeller.

### 5.2.8. NOBACKFILL

Backfill operasyonlarını engeller. Backfill operasyonları cluster’a yeni bir OSD eklendiğinde var olan verinin otomatik olarak yeni eklenen OSD’ye aktarılması işlemidir.

### 5.2.9. NORECOVER

Cluster’ın recover duruma geçmesini engeller.

## 5.3. Cluster Sağlığı

### 5.3.1. Cluster Sağlığı Görüntüleme

Cluster sağlığını izlemek için temel komut:

```
root@ceph1:/home/ceph# ceph health
HEALTH_WARN 168 pgs stuck inactive; 168 pgs stuck unclean; 1 requests are blocked > 32 sec
```

Bu örnekte cluster pek sağlıklı olmadığı belli, biraz daha detaylı bir bilgi almak için:

```
root@ceph1:/home/ceph# ceph health detail | more
HEALTH_WARN 168 pgs stuck inactive; 168 pgs stuck unclean; 1 requests are blocked > 32 sec; 1
osds have slow requests
pg 1.f0 is stuck inactive since forever, current state creating, last acting []
pg 1.27f is stuck inactive since forever, current state creating, last acting []
pg 1.f1 is stuck inactive since forever, current state creating, last acting []
pg 1.1b5 is stuck inactive since forever, current state creating, last acting []
pg 1.ec is stuck inactive since forever, current state creating, last acting []
```

stuck olan pg’ler’n bir kısmını görmüş olduk. Herşey yolunda ise cluster durumunuz.

```
root@cephm:/home/ceph# ceph health detail
HEALTH_OK
```

olarak gözükecektir.

### 5.3.2. Cluster Üzerinde Yapılan I/O Görüntüleme

ceph -w komutu cluster üzerindeki yapılan işlemler sonucu oluşan I/O miktarı ile birlikte cluster sağlığı ile ilgili bilgileri de görüntülemek için kullanılabilir:

```
root@cephm:/home/ceph# ceph -w
cluster c33eca63-f5b5-4689-9fc5-636782f66f5c
health HEALTH_OK
monmap el: 1 mons at {cephm=172.16.3.14:6789/0}, election epoch 1, quorum 0 cephm
osdmap e9744: 20 osds: 20 up, 20 in
pgmap v200822: 666 pgs, 1 pools, 29608 MB data, 7404 objects
93976 MB used, 9208 GB / 9300 GB avail
666 active+clean
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 28 / 42
--	---	--

### 5.3.3. Cluster Kapasite Kullanımı Görüntüleme

Ne kadar disk kapasitesi kullandığınızı görmek için:

```
root@cephm:/home/ceph# ceph df detail
GLOBAL:
  SIZE          AVAIL          RAW USED          %RAW USED          OBJECTS
  9300G          8994G          305G          3.29          7404
POOLS:
  NAME          ID          CATEGORY          USED          %USED          MAX AVAIL          OBJECTS          DIRTY          READ
WRITE
  rbd          0          -          29608M          0.31          2938G          7404          7404          65
87925
```

### 5.3.4. Depolama Birimi Üzerindeki Disk Yapılandırmasını Görüntüleme

```
root@ceph5:/etc/ceph# ceph-disk list
/dev/sda :
/dev/sda1 other, ext4, mounted on /
/dev/sda2 other, 0x5
/dev/sda5 swap, swap
/dev/sdb :
/dev/sdb1 ceph data, active, cluster ceph, osd.20, journal /dev/sde1
/dev/sdc :
/dev/sdc1 ceph data, active, cluster ceph, osd.21, journal /dev/sde2
/dev/sdd :
/dev/sdd1 ceph data, active, cluster ceph, osd.22, journal /dev/sde3
/dev/sde :
/dev/sde1 ceph journal, for /dev/sdb1
/dev/sde2 ceph journal, for /dev/sdc1
/dev/sde3 ceph journal, for /dev/sdd1
/dev/sde4 ceph journal, for /dev/sdf1
/dev/sdf :
/dev/sdf1 ceph data, active, cluster ceph, osd.23, journal /dev/sde4
/dev/sdg other, unknown
/dev/sdh other, unknown
/dev/sr0 other, unknown
/dev/sr1 other, unknown
```

### 5.3.5. Depolama Birimi Üzerindeki Journal Yapılandırmasını Görüntüleme

Journal için kullandığımız diskin sde olması durumunda, ilk partition bilgilerini görüntülemek için:

```
root@ceph5:/home/ceph# sgdisk --info 1 /dev/sde
Partition GUID code: 45B0969E-9B03-4F30-B4C6-B4B80CEFF106 (Unknown)
Partition unique GUID: FA983B70-C559-4D57-8B33-474538891F1D
First sector: 2048 (at 1024.0 KiB)
Last sector: 51202047 (at 24.4 GiB)
Partition size: 51200000 sectors (24.4 GiB)
Attribute flags: 0000000000000000
Partition name: 'ceph journal'
```

## 5.4. POOL İŞLEMLERİ

### 5.4.1. Pool Yaratma

Deneme ismi altında 4096 PG'den oluşan bir pool yaratalım.

```
root@ceph1:/home/ceph# ceph osd pool create deneme 4096
pool 'deneme' created
```

### 5.4.2. Pool'ları Listeleme

Yarattığımız deneme pool'u ile birlikte var olan diğer pool'ları listeleyelim

```
root@ceph1:/home/ceph# ceph osd lspools
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 29 / 42
--	---	--

```
0 rbd,1 tank,3 deneme,
```

pool bilgilerini biraz daha ayrıntı olarak öğrenmek istersek `osd dump` komutunu kullanıyoruz.

```
ceph@cephm:~$ ceph osd dump | grep 'pool'
pool 0 'rbd' replicated size 3 min size 1 crush ruleset 0 object hash rjenkins pg num 666 pgp num 666 last change 1009 flags hashpspool stripe width 0
pool 1 'tank' replicated size 3 min size 1 crush ruleset 0 object hash rjenkins pg num 666 pgp num 666 last change 998 flags hashpspool stripe width 0
```

Bu komut ile iki adet pool'umuz olduğunu replica ve min\_size değerlerini, pg ve pgp\_num değerlerinin öğrenmiş oluyoruz.

### 5.4.3. Pool Silme

deneme ismindeki pool'u silmek için:

```
root@cephm:/home/ceph# ceph osd pool delete deneme deneme --yes-i-really-really-mean-it
pool 'deneme' removed
```

### 5.4.4. Pool İçindeki Objelerin Listelenmesi

Deneme pool'u içindeki objelerin listesini almak için:

```
root@cephm:~$ rados -p deneme ls | more
benchmark data lufer106 30012 object403
benchmark data lufer106 30012 object331
benchmark data lufer106 30012 object367
benchmark data lufer106 30012 object206
benchmark data lufer106 30012 object372
benchmark data lufer106 30012 object495
benchmark data lufer106 30012 object440
benchmark data lufer106 30012 object365
benchmark data lufer106 30012 object2
benchmark data lufer106 30012 object320
benchmark data lufer106 30012 object286
benchmark data lufer106 30012 object487
benchmark data lufer106 30012 object469
benchmark data lufer106 30012 object301
benchmark_data lufer106_30012_object349
```

### 5.4.5. Pool İçindeki Objelerin Silinmesi

Pool içindeki objeleri isimleri ile silmek için `rm` komutunu kullanıyoruz. `benchmark_data_lufer106_30012_object403` objesinin silmek istediğimizde.

```
rados -p deneme rm benchmark data lufer106 30012 object403
```

Peki birden fazla obje silmek istersek ? for döngüsünü kullanmak en kolay yöntem gibi gözüküyor. deneme pool'u altında bench ile başlayan objeleri silmek için:

```
for i in `rados -p deneme ls | awk '/^bench/ {print $1}` ;do echo $i ; rados -p deneme rm $i ; done
```

Bu komutu çalıştırırsanız işlem sıralı olarak ilerleyecektir. Çok sayıda objeniz var ve paralel şekilde silmek isterseniz silinecek objelerin ismini bir dosyaya yazdırıp (obje\_sil) paralel komutu ile aynı anda birden fazla silme işleminin yapılmasını sağlayabilirsiniz.

```
for i in `rados -p deneme ls | awk '/^bench/ {print $1}` ;do echo $i ; done > obje_sil
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 30 / 42
--	---	--

```
cat objesil | parallel rados -p deneme rm {}
```

#### 5.4.6. Pool Replica Sayısının Değiştirilmesi

Pool replica sayısı objenin kaç adet kopyasının tutulacağı bilgisinin içeriyor. Bu sayı örneğin 3 ise yazılan her obje için iki adet yedek olmak üzere toplam üç adet olacağı anlamına geliyor. Ceph atanmış değer olarak replica sayısını 3 olarak belirlenmiş. Bu değeri deneme pool'u için 1'de değiştirmek isterseniz.

```
ceph osd pool set tank size 1
```

Bu komuttan sonra artık bir adet obje tutulacağı için objenin herhangi bir replicası almayacak.

#### 5.4.7. Pool Min\_size Değerinin Değiştirilmesi

Pool min\_size değeri ceph'in bir obje üzerinde okuma yazma işlemi yapması için o objeden kaç adet olması gerektiğini belirliyor. Örneğin replica adediniz 3 ise ve min\_size değerinizi 2 ise, ceph her objeden üç adet tutuyor ve bu objelerden birine erişilememesi durumunda dahi okuma yazma işlemi yapıyor. Eğer 3 replikadan 2 sini kaybeder ve hala okuma yazma işlemi yapmak isterseniz min\_size değerini aşağıdaki komut ile 1'e indirebilirsiniz. (Özellikle recovery durumda kullanılabilir)

```
ceph osd pool set rbd min size 1
```

## 5.5. OSD İŞLEMLERİ

### 5.5.1. OSD Görüntüleme

#### OSD AĞACINI GÖRÜNTÜLEME

Komut ile hangi OSD'lerin çalıştığını hangilerinde problem olduğunu görebilirsiniz. Aşağıdaki komutun çıktısından görüldüğü üzere ceph5 sunucusundaki osd'ler şu anda çalışmıyor durumdadır.

```
root@cephn:/home/ceph# ceph osd tree
# id      weight  type name      up/down reweight
-1        9      root default
-9         0      rack rack03
-8         0      rack rack02
-7         0      rack rack01
-4         1.8    host ceph3
8          0.45    osd.8   up        1
9          0.45    osd.9   up        1
10         0.45    osd.10  up        1
11         0.45    osd.11  up        1
-2         1.8    host ceph1
0          0.45    osd.0   up        1
1          0.45    osd.1   up        1
2          0.45    osd.2   up        1
3          0.45    osd.3   up        1
-3         1.8    host ceph2
4          0.45    osd.4   up        1
5          0.45    osd.5   up        1
6          0.45    osd.6   up        1
7          0.45    osd.7   up        1
-5         1.8    host ceph4
12         0.45    osd.12  up        1
13         0.45    osd.13  up        1
14         0.45    osd.14  up        1
15         0.45    osd.15  up        1
-6         1.8    host ceph5
20         0.45    osd.20  down      1
21         0.45    osd.21  down      1
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 31 / 42
--	---	--

```
22      0.45      osd.22  down    1
23      0.45      osd.23  down    1
```

## OSD DUMP

ceph osd dump komutu ile osd'ler hakkında ayrıntılı bilgiye sahip olabilirsiniz.

```
root@ceph3:/home/ceph# ceph osd dump | head -12
epoch 2094
fsid c33eca63-f5b5-4689-9fc5-636782f66f5c
created 2015-04-01 13:37:43.305444
modified 2015-04-16 06:54:46.154680
flags noout
pool 0 'rbd' replicated size 3 min size 2 crush ruleset 0 object hash rjenkins pg num 666 pgp num 666 last change 1547 flags hashpspool stripe width 0
pool 1 'tank' replicated size 3 min size 2 crush ruleset 0 object hash rjenkins pg num 666 pgp num 666 last change 1552 flags hashpspool stripe width 0
max osd 24
osd.0 up in weight 1 up from 549 up thru 2092 down at 99 last clean interval [4,98)
172.16.3.3:6800/1853 10.10.10.1:6800/1853 10.10.10.1:6801/1853 172.16.3.3:6801/1853 exists,up
81b2f7f8-a2f2-4890-ad0c-d636bb4a3ca0
osd.1 up in weight 1 up from 548 up thru 2092 down at 99 last clean interval [8,98)
172.16.3.3:6809/1975 10.10.10.1:6806/1975 10.10.10.1:6807/1975 172.16.3.3:6810/1975 exists,up
907e89aa-c539-4932-b29b-185bac625f8d
osd.2 up in weight 1 up from 548 up thru 2092 down at 99 last clean interval [13,98)
172.16.3.3:6803/1858 10.10.10.1:6802/1858 10.10.10.1:6803/1858 172.16.3.3:6804/1858 exists,up
b9feeffe-1890-4cfe-a707-230338fbbb07
osd.3 up in weight 1 up from 548 up thru 2092 down at 99 last clean interval [18,98)
172.16.3.3:6806/1890 10.10.10.1:6804/1890 10.10.10.1:6805/1890 172.16.3.3:6807/1890 exists,up
ecc347c2-4672-4c5a-813e-ecb1c8b4108a
```

### 5.5.2. OSD Başlatma/Durdurma

Başlatma (osd.16 için)

```
start ceph-osd id=16
```

Durdurma

```
stop ceph-osd id=16
```

### 5.5.3. OSD Silme

Silmek istediğimiz OSD'mizin osd.16 olduğunu kabul edelim.İlk olarak osd'yi kapatıyoruz. Daha osd'nin bulunduğu sunucuda dizini umount ediyoruz. Sonra sırası ile crush, auth 'tan çıkarıp en son olarak rm komutu ile siliyoruz. (Bu komutlar osd.16'nın olduğu sunucuda çalıştırılacak)

```
ceph osd out osd.16
stop ceph-osd id=16
umount /var/lib/ceph/osd/ceph-16
ceph osd crush remove osd.16
ceph auth del osd.17
ceph osd rm osd.16
```

### 5.5.4. OSD Config Bilgisini Görüntüleme

OSD'lerin herbirini farklı yapılandırmaya sahip olabilir. Tek bir osd'ye ait olan yapılandırma bilgisini görüntülemek için config show komutunu kullanıyoruz. (Örn osd.16)

```
ceph@ceph1: ceph daemon osd.16 config show
{
  "name": "osd.16",
  "cluster": "ceph",
  "debug none": "0\5",
  "debug lockdep": "0\1",
  "debug_context": "0\1",
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 32 / 42
--	---	--

```
"debug crush": "1\\1",
"debug mds": "1\\5",
"debug mds balancer": "1\\5"
```

### 5.5.5. OSD Benchmark

OSD'ler üzerinde sonra basit olarak bir benchmark yapmak isterseniz:

```
ceph tell osd.N bench [Objeler Sayısı] [Bir seferde yazılacak Byte]
```

Komutunu kullanabilirsiniz. Herhangi bir parametre kullanmazsanız ceph 1Gbyte dosyayı 4MByte'lık obje büyüklüğü ile yazıp sonucu size iletiyor. Yine osd.16'ya test yapmak istersek:

```
root@cephm:/home/ceph# ceph tell osd.16 bench
{ "bytes written": 1073741824,
  "blocksize": 4194304,
  "bytes per sec": "179173531.000000" }
```

Tüm OSD'lere test yapmak için osd.\* yazabilirsiniz.

### 5.5.6. Tek Bir OSD'ye ait Değerleri Değiştirme

OSD'de yapılandırmasında tek bir değişkenin değerini değiştirmek için injectargs argümanını kullanıyoruz. Örneğin osd.16'da yazma işlemlerinde bir sıkıntı var ve bunun nedenini öğrenmek için debug seviyesini arttırmak istediğinizi düşünelim. İlk olarak mevcut değere bakalım.

```
root@ceph1:/home/ceph# ceph daemon osd.16 config show | grep debug osd
"debug osd": "0\\0",
```

Değeri arttırmak için:

```
ceph tell osd.16 injectargs '--debug-osd 0/5'
```

## 5.6. Monitör Sunucusu

### 5.6.1. Monitör Sunucusu Restart Etme

```
root@cephm:~# restart ceph-mon id=cephm
```

### 5.6.2. Monitör İstatistikleri

```
root@cephm:~# ceph mon stat
el: 1 mons at {cephm=172.16.3.14:6789/0}, election epoch 1, quorum 0 ceph
```

### 5.6.3. Quorum Bilgisini Öğrenme

```
root@cephm:~# ceph quorum status
{"election_epoch":1,"quorum":[0],"quorum_names":["cephm"],"quorum_leader_n
```



	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 33 / 42
--	---	--

## 5.7. RBD (Rados BLOCK DEVICE) İşlemleri

### 5.7.1. RBD Yaratma

Deneme pool'u içinde 3.000.000 MByte'lık bir rbd oluşturalım.

```
rd create data --size 3000 000 --pool deneme
```

### 5.7.2. RBD Bilgi Alma (RBD info)

Pool içindeki rbd'ları görmek için rbd ls komutunu kullanıyoruz:

```
root@cephm:/home/ceph# rbd ls deneme
data
```

Daha ayrıntılı bir bilgi almak isterseniz:

```
root@cephm:/home/ceph# rbd --image data -p deneme info
rbd image 'data':
  size 2929 GB in 750000 objects
  order 22 (4096 kB objects)
  block name prefix: rb.0.99d69.2ae8944a
  format: 1
```

### 5.7.3. RBD İstemcilere MAP Edilmesi

Deneme altında yarattığımız data rbd'ı kullanmadan önce istemciye map etmemiz gerekiyor bunun için:

```
root@client:~# rbd map data --pool deneme --name client.admin
/dev/rbd1
```

### 5.7.4. İstemcilere MAP edilen RBD'lerin Görüntülenmesi

Map edilen rbd'lerin görüntülenmesi için rbd showmapped komutu kullanabilir.

```
root@client:~# rbd showmapped
id pool image snap device
1 deneme data - /dev/rbd1
```

### 5.7.5. İstemcilere MAP edilen RBD'lerin Unmap Edilmesi

İlk olarak map ettiğiniz rbd mount edilmiş ise umount etmelisiniz. Ardından rbd unmapped komut ile map edilen rbd'yi kaldırabilirsiniz.

```
root@client:~# rbd unmap /dev/rbd1
```

### 5.7.6. RBD Silinmesi

RBD silmek için rbd'nin herhangi bir istemci tarafından map edilmediğine emin olun. Herhangi bir istemciye map edilmiş bir rbd'yi silmek isterseniz aşağıdaki hatayı alırsınız.

```
root@cephm:~# rbd rm deneme/data
2015-04-27 15:25:42.573852 7f2a9ac5f840 -1 librbd: image has watchers - not removing
Removing image: 0% complete...failed.
rbd: error: image still has watchers
This means the image is still open or the client using it crashed. Try again after
closing/unmapping it or waiting 30s for the crashed client to timeout.
```

Bu durumda silme için ilk olarak hangi istemcinin rbd'yi map ettiğine bakalım.

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 34 / 42
--	---	--

```
root@cephm:/home/ceph# rados listwatchers --pool deneme data.rbd
watcher=172.16.2.106:0/3585843564 client.630175 cookie=1
```

İstemciden rbd'yi unmap ettikten sonra komutu tekrar çalıştırırsak:

```
root@cephm:/home/ceph# rbd rm deneme/data
Removing image: 100% complete...done.
```

## 5.8. CEPH Yazılımının Güncellenmesi

### 5.8.1. Giant (0.87.1 -> 0.87.2) Güncellemsi

#### *Mevcut Sürüm Bilgisinin Öğrenilmesi*

0.87.2 Sürümüne yükseltmek için ilk olarak hangi sürümü kullandığımıza bakalım:

```
/home/ceph# ceph --version
ceph version 0.87.1 (283c2e7cfa2457799f534744d7d549f83ea1335e)
```

Ceph yazılımını sürümünü 0.87.1'den 0.87.2'ye yükseltme işleminin gerçekleştireceğiz

#### *İşletim Sistemi Güncelleştirmelerinin Yapılması*

Ceph Upgrade işlemine başlamadan tüm sunucularda işletim sisteminin güncelleştirmelerinde yapıp kernel update'i varsa sistemi restart ettikten sonra ceph-ceploy'un en son sürümünü kullanıp kullanmadığımıza emin oluyoruz:

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

#### *Monitör Sunucusu Güncellenmesi*

```
root@cephm: ceph-deploy install --release giant
```

Güncelleme işleminin cephm monitör nodu için yapıldığından emin olalım:

```
root@cephm:/home/ceph# ceph --version
ceph version 0.87.2 (87a7cec9ab11c677de2ab23a7668a77d2f5b955e)
```

Monitör sunucusunu ceph servisini restart edelim:

```
restart ceph-mon id=cephm
ceph-mon (ceph/cephm) start/running, process 11760
```

#### *Depolama Sunucuları Güncellenmesi*

Güncelleştirmeler için iki yöntem izlenebilir. İlki güncelleştirme işleminin işletim sistemi paket yöneticisine bırakmak ve diğer işletim sistemi güncelleştirmeleri ile birlikte ceph paketlerindeki güncellenmesini sağlamaktır. İkinci yöntem ceph-deploy aracını kullanmaktır. Güncelleştirmelerde sunucular up/down olurken osd'lerin cluster dışına atılmaması için cluter'ın noout moduna geçiriyoruz. Daha sonra upgrade işlemlerini yapıyoruz.

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 35 / 42
--	---	--

```
ceph@cephm: ceph osd set noout
ceph@cephm: ceph-deploy install --release giant ceph1
ceph@cephm: ceph-deploy install --release giant ceph2
ceph@cephm: ceph-deploy install --release giant ceph3
ceph@cephm: ceph-deploy install --release giant ceph4
ceph@cephm: ceph-deploy install --release giant ceph5
```

Bu işlem bittikten sonra her bir node'taki osd'leri restart ediyoruz. Örneğin ceph1 üzerinde bulunan osd.16,osd.17,osd.18,osd.19 restart edelim.

```
ceph@cephm: ssh ceph@ceph1 sudo restart ceph-osd id=16
ceph@cephm: ssh ceph@ceph1 sudo restart ceph-osd id=17
ceph@cephm: ssh ceph@ceph1 sudo restart ceph-osd id=18
ceph@cephm: ssh ceph@ceph1 sudo restart ceph-osd id=19
```

### 5.8.2. Giant (0.87.2) -> Hammer (0.94.1) Güncellemesi

İlk kısımda anlattığımız adımlar burada da geçerli, işletim sistemi güncelleştirmelerin yapıldığından emin olun, ceph-deploy'un en son sürümüne güncelleyin.

#### Monitör Sunucusu

```
ceph@cephm:~/bulut$ ceph-deploy install --release hammer
cephmceph@cephm:~/bulut$ sudo restart ceph-mon id=cephm
```

#### Depolama Sunucuları

```
ceph osd set noout
ceph@cephm: ceph-deploy install --release hammer ceph1
ceph@cephm: ssh ceph@ceph1 sudo restart ceph-osd id=16
ceph@cephm: ssh ceph@ceph1 sudo restart ceph-osd id=17
ceph@cephm: ssh ceph@ceph1 sudo restart ceph-osd id=18
ceph@cephm: ssh ceph@ceph1 sudo restart ceph-osd id=19

ceph@cephm: ceph-deploy install --release hammer ceph2
ceph@cephm: ssh ceph@ceph2 sudo restart ceph-osd id=4
ceph@cephm: ssh ceph@ceph2 sudo restart ceph-osd id=5
ceph@cephm: ssh ceph@ceph2 sudo restart ceph-osd id=6
ceph@cephm: ssh ceph@ceph2 sudo restart ceph-osd id=7

ceph@cephm: ceph-deploy install --release hammer ceph3
ceph@cephm: ssh ceph@ceph3 sudo restart ceph-osd id=8
ceph@cephm: ssh ceph@ceph3 sudo restart ceph-osd id=9
ceph@cephm: ssh ceph@ceph3 sudo restart ceph-osd id=10
ceph@cephm: ssh ceph@ceph3 sudo restart ceph-osd id=11

ceph@cephm: ceph-deploy install --release hammer ceph4
ceph@cephm: ssh ceph@ceph4 sudo restart ceph-osd id=12
ceph@cephm: ssh ceph@ceph4 sudo restart ceph-osd id=13
ceph@cephm: ssh ceph@ceph4 sudo restart ceph-osd id=14
ceph@cephm: ssh ceph@ceph4 sudo restart ceph-osd id=15

ceph@cephm: ceph-deploy install --release hammer ceph5
ceph@cephm: ssh ceph@ceph5 sudo restart ceph-osd id=0
ceph@cephm: ssh ceph@ceph5 sudo restart ceph-osd id=1
ceph@cephm: ssh ceph@ceph5 sudo restart ceph-osd id=2
ceph@cephm: ssh ceph@ceph5 sudo restart ceph-osd id=3

ceph osd unset noout
```

#### Alınan Hatalar

```
ceph@cephm:~/bulut$ ceph health
HEALTH_WARN too many PGs per OSD (714 > max 300); pool deneme pg_num 4096 > pgp_num 1024
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 36 / 42
--	---	--

Sorunun çözümü için pg sayısının azaltılması gerekiyor. PG sayısını arttırmak mümkün fakat ceph pg sayısının azaltılmasının henüz desteklemiyor. (PG merging). Bu nedenle hata veren pool'u silmek dışında yapacak bir şey yok. PG sayısını ilk başta düzgün olarak ayarlamak gerekiyor(muş)

### *İstemcilerin Güncellenmesi*

İstemciler en son olarak güncellenebilir. Client1 istemcisi için ilk olarak en son giant sürümüne daha sonra hammer sürümüne upgrade ediyoruz.

```
ceph@cephm:~/bulut: ceph-deploy install --release giant client1
ceph@cephm:~/bulut: ceph-deploy install --release hammer client1
```

## 6. CEPH TESTLERİ

### 6.1. Test İçin Kullanılan Programlar

#### 6.1.1. Fio

#### *Fio Programı Argümanları*

Fio kullanılabilecek olan disk bechmark yazılımlarından birisi, ilk olarak hangi argümanlar ile çağrılabilceğine bakalım:

- runtime= Saniye olarak programın ne kadar çalıştırılacağı
- max-jobs =Açılacak maksimum program/iş parçacığı (process/thread) sayısı
- numjobs= Aynı workload üzerinde çalışacak iş sayısı
- ramp\_time = Saniye olarak istatistikler alınmaya başlamadan geçmesi gereken süre . Bu süreyi programın tampon bellekleri doldurması veya stabil sonuçlar almak için kullanabilirsiniz. Bu değeri 15 runtime süresini 60 olarak ayarlarsınız program toplam 75 saniye çalışacak 15 saniye sonra istatistikleri yazmaya başlayacaktır.
- directory = Test dosyalarının yaratılacağı dizinin adı
- name= Test dosyasın adı
- bs= blok büyüklüğü (4k, 8k vb)
- size = test için yaratılacak olan dosyanın büyüklüğü, test normalde bu dosya büyüklüne erişilinceye kadar devam eder. runtime ile verilen çalışma süresine erişilir ise dosya büyüklüğü değerine erişilememiş olsa dahi program sonlandırılır.
- time\_based = size argümanında tanımlanan dosya boyutuna erişilse bile testler runtime argümanında tanımlanan süre boyunca devam eder
- group\_reporting = raporlar grup olarak üretilir. (max\_jobs ile tanımlanan her iş için ayrı rapor üretilmez)
- ioengine= I/O işlemi nasıl yapılacağını tanımları (sync vb)
- direct = O\_DIRECT sistem çağrısı ( non-buffered I/O) ile çekirdek page tablosu (kernel page cache) kullanılmaması
- sync: O\_DSYNC çağrılarak I/O yazma işlemi tamamlandığından emin olması.
- iodepth= Tek bir dosya üzerinde iş yapan iş parçacıklarının sayısı. Yazma işlemi sync olarak yapan

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 37 / 42
--	---	--

ioengine'ler için arttırmanın bir değer artışına neden olma ihtimali düşük. Async olarak yapan ioengine'ler için arttırılması denenebilir.

**-rw=** Yapılacak olan I/O'nun çeşidi

- read = Sıralı okuma
- write = Sıralı yazma
- trim = Sıralı trim (linux)
- randread = rastgele okuma
- randwrite = rastgele yazma
- randtrim = rastgele trim
- rw = rastgele rw
- randrw = rastgele okuma/yazma

**-rwmixread** = rastgele yapılacak olan okuma/yazma testlerinde okuma oranı (atanmış değeri 50:50)

**-rwmixwrite** = rastgele yapılacak olan okuma/yazma testlerinde yazma oranı (atanmış değeri 50:50)

**-output** = test sonuçlarının yazılacağı dosya adı

**-write\_bw\_log** = Benchmark sırasında bantgenişliği değerlerinin yazılacağı dosyanın ismi. Genellikle grafik çizme amaçlı olarak kullanılır.

**-write\_iops\_log** = Benchmark sırasında bantgenişliği değerlerinin yazılacağı dosyanın ismi. Genellikle grafik çizme amaçlı olarak kullanılır.

**-write\_lat\_log** = I/O Operasyonlarının ne kadar zamanda bitirildiğinin yazılacağı dosyanın adı.

**-cpuload** = fio programının işlemci kullanımını verilen değeri maksimum yüzde kullanacak şekilde ayarlar. (1-100 arası değer alır)

### Örnek Kullanım

Test ederken numjobs,iodepth değerlerini 1 olarak başlayıp I/O satüre olacak değlere kadar arttırmanızı tavsiye ederim.

Ceph benchmarkları sırasında 128k blok büyüklüğü için için kullandığım fio komutu. numjobs değeri istemcinin yaratabileceği maksimum I/O miktarı yaratması için 128 olarak ayarlandı. Bu değer muhtamalen 128'e ulaşmadan I/O satüre olacaktır. 1'den başlayarak arttırarak sizin sisteminiz için hangi değerde satüre olacağını deneyebilirsiniz.

```
/usr/local/bin/fio --directory=/ceph/ --time based --direct=1 --rw=write --bs=128k --size=20G --numjobs=128 --runtime=240 --group_reporting --name testfile --output=ceph128kw.txt
```

fio sonuçlarının daha anlamlı olarak görebilmek için [fioparse.sh](http://fioparse.sh) betiğini kullanıyoruz. Github'tan indirip kurabilirsiniz. 1Gbps uplink'olan sunucudan

test	users	size	MB	ms	min	max	std	IOPS	50us	1ms	4ms	10ms	20ms
50ms	.1s	1s	2s	2s+									
writel28	128K	w	101.428	157.690	6.000	5374.000	166.140	811,				1	11
17	12	56	0	0									

10Gbps uplink olan sunucudan

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 38 / 42
--	---	--

test	users	size	MB	ms	min	max	std	IOPS	50us	1ms	4ms	10ms	20ms
50ms	.1s	1s	2s	2s+									
write128	128K	w	121.529	131.200	6.000	2124.000	134.880	972,				3	27
11	3	53	0	0									

### Genfio ile İş Dosyası Oluşturma

Benchmark işine başladığınızda testiniizi farklı okuma örüntüleri (okuma,yazma, rastgele yazma vb) blok büyüklükleri için tektar etmeniz gerekecektir. fio Bu iş için tek tek komut yazmak yerine yapılacak testleri bir dosyaya yazıp (job file) daha sonra oradan okumanıza izin veriyor.

Çok fazla parametre test edeceksiniz iş dosyasının yazımında ayrıca bir iş yükü olduğundan bu işi halletmek için [genfio](#) betiği kullanıyoruz. Genfio fio kaynak kodlarını github'dan indirdiğinizde tools dizini altında geliyor. İlk olarak genfio betiğinin alabileceği argümanlara bakalım:

- c = Test ederken tampo bellek kullanını açıyor. Atanmış değeri kapalı.
  - a = Testleri sıralı olarak yapıyor. Atanmış değeri kapalı.
  - s = Testleri disk bazında sıralı olarak yapmak için. Atanmış değeri kapalı.
  - p = Testleri tüm diskler için birbiri ardında çalışacak şekilde yapmak için. Atanmış değeri açık.
  - d = Disk (/dev/sda vb)
  - r = Programın koşma süresi
  - b = Test blok büyüklüğü
  - m = I/O örüntüsü (fio'da -rw ile tanımlananlardan birisi, read, write vb)
  - x prefix = Yaratılacak iş dosyası için kullanılacak ön ek.
  - A = fio işleri çalıştıktan sonra çalıştırılacak sistem komutları. (Örn: Dosya silme)
  - B = fio işleri çalıştırılmadan önce çalıştırılacak sistem komutları (Örn: disk mount etme)
- Yukarıda elle çalıştırdığımız komut bir iş dosyası oluşturarak genfio ile oluşturmak için:

```
genfio -m write -b 128k -d /ceph -D 1 -p
```

Yazamız yereli. Komut sunucunun ismi ile başlayan ve .fio ile biten bir iş dosyası oluşturuyor: fio ile dosyayı çağırmadan önce kırmızı ile işaretli kısımları değiştirip:

```
[global]
ioengine=libaio
invalidate=1
ramp time=5
iodepth=1
runtime=300
time based
direct=1

[write-ceph-128k-para]
bs=128k
stonewall
filename=/ceph
rw=write
write bw log=lufer106-128k-ceph-write-para.results
write iops log=lufer106-128k-ceph-write-para.results
```

Mavi ile işaretli kısmı ekliyoruz

```
[global]
ioengine=sync
invalidate=1
ramp time=5
iodepth=1
runtime=300
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 39 / 42
--	---	--

```
time based
direct=1
#elle ekle
size=20G

[write-ceph-128k-para]
bs=128k
stonewall
#filename Directory olarak degistir
directory=/ceph
rw=write
write bw log=lufer106-128k-ceph-write-para.results
write iops log=lufer106-128k-ceph-write-para.results
```

programı çalıştırmak için:

```
fio <is_dosyasi_adi.fio>
```

yazmak yeterli.

### *Fio Sonuçlarının Grafik Haline Getirilmesi*

write\_bw\_log, write\_iops\_log, write\_cal\_log komutları ile yazdırdığımız test sonuçlarını fio2gnuplot yazılımı kullanıyoruz. Tüm sonuçlar için grafik çizdirmek için:

```
fio2gnuplot -p "**bw.log*" -g
fio2gnuplot -p "**iops.log*" -g
fio2gnuplot -p "**lat.log*" -g
```

komutlarını yazmak yeterli. fio2gnuplot komutunu kullanmadan önce gnuplot yazılımının sistemde kurulu olduğundan emin olun. Grafiklere başlık yazdırmak istiyorsanız -t aldığınız veri setinin bellir bir saniyesinden sonra grafik oluşturulmasını istiyorsanız -m argümanlarını kullanabilirsiniz.

## **6.1.2. Rados Bench**

### *Rados Bench ile Ceph Benchmark*

Rados bench kurduğunuz ceph cluster'ın okuma ve yazma testlerini gerçekleştirmeniz için ceph ile birlikte gelen araçlardan birisi. Alacağı argüman listesi:

```
rados bench -p <pool ismi> <test süresi saniye > -b <obje büyüklüğü> -t <iş parçacığı adedi>
```

herhangi bir değer vermez iseniz obje büyüklüğünü 4Mbyte, iş parçası büyüklüğünü 16 olarak kabul ediyor. Okuma testi yapmak için ilk olarak --no-cleanup argümanı ile yazma testi yapmanız gerekiyor. Tank isimindeki bir pool'da 240 saniye süreli yazma testi yapmak için:

```
rados bench -p tank 240 write --no-cleanup
```

yazmak yeterli.

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 40 / 42
--	---	--

## 7. CEPH Betikleri

### 7.1. Rados Benchmark Betiği

```
#!/bin/bash
SANIYE=60
POOL=tank
THREAD=16
for i in 4K 8K 16K 32K 64K 128K 512k 1M 4M
do
    #Islem yapmadan page cahe drop et!
    echo 3 > /proc/sys/vm/drop caches
    #Yazilacak birsey varsa sync et!
    sync

    for k in write seq rand
    do
        echo "rados bench -p $POOL $SANIYE -b $i -t $THREAD $k --no-cleanup >
$i.$k.txt"
        rados bench -p $POOL $SANIYE -b $i -t $THREAD $k --no-cleanup > $i.$k.txt
    done
done
```

### 7.2. Fio Test Betiği

#### 7.2.1. Raw Device Test

```
#!/bin/bash
RUN TIME=300
RAMP TIME=240
IO DEPTH=32
TEST=write
RBD DEVICE=/dev/rbd1

for i in 4k 8k 16k 64k 128k 256k 512k 1M 4M
do
    /usr/local/bin/fio --filename=$RBD DEVICE --time based --invalidate=0 --time based --direct=1 --
iodepth=$IO DEPTH --ioengine=rbd --rbd
name=data --pool=tank --clientname=admin --rw=$TEST --bs=$i --size=16G --numjobs=48 --
runtime=$RUN TIME --ramp time=$RAMP TIME --gro
up reporting --name=test --output=ceph.$i.wr.txttxt
done
```

#### 7.2.2. Fio rbd Engine ile Test

```
[global]
ioengine=rbd
clientname=admin
pool=tank
rbdname=data
invalidate=0 # mandatory
direct=1
#sync=1
size=8G
numjobs=1
rw=write
group reporting
ramp time=15
#size'a eriste bile devam et
time based
#new group
runtime=90
```



	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 41 / 42
--	---	--

```
[rbd 4k]
bs=4k
iodepth=32
stonewall
write bw log=4k.bw.wr.txt
write iops log=4k.io.wr.txt
write lat log=4k.lat.wr.txt

[rbd 64k]
bs=64k
iodepth=32
stonewall
write bw log=64k.bw.wr.txt
write iops log=64k.io.wr.txt
write lat log=64k.lat.wr.txt

[rbd 512k]
bs=512k
iodepth=32
stonewall
write bw log=512k.bw.wr.txt
write iops log=512k.io.wr.txt
write_lat_log=512k.lat.wr.txt
```

### 7.2.3. Dosya Sistemi Testi

```
[global]
ioengine=libaio
ramp time=5
iodepth=32
runtime=90
time based
direct=1
#elle ekle
size=1G

[4k]
bs=4k
stonewall
#filename Directory olarak degistir
directory=/mnt
rw=write
write bw log=4k.bw.wr.txt
write iops log=4k.io.wr.txt
write lat log=4k.lat.wr.txt
```

## 7.3. Ceph Data Node Yaratma Betiği

```
#!/bin/bash
echo $1
if [ -z "$1" ]
then
echo "kullanım $0 ceph1"
else
ceph-deploy disk zap $1:sdb $1:sdc $1:sdd $1:sde $1:sdf
ceph-deploy osd prepare $1:sdb:/dev/sde
ceph-deploy osd prepare $1:sdc:/dev/sde
ceph-deploy osd prepare $1:sdd:/dev/sde
ceph-deploy osd prepare $1:sdf:/dev/sde
ceph-deploy admin $i
fi
```

## 7.4. Ceph OSD Silme Betiği

```
#!/bin/bash
for i in 16 17 18 19
```

	<b>CEPH KURLUMU VE İŞLETİMİ EL KİTABI</b>	Sürüm : 1.0 Yayın Tarihi : 29.04.2015 Sayfa No : 42 / 42
--	---	--

```
do
ceph osd out osd.$i
stop ceph-osd id=$i
umount /var/lib/ceph/osd/ceph-$i
ceph osd crush remove osd.$i
ceph auth del osd.$i
ceph osd rm osd.$i
done
```