Creating separate HDD and SSD pools with Ceph Mimic

Alan Johnson February 2019

Introduction

This document describes how to add two separate pools based on device type – HDD and SSD. Typically, the HDD pool is used for capacity-oriented applications with the SSD pool used for lower capacity, performance-oriented applications. This document assumes an already deployed Ceph installation. Although the instructions below are using virtual machines they can be readily adapted for physical machines.

Software Environment

- CentOS 7.6
- Ceph Release Mimic

Hardware summary

There are 4 nodes in total – 1 MON node and 3 OSD nodes. All nodes are configured as Oracle VirtualBox VMs as the intent is to describe the methodology rather than to implement an actual production deployment. Each node has two network interfaces – ens33 for Mgmt and ens34 for use as a Ceph Public network.

The following table shows the hostnames and network addresses used in the testing. All networks are 24 bit.

Table 1 Configuration node names and IPs during testing

Host name	*Primary role*	*Mgmt IP*	*Ceph Public IP*
osd0	OSD	DHCP	10.10.10.31
osd1	OSD	DHCP	10.10.10.32
osd2	OSD	DHCP	10.10.10.33
mon0	Mon	DHCP	10.10.10.30

The node configuration is shown below, there is a total of 9 HDDs and 6 SSDs available to the cluster.

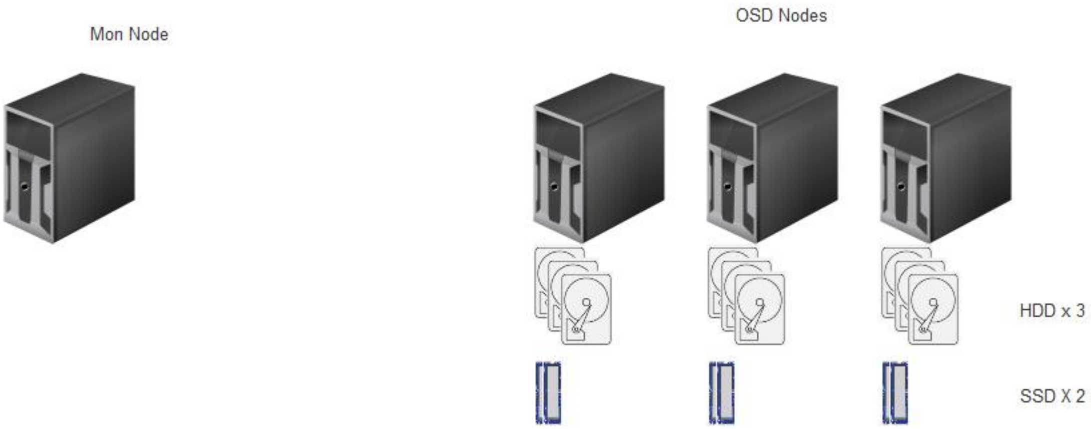


Figure 1 Node layout

OSD Preparation

Verify the disks on each node

```
[cephuser@osd0 ~]$ lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda               8:0    0   20G  0 disk
├─sda1            8:1    0    1G  0 part /boot
└─sda2            8:2    0   19G  0 part
  ├─centos-root 253:0    0   17G  0 lvm  /
  └─centos-swap 253:1    0    2G  0 lvm  [SWAP]
sdb               8:16   0   30G  0 disk
sdc               8:32   0   30G  0 disk
sdd               8:48   0   30G  0 disk
sr0              11:0    1 1024M  0 rom
nvme0n1         259:0    0   20G  0 disk
nvme0n2         259:1    0   20G  0 disk
[cephuser@osd0 ~]$
```

Repeat for nodes osd1 and osd2

The next step is to create the OSDs on the HDD and NVMe devices.

First clear the HDDs on nodes osd0, osd1 and osd2

```
for i in {b..d}; do ceph-deploy disk zap osd0 /dev/sd$i; done
for i in {b..d}; do ceph-deploy disk zap osd1 /dev/sd$i; done
for i in {b..d}; do ceph-deploy disk zap osd2 /dev/sd$i; done
```

Then zap the NVMe devices

```
for i in {1..2}; do ceph-deploy disk zap osd0 /dev/nvme0n$i; done
for i in {1..2}; do ceph-deploy disk zap osd1 /dev/nvme0n$i; done
for i in {1..2}; do ceph-deploy disk zap osd2 /dev/nvme0n$i; done
```

Next create the OSDs

```
for i in {b..d}; do ceph-deploy osd create osd0 --data /dev/sd$i; done
for i in {b..d}; do ceph-deploy osd create osd1 --data /dev/sd$i; done
for i in {b..d}; do ceph-deploy osd create osd2 --data /dev/sd$i; done
for i in {1..2}; do ceph-deploy osd create osd0 --data /dev/nvme0n$i; done
for i in {1..2}; do ceph-deploy osd create osd1 --data /dev/nvme0n$i; done
for i in {1..2}; do ceph-deploy osd create osd2 --data /dev/nvme0n$i; done
```

Checking the OSD tree shows the device class has been set as HDD for the HDDs and SSD for the NVME devices.

```
[cephuser@mon0 ~]$ ceph osd tree
ID CLASS WEIGHT  TYPE NAME      STATUS REWEIGHT PRI-AFF
-1       0.38058 root default
-3       0.12686     host osd0
 0   hdd 0.02930         osd.0      up  1.00000 1.00000
 1   hdd 0.02930         osd.1      up  1.00000 1.00000
 2   hdd 0.02930         osd.2      up  1.00000 1.00000
 9   ssd 0.01949         osd.9      up  1.00000 1.00000
10   ssd 0.01949         osd.10     up  1.00000 1.00000
-5       0.12686     host osd1
 3   hdd 0.02930         osd.3      up  1.00000 1.00000
 4   hdd 0.02930         osd.4      up  1.00000 1.00000
 5   hdd 0.02930         osd.5      up  1.00000 1.00000
11   ssd 0.01949         osd.11     up  1.00000 1.00000
12   ssd 0.01949         osd.12     up  1.00000 1.00000
-7       0.12686     host osd2
 6   hdd 0.02930         osd.6      up  1.00000 1.00000
 7   hdd 0.02930         osd.7      up  1.00000 1.00000
 8   hdd 0.02930         osd.8      up  1.00000 1.00000
13   ssd 0.01949         osd.13     up  1.00000 1.00000
14   ssd 0.01949         osd.14     up  1.00000 1.00000
[cephuser@mon0 ~]$
```

OSDs 0 through 8 are HDD based ad OSDs 9 through 14 are SSD based.

Creating the rulesets

Now that the class of device has been correctly recognized, a new ruleset must be created. The format is `ceph osd crush rule create-replicated <rulesetname> default <failure-domain> <class>.` So the replicated rule for SSD classes using host as the failure domain is:

```
$ ceph osd crush rule create-replicated highspeedpool default host ssd
```

and for HDD classes

```
$ ceph osd crush rule create-replicated highcapacitypool default host hdd
```

Showing the new rules

The new rules can be shown with -

```
ceph osd crush rule dump
[
    {
        "rule_id": 0,
        "rule_name": "replicated_rule",
        "ruleset": 0,
        "type": 1,
        "min_size": 1,
        "max_size": 10,
        "steps": [
            {
                "op": "take",
                "item": -1,
                "item_name": "default"
            },
            {
                "op": "chooseleaf_firstn",
                "num": 0,
                "type": "host"
            },
            {
                "op": "emit"
            }
        ]
    },
    {
        "rule_id": 1,
        "rule_name": "highspeedpool",
        "ruleset": 1,
        "type": 1,
        "min_size": 1,
        "max_size": 10,
        "steps": [
            {
                "op": "take",
                "item": -12,
                "item_name": "default~ssd"
            },
            {
                "op": "chooseleaf_firstn",
                "num": 0,
                "type": "host"
            },
            {
                "op": "emit"
            }
        ]
    },
    {
        "rule_id": 2,
        "rule_name": "highcapacitypool",
        "ruleset": 2,
        "type": 1,
        "min_size": 1,
        "max_size": 10,
        "steps": [
            {
                "op": "take",
                "item": -2,
                "item_name": "default~hdd"
            },
            {
                "op": "chooseleaf_firstn",
                "num": 0,
                "type": "host"
            },
            {
                "op": "emit"
            }
        ]
    }
]
```

To show the device classes –

```
$ ceph osd crush class ls
[
    "hdd",
    "ssd"
]
```

Creating the pools with the new ruleset

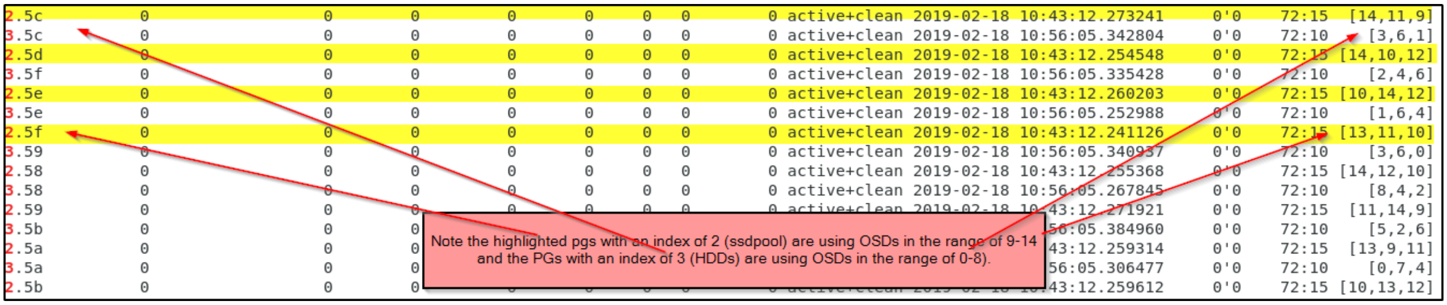
```
ceph osd pool create ssdpool 128 128 highspeedpool
ceph osd pool create hddpool 256 256 highcapacitypool
```

Showing the pools

```
[cephuser@mon0 ~]$ ceph osd pool ls detail
pool 2 'ssdpool' replicated size 3 min_size 2 crush_rule 1 object_hash rjenkins pg_num 128 pgp_num
128 last_change 66 flags hashpspool stripe_width 0
pool 3 'hddpool' replicated size 3 min_size 2 crush_rule 2 object_hash rjenkins pg_num 256 pgp_num
256 last_change 71 flags hashpspool stripe_width 0
```

Now if all is correct then the NVMe devices should belong to the pool with an index of 2 (ssdpool) and the HDD devices should belong to the pool with an index of 3 (hddpool). We can check this by looking at the output of `pg dump` which shows which OSDs are associated with which pool.

```
[cephuser@mon0 ~]$ ceph pg dump | grep "^[2-3]"
```



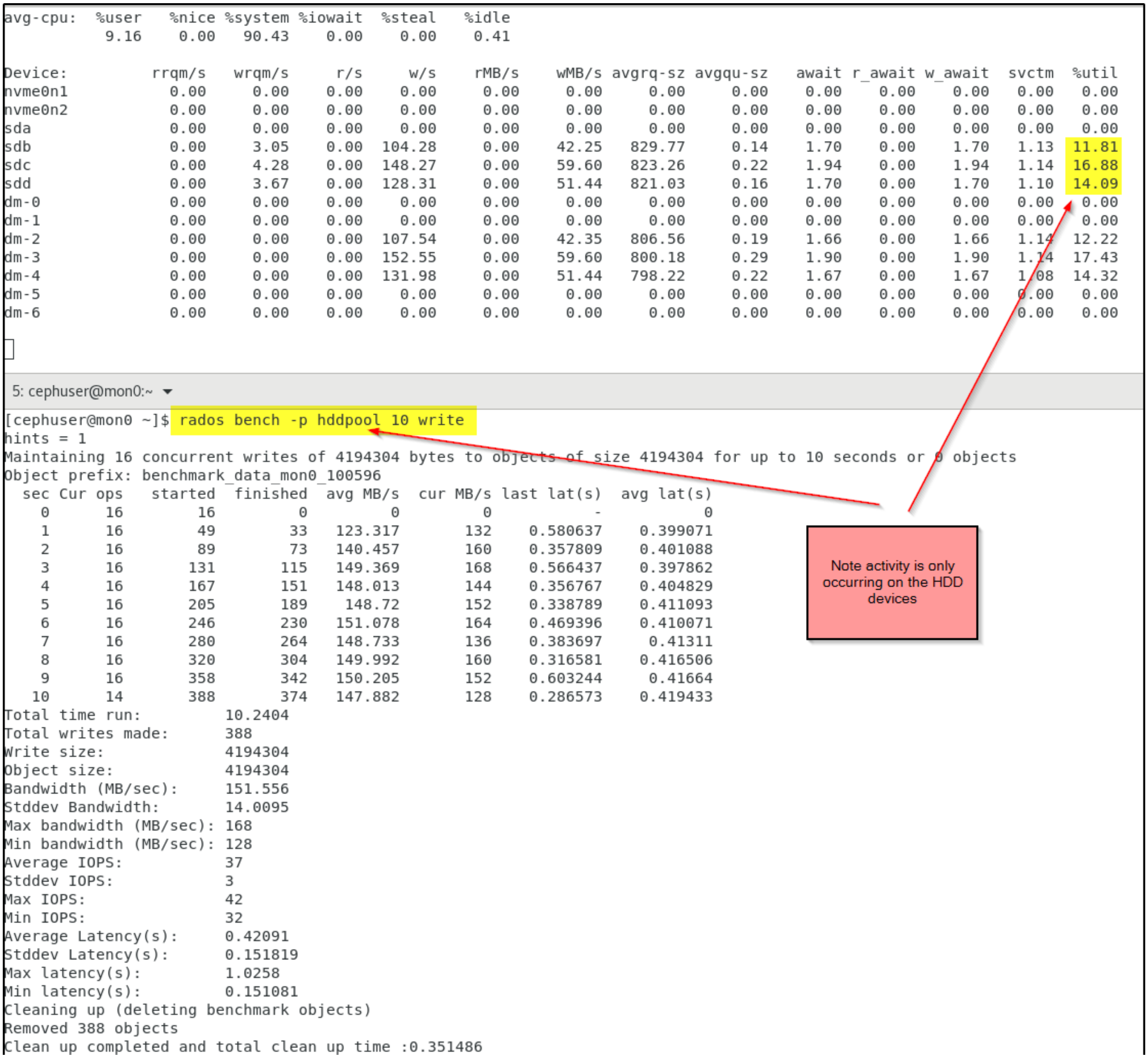
```
2.5c       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:43:12.273241     0'0     72:15  [14,11,9]
3.5c       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:56:05.342804     0'0     72:10    [3,6,1]
2.5d       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:43:12.254548     0'0     72:15 [14,10,12]
3.5f       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:56:05.335428     0'0     72:10    [2,4,6]
2.5e       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:43:12.260203     0'0     72:15 [10,14,12]
3.5e       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:56:05.252988     0'0     72:10    [1,6,4]
2.5f       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:43:12.241126     0'0     72:15 [13,11,10]
3.59       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:56:05.340937     0'0     72:10    [3,6,0]
2.58       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:43:12.255368     0'0     72:15 [14,12,10]
3.58       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:56:05.267845     0'0     72:10    [8,4,2]
2.59       0             0     0         0      0    0   0           0 active+clean 2019-02-18 10:43:12.271921     0'0     72:15  [11,14,9]
3.5b       0             0     0                                                  56:05.384960     0'0     72:10    [5,2,6]
2.5a       0             0     0                                                  43:12.259314     0'0     72:15  [13,9,11]
3.5a       0             0     0                                                  56:05.306477     0'0     72:10    [0,7,4]
2.5b       0             0     0                                                  43:12.259612     0'0     72:15 [10,13,12]
```

Note the highlighted pgs with an index of 2 (ssdpool) are using OSDs in the range of 9-14 and the PGs with an index of 3 (HDDs) are using OSDs in the range of 0-8).

Refer to the earlier output of `ceph osd tree` to check the OSD's device class!

Note extra points for commenting on the OSD distribution within each device class!

Testing the system



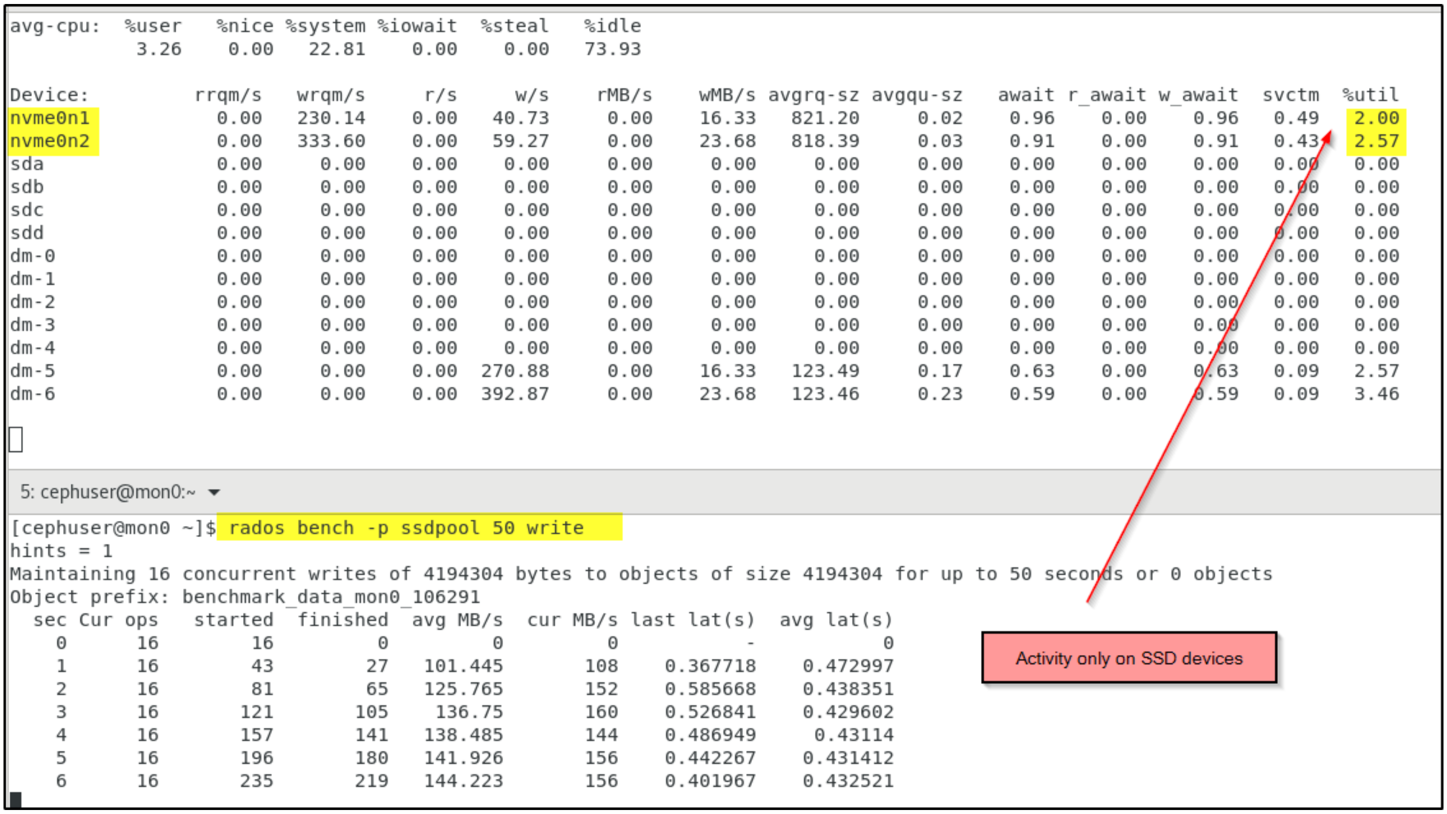
```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           9.16    0.00   90.43    0.00    0.00    0.41

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
nvme0n1           0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
nvme0n2           0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
sda               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
sdb               0.00     3.05    0.00  104.28     0.00    42.25   829.77     0.14    1.70    0.00    1.70   1.13  11.81
sdc               0.00     4.28    0.00  148.27     0.00    59.60   823.26     0.22    1.94    0.00    1.94   1.14  16.88
sdd               0.00     3.67    0.00  128.31     0.00    51.44   821.03     0.16    1.70    0.00    1.70   1.10  14.09
dm-0              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-1              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-2              0.00     0.00    0.00  107.54     0.00    42.35   806.56     0.19    1.66    0.00    1.66   1.14  12.22
dm-3              0.00     0.00    0.00  152.55     0.00    59.60   800.18     0.29    1.90    0.00    1.90   1.14  17.43
dm-4              0.00     0.00    0.00  131.98     0.00    51.44   798.22     0.22    1.67    0.00    1.67   1.08  14.32
dm-5              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-6              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
```

```
[cephuser@mon0 ~]$ rados bench -p hddpool 10 write
hints = 1
Maintaining 16 concurrent writes of 4194304 bytes to objects of size 4194304 for up to 10 seconds or 0 objects
Object prefix: benchmark_data_mon0_100596
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
    0      16        16         0         0         0           -           0
    1      16        49        33   123.317       132    0.580637    0.399071
    2      16        89        73   140.457       160    0.357809    0.401088
    3      16       131       115   149.369       168    0.566437    0.397862
    4      16       167       151   148.013       144    0.356767    0.404829
    5      16       205       189    148.72       152    0.338789    0.411093
    6      16       246       230   151.078       164    0.469396    0.410071
    7      16       280       264   148.733       136    0.383697     0.41311
    8      16       320       304   149.992       160    0.316581    0.416506
    9      16       358       342   150.205       152    0.603244     0.41664
   10      14       388       374   147.882       128    0.286573    0.419433
Total time run:         10.2404
Total writes made:      388
Write size:             4194304
Object size:            4194304
Bandwidth (MB/sec):     151.556
Stddev Bandwidth:       14.0095
Max bandwidth (MB/sec): 168
Min bandwidth (MB/sec): 128
Average IOPS:           37
Stddev IOPS:            3
Max IOPS:               42
Min IOPS:               32
Average Latency(s):     0.42091
Stddev Latency(s):      0.151819
Max latency(s):         1.0258
Min latency(s):         0.151081
Cleaning up (deleting benchmark objects)
Removed 388 objects
Clean up completed and total clean up time :0.351486
```

Note activity is only occurring on the HDD devices

The screen capture above shows the `hddpool` under test with the output of `iostat` only showing activity on the HDDs (as expected) and the screen capture below shows that the test is only using the NVMe devices.



```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           3.26    0.00   22.81    0.00    0.00   73.93

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
nvme0n1           0.00   230.14    0.00   40.73     0.00    16.33   821.20     0.02    0.96    0.00    0.96   0.49   2.00
nvme0n2           0.00   333.60    0.00   59.27     0.00    23.68   818.39     0.03    0.91    0.00    0.91   0.43   2.57
sda               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
sdb               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
sdc               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
sdd               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-0              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-1              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-2              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-3              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-4              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-5              0.00     0.00    0.00  270.88     0.00    16.33   123.49     0.17    0.63    0.00    0.63   0.09   2.57
dm-6              0.00     0.00    0.00  392.87     0.00    23.68   123.46     0.23    0.59    0.00    0.59   0.09   3.46
```

```
[cephuser@mon0 ~]$ rados bench -p ssdpool 50 write
hints = 1
Maintaining 16 concurrent writes of 4194304 bytes to objects of size 4194304 for up to 50 seconds or 0 objects
Object prefix: benchmark_data_mon0_106291
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
    0      16        16         0         0         0           -           0
    1      16        43        27   101.445       108    0.367718    0.472997
    2      16        81        65   125.765       152    0.585668    0.438351
    3      16       121       105    136.75       160    0.526841    0.429602
    4      16       157       141   138.485       144    0.486949     0.43114
    5      16       196       180   141.926       156    0.442267    0.431412
    6      16       235       219   144.223       156    0.401967    0.432521
```

Activity only on SSD devices

Creating an erasure coded ruleset for SSD devices

```
ceph osd erasure-code-profile set ssdprofile ruleset k=2 m=1 crush-device-class=ssd crush-failure-domain=host
```

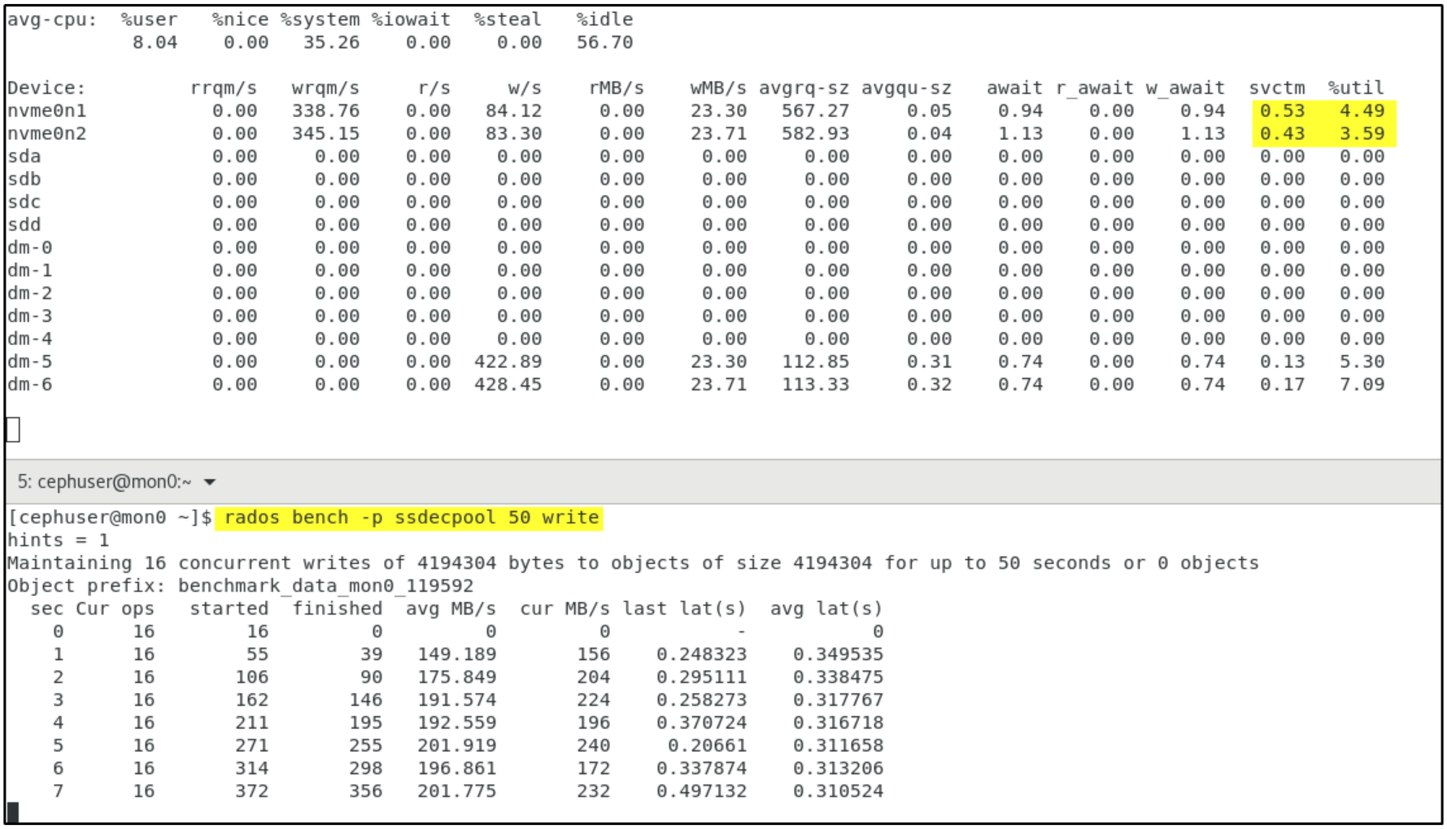
Retrieve the ruleset

```
$ ceph osd erasure-code-profile get ssdprofile
crush-device-class=ssd
crush-failure-domain=host
crush-root=default
jerasure-per-chunk-alignment=false
k=2
m=1
plugin=jerasure
ruleset=
technique=reed_sol_van
w=8
```

Create a pool

```
ceph osd pool create ssdecpool 128 128 erasure ssdprofile
```

Test the new pool



```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           8.04    0.00   35.26    0.00    0.00   56.70

Device:         rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
nvme0n1           0.00   338.76    0.00   84.12     0.00    23.30   567.27     0.05    0.94    0.00    0.94   0.53   4.49
nvme0n2           0.00   345.15    0.00   83.30     0.00    23.71   582.93     0.04    1.13    0.00    1.13   0.43   3.59
sda               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
sdb               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
sdc               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
sdd               0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-0              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-1              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-2              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-3              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-4              0.00     0.00    0.00    0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00   0.00   0.00
dm-5              0.00     0.00    0.00  422.89     0.00    23.30   112.85     0.31    0.74    0.00    0.74   0.13   5.30
dm-6              0.00     0.00    0.00  428.45     0.00    23.71   113.33     0.32    0.74    0.00    0.74   0.17   7.09
```

```
[cephuser@mon0 ~]$ rados bench -p ssdecpool 50 write
hints = 1
Maintaining 16 concurrent writes of 4194304 bytes to objects of size 4194304 for up to 50 seconds or 0 objects
Object prefix: benchmark_data_mon0_119592
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
    0      16        16         0         0         0           -           0
    1      16        55        39   149.189       156    0.248323    0.349535
    2      16       106        90   175.849       204    0.295111    0.338475
    3      16       162       146   191.574       224    0.258273    0.317767
    4      16       211       195   192.559       196    0.370724    0.316718
    5      16       271       255   201.919       240     0.20661    0.311658
    6      16       314       298   196.861       172    0.337874    0.313206
    7      16       372       356   201.775       232    0.497132    0.310524
```

For more information consult the ceph documentaion at http://docs.ceph.com/docs/master/