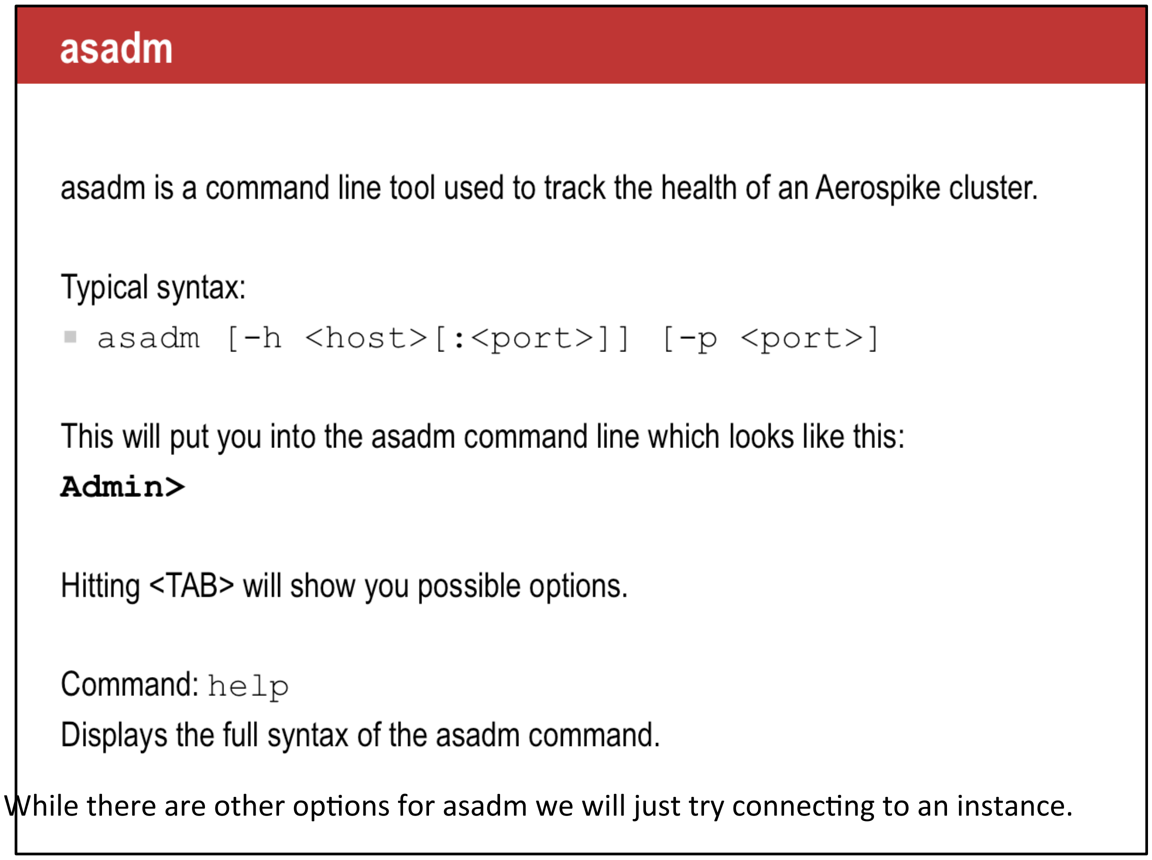
AEROSPIKE

Exercises and Tools

asadm



asadm

asadm is a command line tool used to track the health of an Aerospike cluster.

Typical syntax:

- `asadm [-h <host>[:<port>]] [-p <port>]`

This will put you into the asadm command line which looks like this:

```
Admin>
```

Hitting <TAB> will show you possible options.

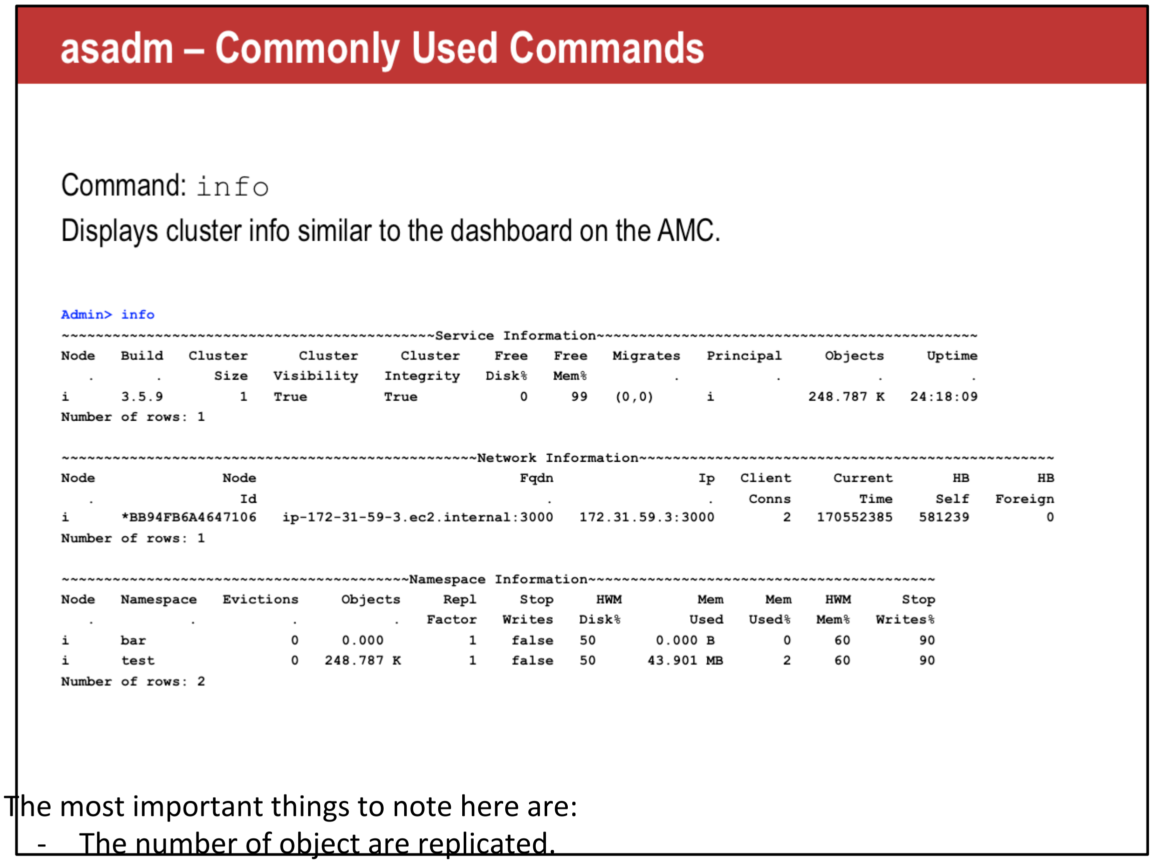
Command: `help`

Displays the full syntax of the asadm command.

While there are other options for asadm we will just try connecting to an instance.

The asadm does not need to be on the same host. You need only give asadm one host/IP address and it will connect to the other nodes in the cluster.

Just entering asadm without any parameters will put you into the local Aerospike node on port 3000.



asadm – Commonly Used Commands

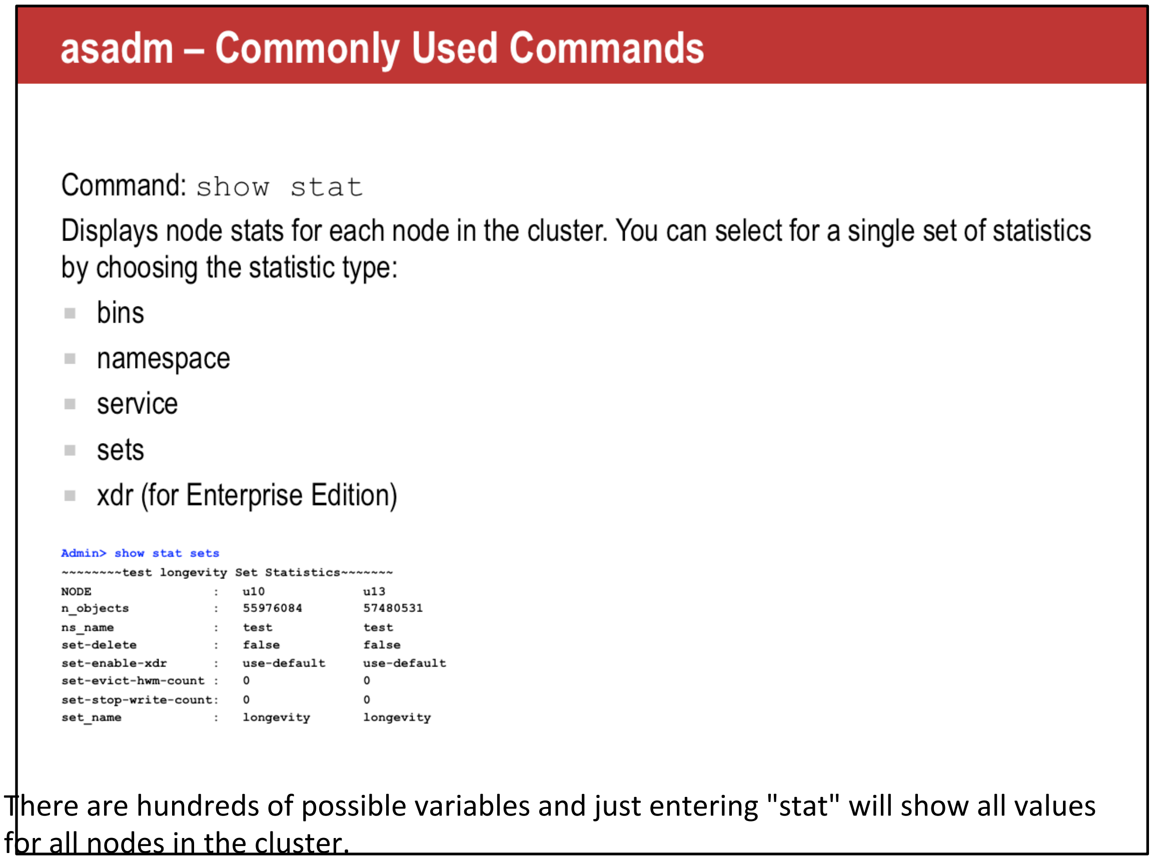
Command: `info`

Displays cluster info similar to the dashboard on the AMC.

```
Admin> info
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Service Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node   Build    Cluster      Cluster     Cluster    Free   Free   Migrates   Principal     Objects     Uptime
   .       .      Size   Visibility   Integrity   Disk%   Mem%           .           .           .          .
i      3.5.9         1   True         True            0     99    (0,0)      i           248.787 K    24:18:09
Number of rows: 1

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Network Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node               Node                               Fqdn                 Ip   Client      Current        HB        HB
   .                 Id                                  .                  .    Conns         Time      Self   Foreign
i      *BB94FB6A4647106   ip-172-31-59-3.ec2.internal:3000   172.31.59.3:3000        2   170552385   581239         0
Number of rows: 1

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Namespace Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node   Namespace   Evictions     Objects     Repl     Stop     HWM         Mem     Mem    HWM       Stop
   .           .           .           .   Factor   Writes   Disk%        Used   Used%   Mem%    Writes%
i      bar                 0     0.000          1    false   50        0.000 B       0     60         90
i      test                0   248.787 K        1    false   50      43.901 MB       2     60         90
Number of rows: 2
```

The most important things to note here are:

- The number of object are replicated.
- If the number of migrates is non-zero, the cluster is in a dynamic state.
- There are counters for the number of evicted objects, if this is increasing, the system is short on configured resources.



asadm – Commonly Used Commands

Command: `show stat`

Displays node stats for each node in the cluster. You can select for a single set of statistics by choosing the statistic type:

- bins
- namespace
- service
- sets
- xdr (for Enterprise Edition)

```
Admin> show stat sets
~~~~~~~~test longevity Set Statistics~~~~~~~
NODE                   :   u10             u13
n_objects              :   55976084        57480531
ns_name                :   test            test
set-delete             :   false           false
set-enable-xdr         :   use-default     use-default
set-evict-hwm-count    :   0               0
set-stop-write-count:      0               0
set_name               :   longevity       longevity
```

There are hundreds of possible variables and just entering "stat" will show all values for all nodes in the cluster.

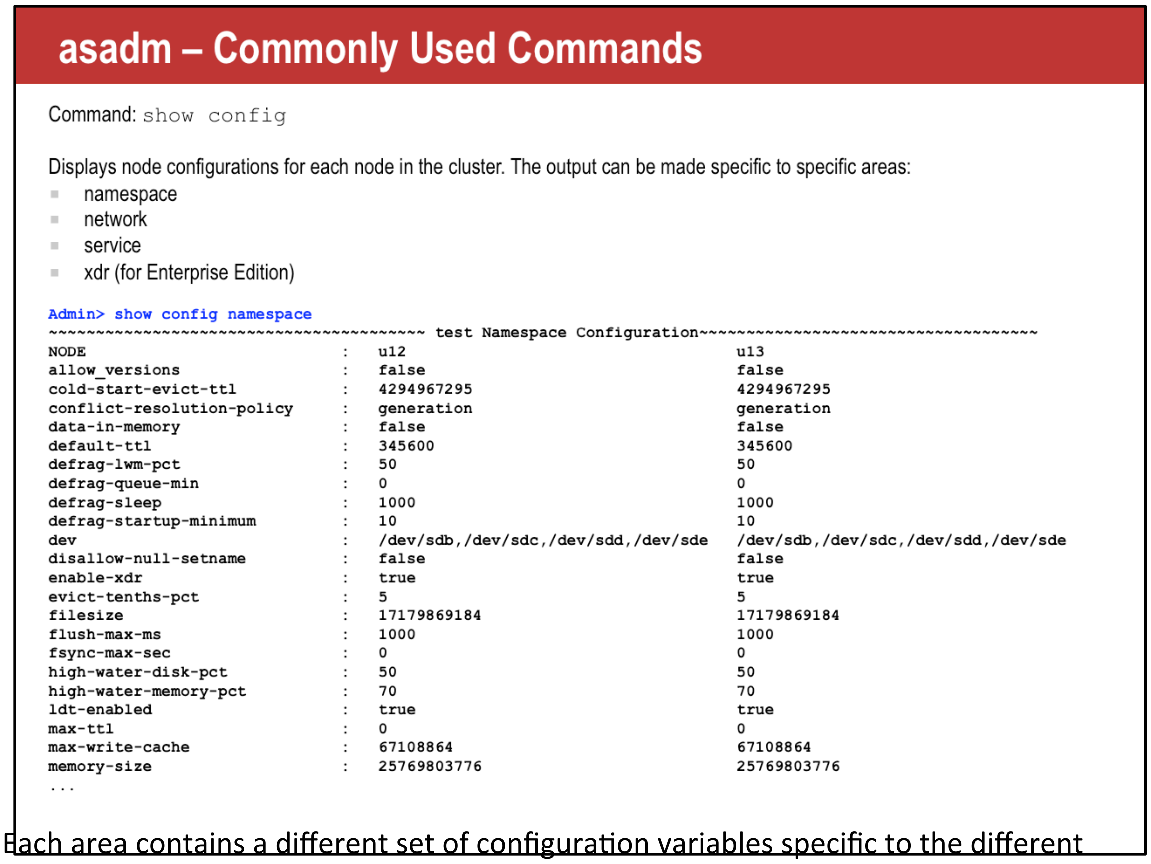
asadm – Commonly Used Commands

Command: `show stat`

Displays node stats for each node in the cluster. The output can be very long, so filter with the “like” modifer.

```
Admin> show stat like total
~~~~~~~~~~~~~~~~~~~~~~~Service Statistics~~~~~~~~~~~~~~~~~~~~~~~
NODE               :   u10            u12            u13
total-bytes-disk   :   800197705728   800197705728   948214693888
total-bytes-memory:   25769803776    25769803776    25769803776

~~~~~~~~~~~~~~~~~~ test Namespace Statistics~~~~~~~~~~~~~~~~~~~~
NODE               :   u10            u12            u13
total-bytes-disk   :   800197705728   800197705728   948214693888
total-bytes-memory:   25769803776    25769803776    25769803776
```

It is often easier to filter for just the variables you are interested in. Use the "like" will limit the variables to those that contain the string.



asadm – Commonly Used Commands

Command: `show config`

Displays node configurations for each node in the cluster. The output can be made specific to specific areas:

- namespace
- network
- service
- xdr (for Enterprise Edition)

```
Admin> show config namespace
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ test Namespace Configuration~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
NODE                           :   u12                                   u13
allow_versions                 :   false                                 false
cold-start-evict-ttl           :   4294967295                            4294967295
conflict-resolution-policy     :   generation                            generation
data-in-memory                 :   false                                 false
default-ttl                    :   345600                                345600
defrag-lwm-pct                 :   50                                    50
defrag-queue-min               :   0                                     0
defrag-sleep                   :   1000                                  1000
defrag-startup-minimum         :   10                                    10
dev                            :   /dev/sdb,/dev/sdc,/dev/sdd,/dev/sde   /dev/sdb,/dev/sdc,/dev/sdd,/dev/sde
disallow-null-setname          :   false                                 false
enable-xdr                     :   true                                  true
evict-tenths-pct               :   5                                     5
filesize                       :   17179869184                           17179869184
flush-max-ms                   :   1000                                  1000
fsync-max-sec                  :   0                                     0
high-water-disk-pct            :   50                                    50
high-water-memory-pct          :   70                                    70
ldt-enabled                    :   true                                  true
max-ttl                        :   0                                     0
max-write-cache                :   67108864                              67108864
memory-size                    :   25769803776                           25769803776
...
```

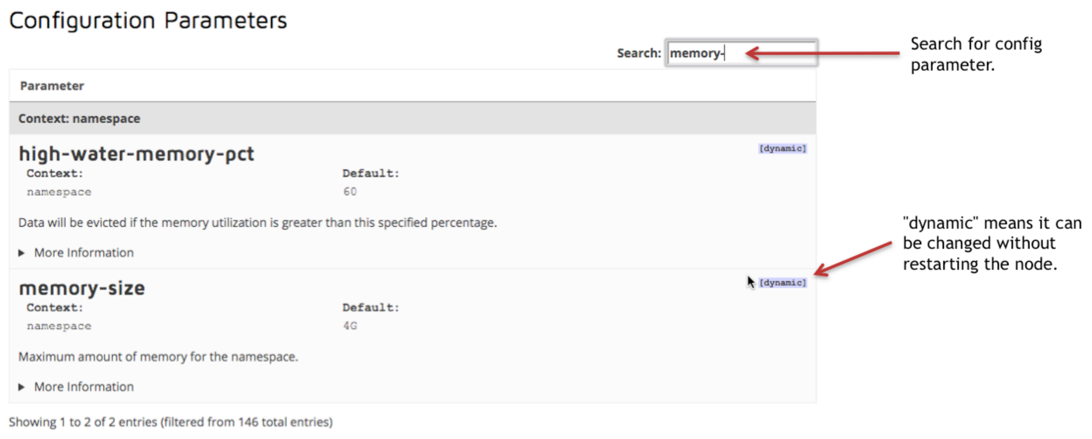
Each area contains a different set of configuration variables specific to the different contexts in the configuration file.

asadm – Commonly Used Commands

Command: `asinfo -v`

Dynamically alters the configuration of the nodes in the cluster. The context will match the area in the configuration file. Not all variables are dynamically changeable.

Go to http://www.aerospike.com/docs/reference/configuration/



Configuration Parameters

Search: memory-

Search for config parameter.

Parameter	
Context: namespace	

high-water-memory-pct [dynamic]

Context:	Default:
namespace	60

Data will be evicted if the memory utilization is greater than this specified percentage.

▸ More Information

memory-size [dynamic]

Context:	Default:
namespace	4G

Maximum amount of memory for the namespace.

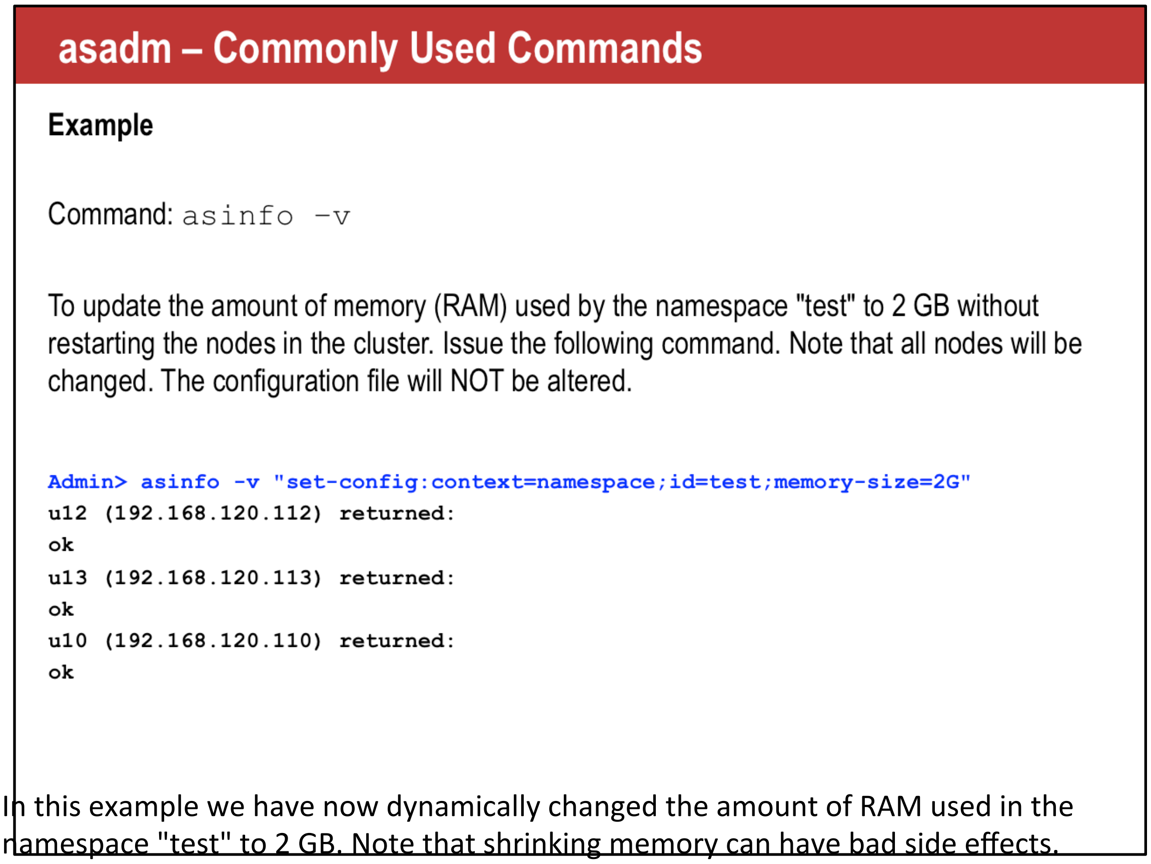
▸ More Information

Showing 1 to 2 of 2 entries (filtered from 146 total entries)

"dynamic" means it can be changed without restarting the node.

You may find that you need to change a configuration variable. First, you may want to determine if it can be changed without restarting the node.

While this is often true, it is not always true. You can find out if the parameter you want to change is by looking at the Aerospike web site.



asadm – Commonly Used Commands

Example

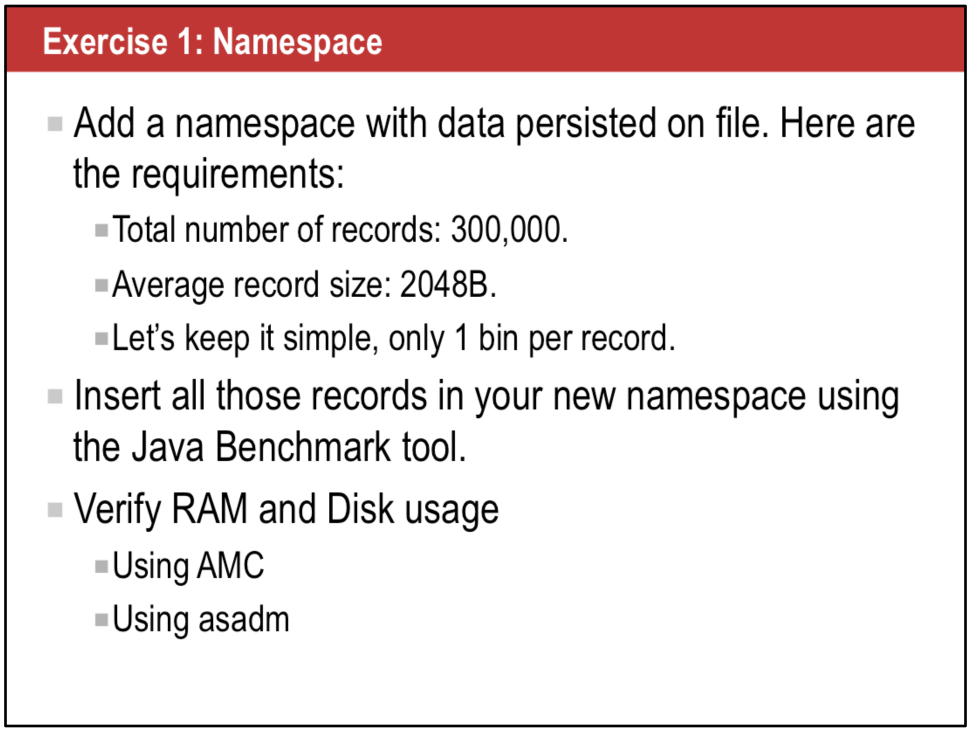
Command: `asinfo -v`

To update the amount of memory (RAM) used by the namespace "test" to 2 GB without restarting the nodes in the cluster. Issue the following command. Note that all nodes will be changed. The configuration file will NOT be altered.

```
Admin> asinfo -v "set-config:context=namespace;id=test;memory-size=2G"
u12 (192.168.120.112) returned:
ok
u13 (192.168.120.113) returned:
ok
u10 (192.168.120.110) returned:
ok
```

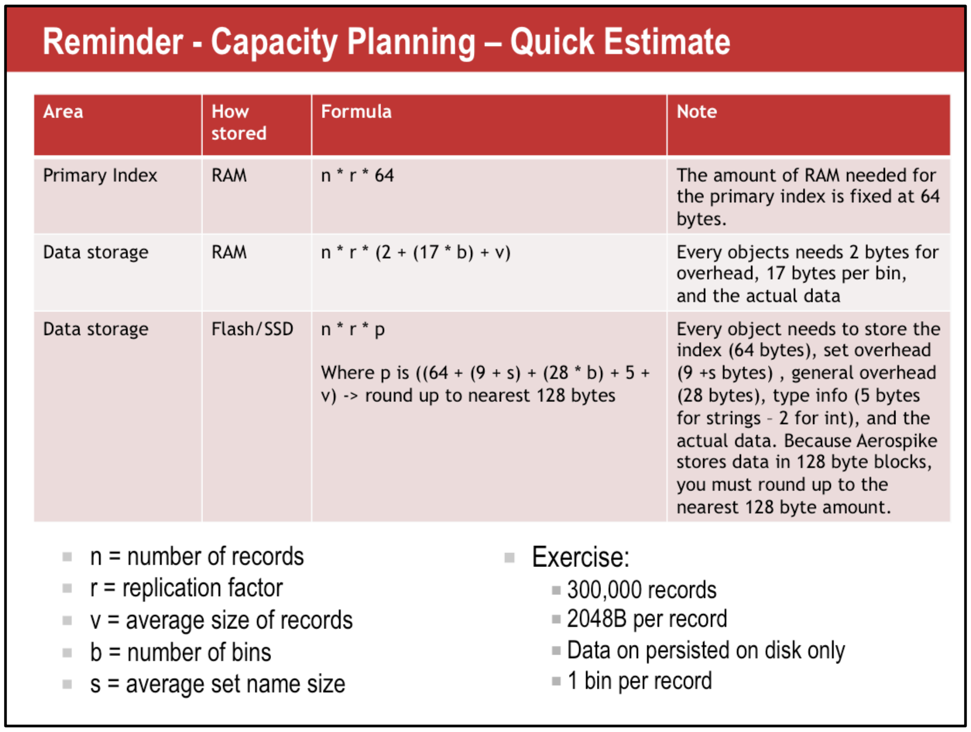
In this example we have now dynamically changed the amount of RAM used in the namespace "test" to 2 GB. Note that shrinking memory can have bad side effects.

Exercises



Exercise 1: Namespace

- Add a namespace with data persisted on file. Here are the requirements:
  - Total number of records: 300,000.
  - Average record size: 2048B.
  - Let's keep it simple, only 1 bin per record.
- Insert all those records in your new namespace using the Java Benchmark tool.
- Verify RAM and Disk usage
  - Using AMC
  - Using asadm



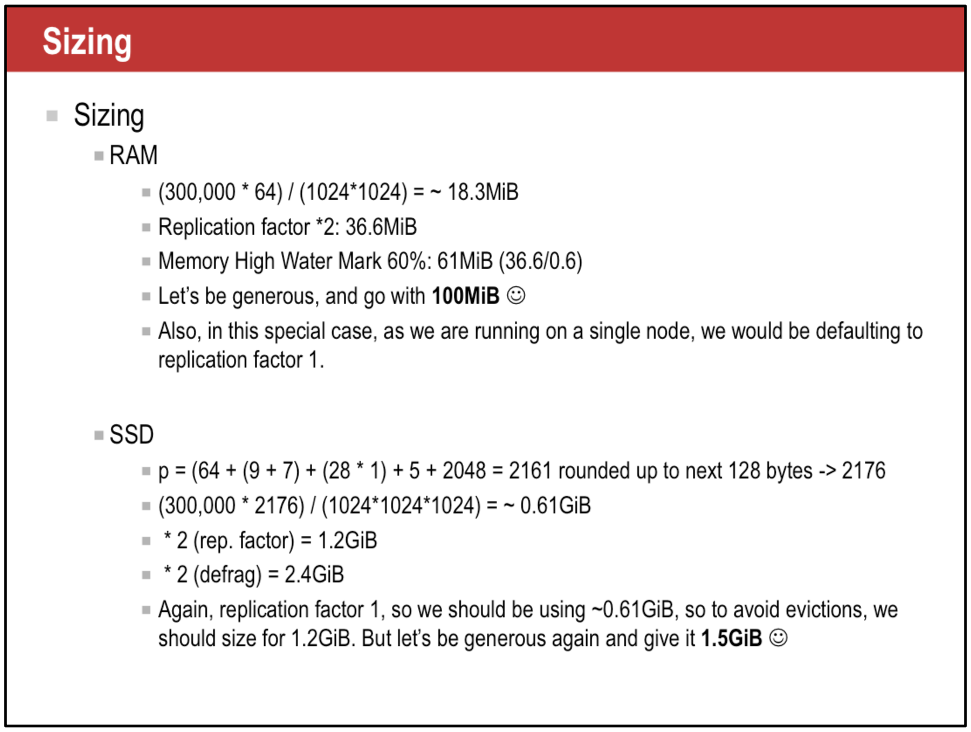
Reminder - Capacity Planning – Quick Estimate

Area	How stored	Formula	Note
Primary Index	RAM	n * r * 64	The amount of RAM needed for the primary index is fixed at 64 bytes.
Data storage	RAM	n * r * (2 + (17 * b) + v)	Every objects needs 2 bytes for overhead, 17 bytes per bin, and the actual data
Data storage	Flash/SSD	n * r * p Where p is ((64 + (9 + s) + (28 * b) + 5 + v) -> round up to nearest 128 bytes	Every object needs to store the index (64 bytes), set overhead (9 +s bytes) , general overhead (28 bytes), type info (5 bytes for strings - 2 for int), and the actual data. Because Aerospike stores data in 128 byte blocks, you must round up to the nearest 128 byte amount.

- n = number of records
- r = replication factor
- v = average size of records
- b = number of bins
- s = average set name size

- Exercise:
  - 300,000 records
  - 2048B per record
  - Data on persisted on disk only
  - 1 bin per record

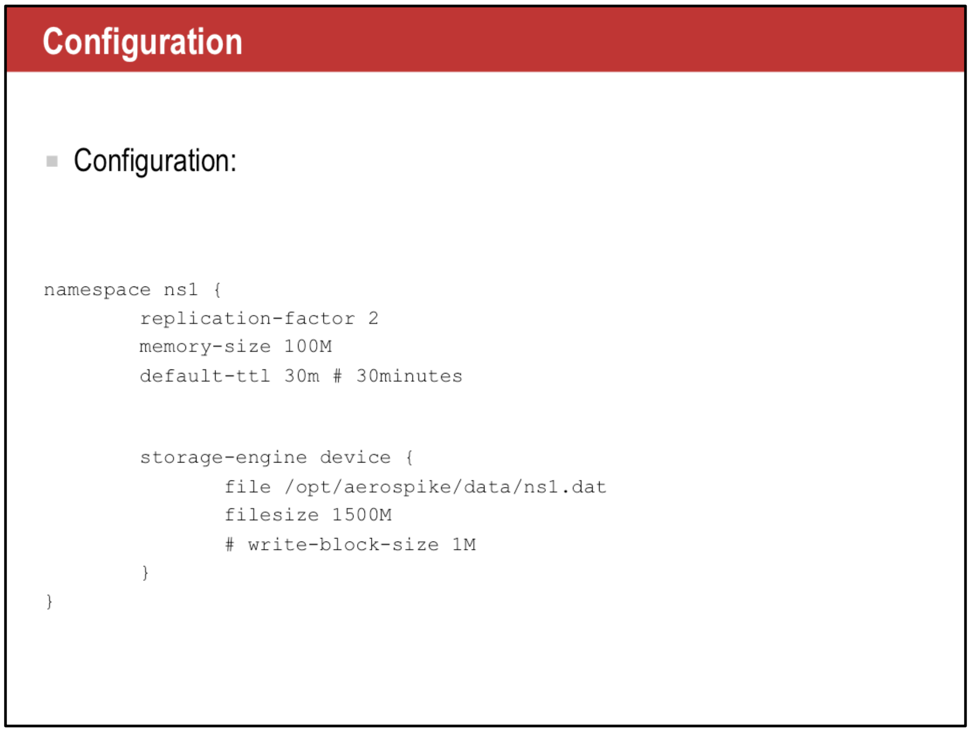
For the exercise: 2048 + 5 + 28 +64 = 2145, closest 128 bytes is 17 * 128 = 2176.


Sizing

- Sizing
  - RAM
    - (300,000 * 64) / (1024*1024) = ~ 18.3MiB
    - Replication factor *2: 36.6MiB
    - Memory High Water Mark 60%: 61MiB (36.6/0.6)
    - Let's be generous, and go with **100MiB** ☺
    - Also, in this special case, as we are running on a single node, we would be defaulting to replication factor 1.
  - SSD
    - p = (64 + (9 + 7) + (28 * 1) + 5 + 2048 = 2161 rounded up to next 128 bytes -> 2176
    - (300,000 * 2176) / (1024*1024*1024) = ~ 0.61GiB
    - * 2 (rep. factor) = 1.2GiB
    - * 2 (defrag) = 2.4GiB
    - Again, replication factor 1, so we should be using ~0.61GiB, so to avoid evictions, we should size for 1.2GiB. But let's be generous again and give it **1.5GiB** ☺

Why *2 for defrag? At defrag_lwm_pct 50, to write 1 new block, would need to defrag 2 blocks at 50% each, so, you would need to read 2 such blocks, write 1 resulting block after defrag, and then the block holding new records. Would end up writing 2 blocks (1 new data, 1 for having defragged 2 blocks at 50% each) and reading 2 blocks. So write amplification is 2X. At 75% defrag_lwm_pct, write amplification is 4X. Disk will wear faster and performance will be impacted too.

Real sizing would also need to account for read/write peak tps and potential secondary index using more RAM.



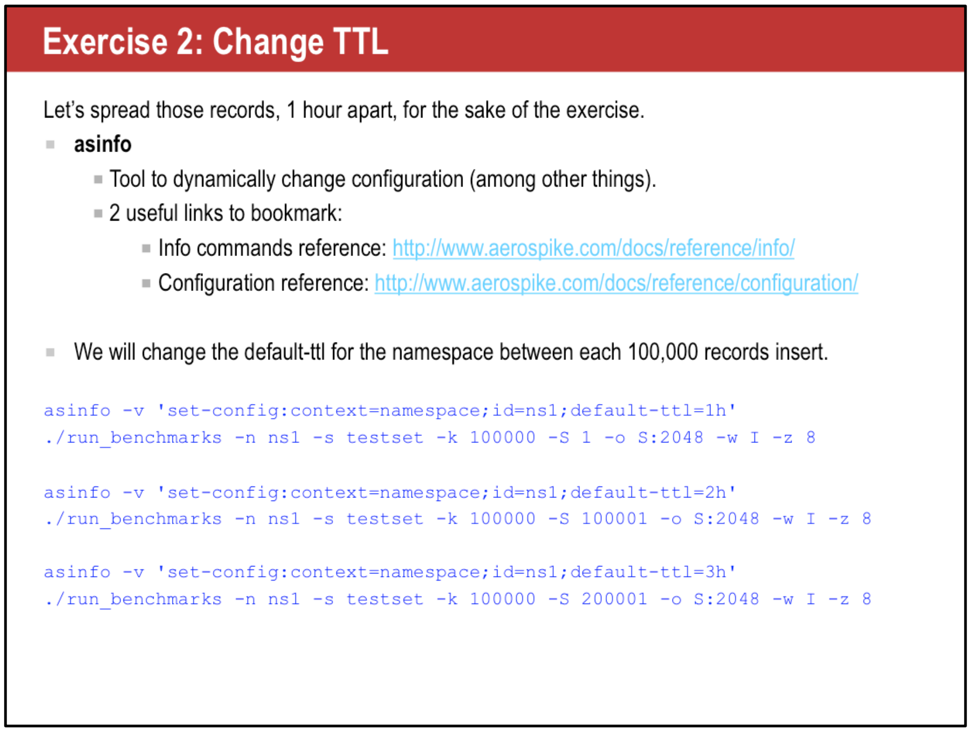
Configuration

- Configuration:

```
namespace ns1 {
        replication-factor 2
        memory-size 100M
        default-ttl 30m # 30minutes

        storage-engine device {
               file /opt/aerospike/data/ns1.dat
               filesize 1500M
               # write-block-size 1M
        }
}
```

Why *2 for defrag? At defrag_lwm_pct 50, to write 1 new block, would need to defrag 2 blocks at 50% each, so, you would need to read 2 such blocks, write 1 resulting block after defrag, and then the block holding new records. Would end up writing 2 blocks (1 new data, 1 for having defragged 2 blocks at 50% each) and reading 2 blocks. So write amplification is 2X. At 75% defrag_lwm_pct, write amplification is 4X. Disk will wear faster and performance will be impacted too.

Real sizing would also need to account for read/write peak tps and potential secondary index using more RAM.



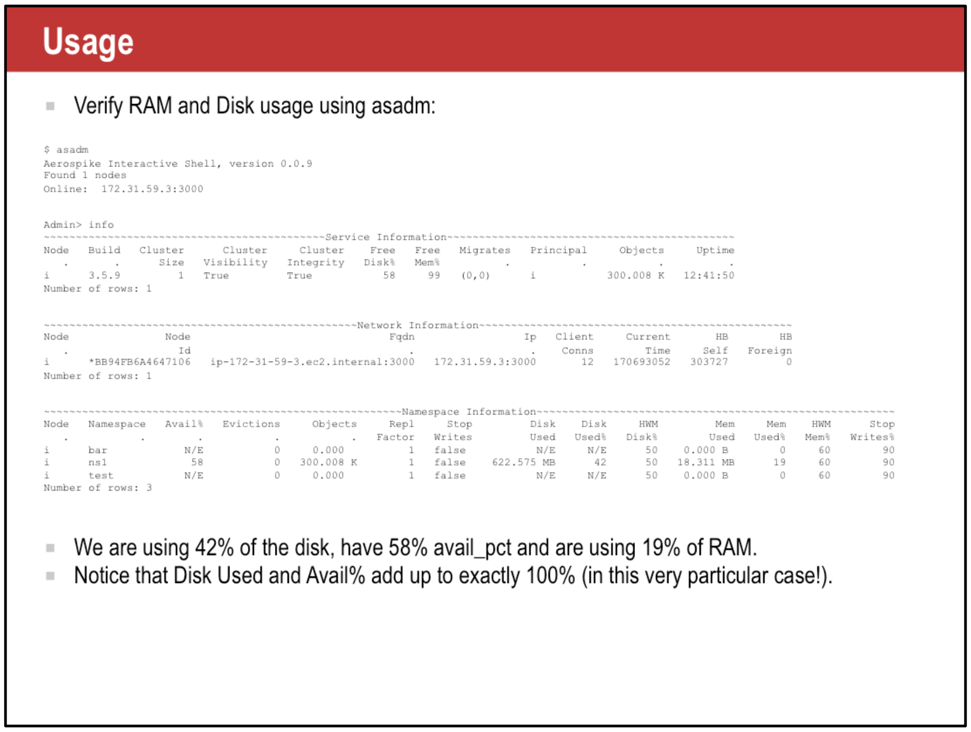
Exercise 2: Change TTL

Let's spread those records, 1 hour apart, for the sake of the exercise.

- **asinfo**
  - Tool to dynamically change configuration (among other things).
  - 2 useful links to bookmark:
    - Info commands reference: http://www.aerospike.com/docs/reference/info/
    - Configuration reference: http://www.aerospike.com/docs/reference/configuration/

- We will change the default-ttl for the namespace between each 100,000 records insert.

```
asinfo -v 'set-config:context=namespace;id=ns1;default-ttl=1h'
./run_benchmarks -n ns1 -s testset -k 100000 -S 1 -o S:2048 -w I -z 8

asinfo -v 'set-config:context=namespace;id=ns1;default-ttl=2h'
./run_benchmarks -n ns1 -s testset -k 100000 -S 100001 -o S:2048 -w I -z 8

asinfo -v 'set-config:context=namespace;id=ns1;default-ttl=3h'
./run_benchmarks -n ns1 -s testset -k 100000 -S 200001 -o S:2048 -w I -z 8
```



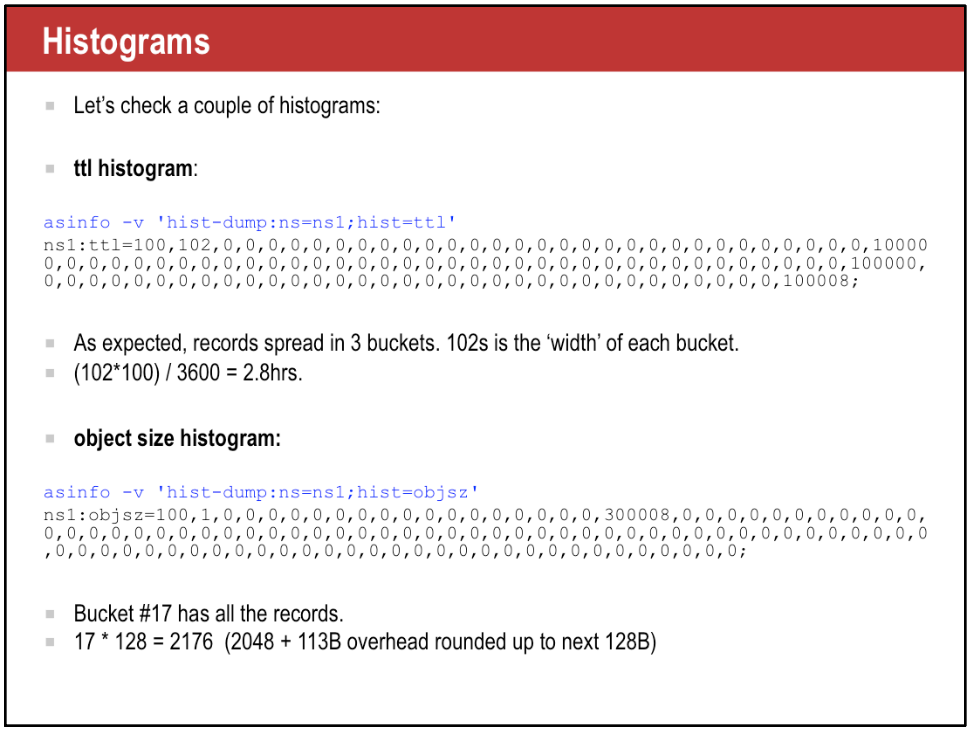
Usage

- Verify RAM and Disk usage using asadm:

```
$ asadm
Aerospike Interactive Shell, version 0.0.9
Found 1 nodes
Online:  172.31.59.3:3000

Admin> info
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Service Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node   Build   Cluster      Cluster     Cluster    Free   Free   Migrates   Principal     Objects      Uptime
   .       .      Size   Visibility   Integrity   Disk%   Mem%          .           .           .           .
i      3.5.9         1   True         True           58     99   (0,0)      i           300.008 K   12:41:50
Number of rows: 1

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Network Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node               Node                               Fqdn                 Ip   Client     Current       HB        HB
   .                 Id                                  .                  .    Conns        Time     Self   Foreign
i      *BB94FB6A4647106   ip-172-31-59-3.ec2.internal:3000   172.31.59.3:3000       12   170693052   303727         0
Number of rows: 1

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Namespace Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node   Namespace   Avail%   Evictions      Objects     Repl    Stop          Disk    Disk     HWM          Mem    Mem    HWM      Stop
   .           .        .           .            .   Factor   Writes         Used   Used%   Disk%         Used   Used%   Mem%   Writes%
i      bar            N/E           0      0.000          1   false           N/E     N/E      50    0.000 B        0     60        90
i      ns1             58           0    300.008 K        1   false    622.575 MB      42      50   18.311 MB      19     60        90
i      test           N/E           0      0.000          1   false           N/E     N/E      50    0.000 B        0     60        90
Number of rows: 3
```

- We are using 42% of the disk, have 58% avail_pct and are using 19% of RAM.
- Notice that Disk Used and Avail% add up to exactly 100% (in this very particular case!).

Disk Used and Avail% do add up because we only inserted new records and never updated/deleted any of them. So all blocks are full, no fragmentation.



Histograms

- Let's check a couple of histograms:
- **ttl histogram**:

```
asinfo -v 'hist-dump:ns=ns1;hist=ttl'
ns1:ttl=100,102,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,10000
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,100000,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,100008;
```

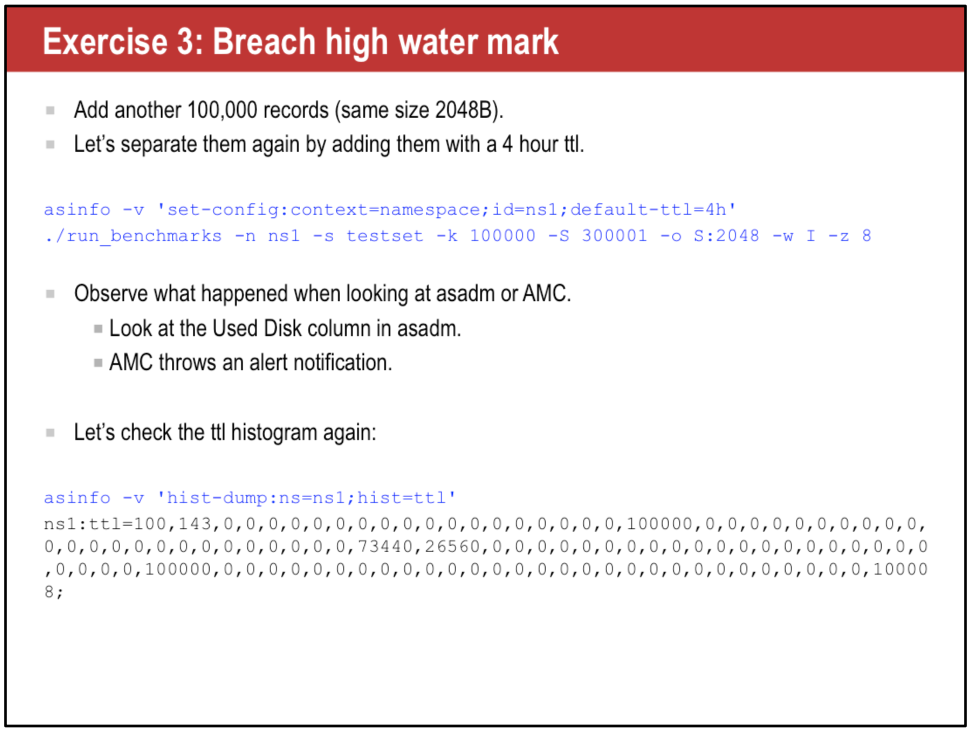
- As expected, records spread in 3 buckets. 102s is the 'width' of each bucket.
- (102*100) / 3600 = 2.8hrs.
- **object size histogram:**

```
asinfo -v 'hist-dump:ns=ns1;hist=objsz'
ns1:objsz=100,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,300008,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
```

- Bucket #17 has all the records.
- 17 * 128 = 2176 (2048 + 113B overhead rounded up to next 128B)

Firs number of buckets
2nd bucket width = 102 second
The the buckets = the number of records to expire in each bucket

Number of buckets
Always 1 : record block = 128b
Always 0



Exercise 3: Breach high water mark

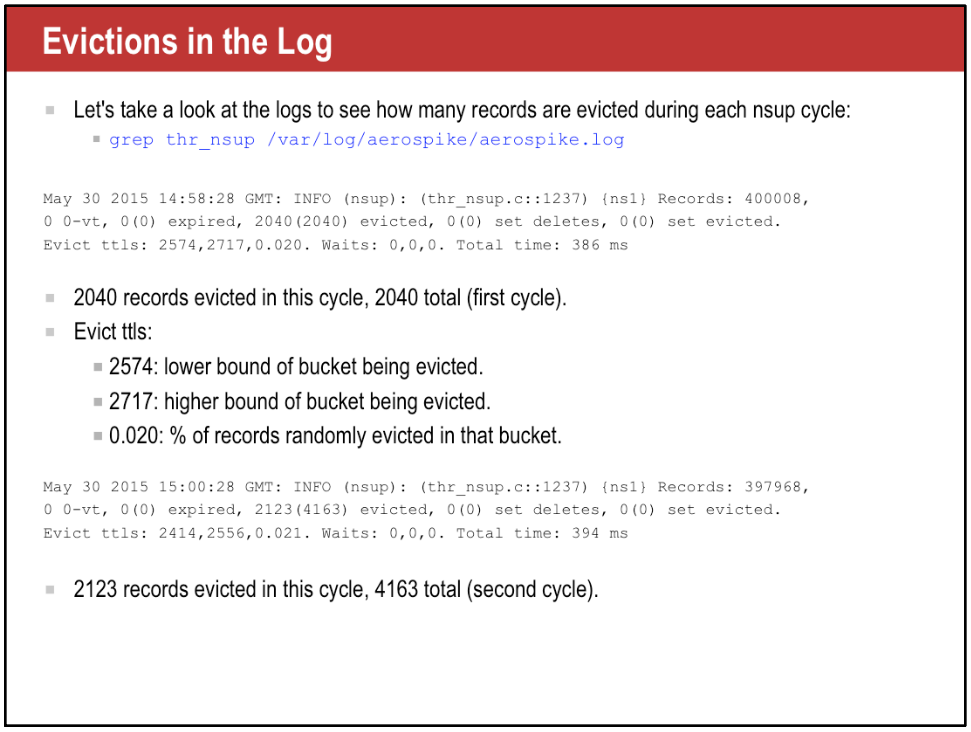
- Add another 100,000 records (same size 2048B).
- Let's separate them again by adding them with a 4 hour ttl.

```
asinfo -v 'set-config:context=namespace;id=ns1;default-ttl=4h'
./run_benchmarks -n ns1 -s testset -k 100000 -S 300001 -o S:2048 -w I -z 8
```

- Observe what happened when looking at asadm or AMC.
  - Look at the Used Disk column in asadm.
  - AMC throws an alert notification.

- Let's check the ttl histogram again:

```
asinfo -v 'hist-dump:ns=ns1;hist=ttl'
ns1:ttl=100,143,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,100000,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,73440,26560,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,100000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,10000
8;
```



Evictions in the Log

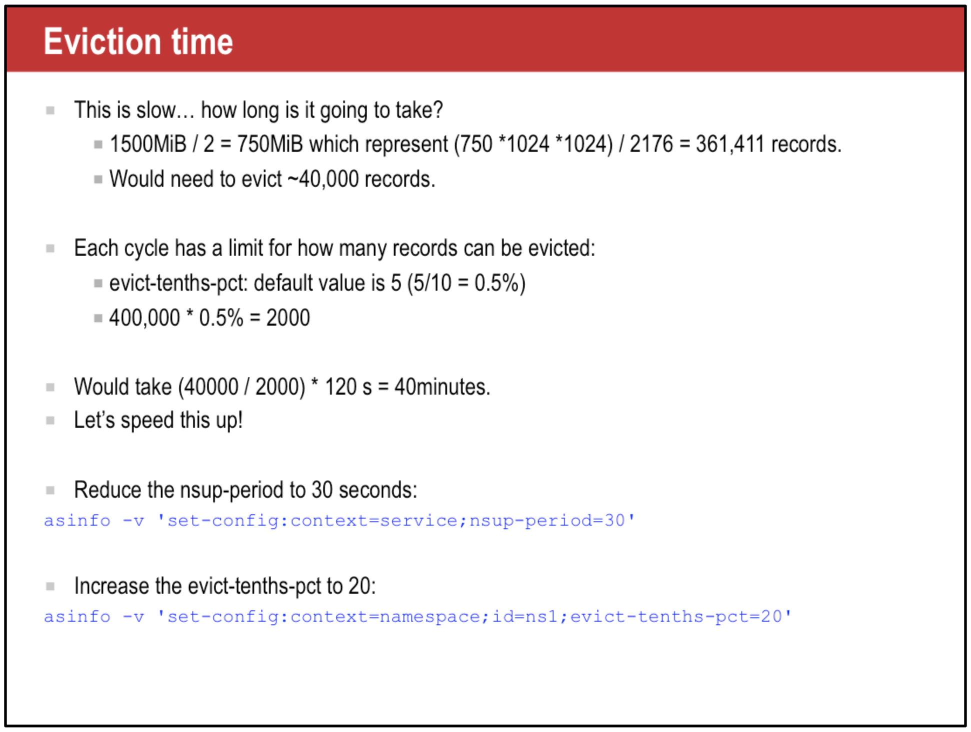
- Let's take a look at the logs to see how many records are evicted during each nsup cycle:
  - `grep thr_nsup /var/log/aerospike/aerospike.log`

```
May 30 2015 14:58:28 GMT: INFO (nsup): (thr_nsup.c::1237) {ns1} Records: 400008,
0 0-vt, 0(0) expired, 2040(2040) evicted, 0(0) set deletes, 0(0) set evicted.
Evict ttls: 2574,2717,0.020. Waits: 0,0,0. Total time: 386 ms
```

- 2040 records evicted in this cycle, 2040 total (first cycle).
- Evict ttls:
  - 2574: lower bound of bucket being evicted.
  - 2717: higher bound of bucket being evicted.
  - 0.020: % of records randomly evicted in that bucket.

```
May 30 2015 15:00:28 GMT: INFO (nsup): (thr_nsup.c::1237) {ns1} Records: 397968,
0 0-vt, 0(0) expired, 2123(4163) evicted, 0(0) set deletes, 0(0) set evicted.
Evict ttls: 2414,2556,0.021. Waits: 0,0,0. Total time: 394 ms
```

- 2123 records evicted in this cycle, 4163 total (second cycle).



Eviction time

- This is slow… how long is it going to take?
  - 1500MiB / 2 = 750MiB which represent (750 *1024 *1024) / 2176 = 361,411 records.
  - Would need to evict ~40,000 records.

- Each cycle has a limit for how many records can be evicted:
  - evict-tenths-pct: default value is 5 (5/10 = 0.5%)
  - 400,000 * 0.5% = 2000

- Would take (40000 / 2000) * 120 s = 40minutes.
- Let's speed this up!

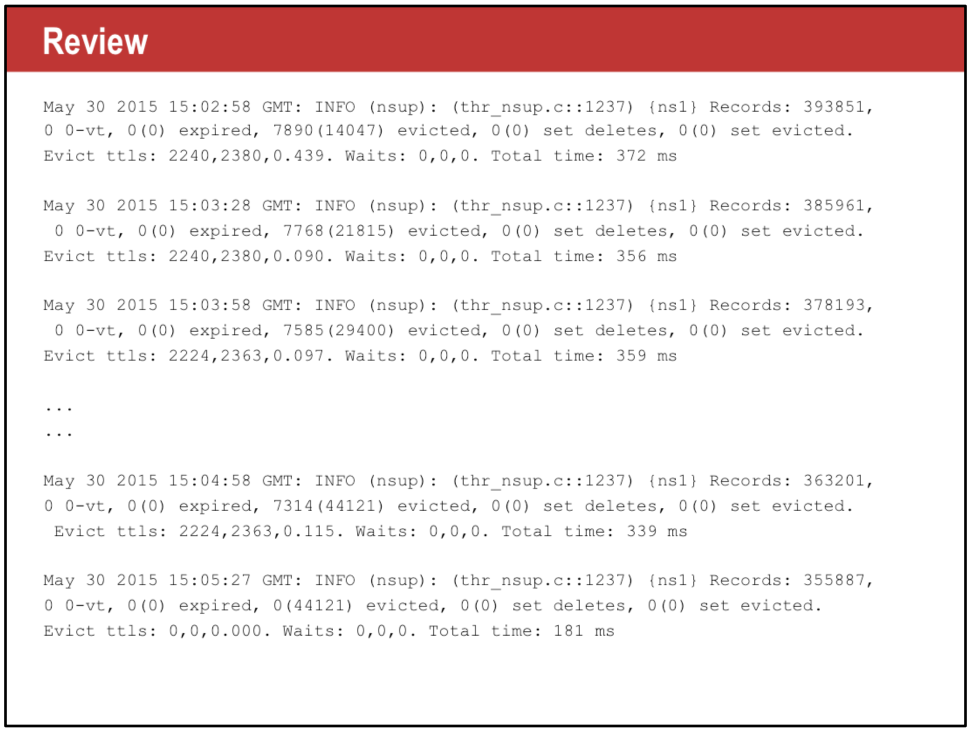
- Reduce the nsup-period to 30 seconds:

```
asinfo -v 'set-config:context=service;nsup-period=30'
```

- Increase the evict-tenths-pct to 20:

```
asinfo -v 'set-config:context=namespace;id=ns1;evict-tenths-pct=20'
```

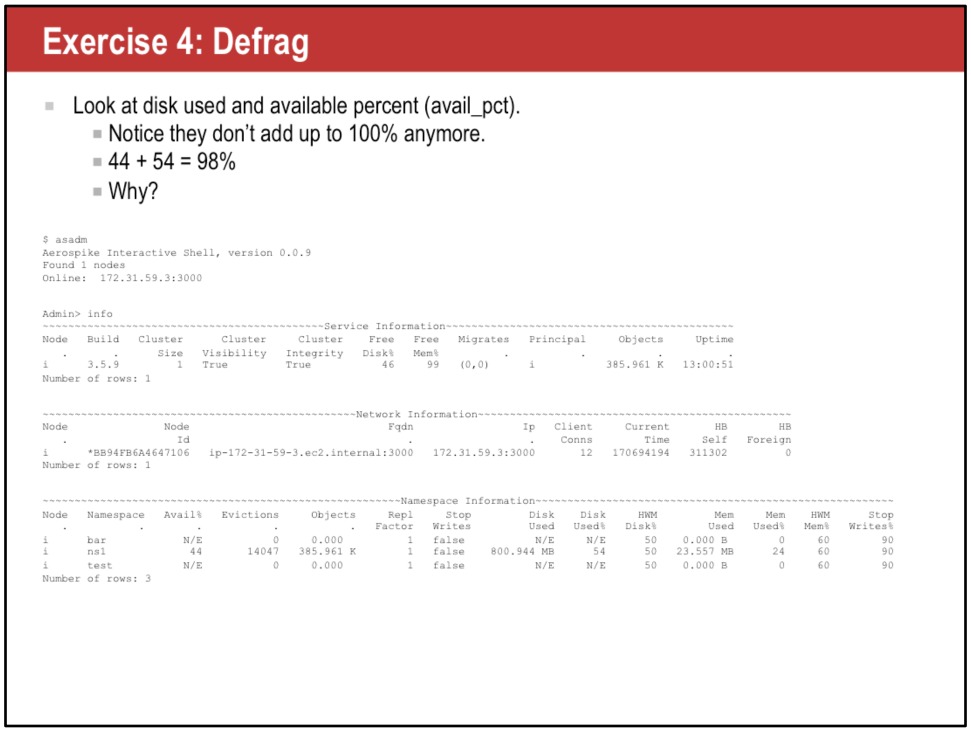
Do the math
Speed it up



Review

```
May 30 2015 15:02:58 GMT: INFO (nsup): (thr_nsup.c::1237) {ns1} Records: 393851,
0 0-vt, 0(0) expired, 7890(14047) evicted, 0(0) set deletes, 0(0) set evicted.
Evict ttls: 2240,2380,0.439. Waits: 0,0,0. Total time: 372 ms

May 30 2015 15:03:28 GMT: INFO (nsup): (thr_nsup.c::1237) {ns1} Records: 385961,
 0 0-vt, 0(0) expired, 7768(21815) evicted, 0(0) set deletes, 0(0) set evicted.
Evict ttls: 2240,2380,0.090. Waits: 0,0,0. Total time: 356 ms

May 30 2015 15:03:58 GMT: INFO (nsup): (thr_nsup.c::1237) {ns1} Records: 378193,
 0 0-vt, 0(0) expired, 7585(29400) evicted, 0(0) set deletes, 0(0) set evicted.
Evict ttls: 2224,2363,0.097. Waits: 0,0,0. Total time: 359 ms

...
...

May 30 2015 15:04:58 GMT: INFO (nsup): (thr_nsup.c::1237) {ns1} Records: 363201,
0 0-vt, 0(0) expired, 7314(44121) evicted, 0(0) set deletes, 0(0) set evicted.
 Evict ttls: 2224,2363,0.115. Waits: 0,0,0. Total time: 339 ms

May 30 2015 15:05:27 GMT: INFO (nsup): (thr_nsup.c::1237) {ns1} Records: 355887,
0 0-vt, 0(0) expired, 0(44121) evicted, 0(0) set deletes, 0(0) set evicted.
Evict ttls: 0,0,0.000. Waits: 0,0,0. Total time: 181 ms
```

Notes here

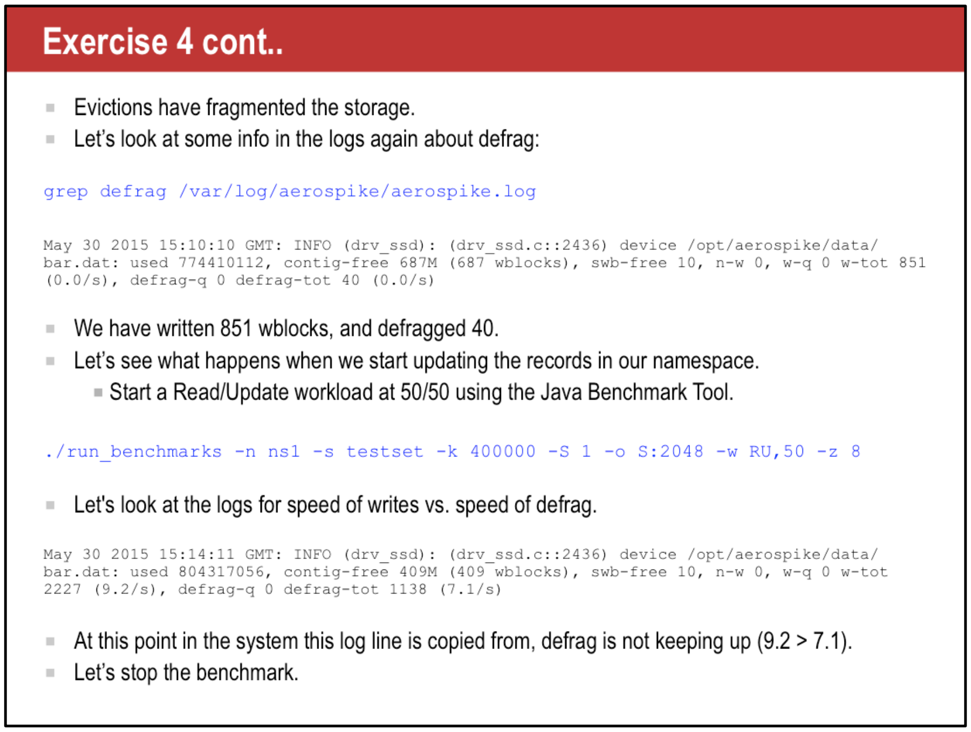


Exercise 4: Defrag

- Look at disk used and available percent (avail_pct).
  - Notice they don't add up to 100% anymore.
  - 44 + 54 = 98%
  - Why?

```
$ asadm
Aerospike Interactive Shell, version 0.0.9
Found 1 nodes
Online:  172.31.59.3:3000

Admin> info
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Service Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node   Build  Cluster      Cluster     Cluster    Free   Free   Migrates   Principal     Objects      Uptime
   .       .     Size   Visibility   Integrity   Disk%   Mem%          .           .           .           .
i      3.5.9        1   True         True           46     99   (0,0)      i           385.961 K   13:00:51
Number of rows: 1

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Network Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node               Node                                Fqdn                   Ip   Client      Current       HB        HB
   .                 Id                                   .                    .    Conns         Time     Self   Foreign
i      *BB94FB6A4647106   ip-172-31-59-3.ec2.internal:3000   172.31.59.3:3000       12   170694194   311302         0
Number of rows: 1

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Namespace Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node   Namespace   Avail%   Evictions     Objects     Repl     Stop         Disk     Disk     HWM         Mem     Mem    HWM      Stop
   .           .        .           .           .   Factor   Writes         Used    Used%   Disk%        Used   Used%   Mem%   Writes%
i      bar            N/E           0     0.000          1   false           N/E      N/E      50    0.000 B       0     60        90
i      ns1             44       14047   385.961 K        1   false    800.944 MB       54      50   23.557 MB      24     60        90
i      test           N/E           0     0.000          1   false           N/E      N/E      50    0.000 B       0     60        90
Number of rows: 3
```

2 shells run benchmark and watch defrag



Exercise 4 cont..

- Evictions have fragmented the storage.
- Let's look at some info in the logs again about defrag:

```
grep defrag /var/log/aerospike/aerospike.log
```

```
May 30 2015 15:10:10 GMT: INFO (drv_ssd): (drv_ssd.c::2436) device /opt/aerospike/data/
bar.dat: used 774410112, contig-free 687M (687 wblocks), swb-free 10, n-w 0, w-q 0 w-tot 851
(0.0/s), defrag-q 0 defrag-tot 40 (0.0/s)
```

- We have written 851 wblocks, and defragged 40.
- Let's see what happens when we start updating the records in our namespace.
  - Start a Read/Update workload at 50/50 using the Java Benchmark Tool.

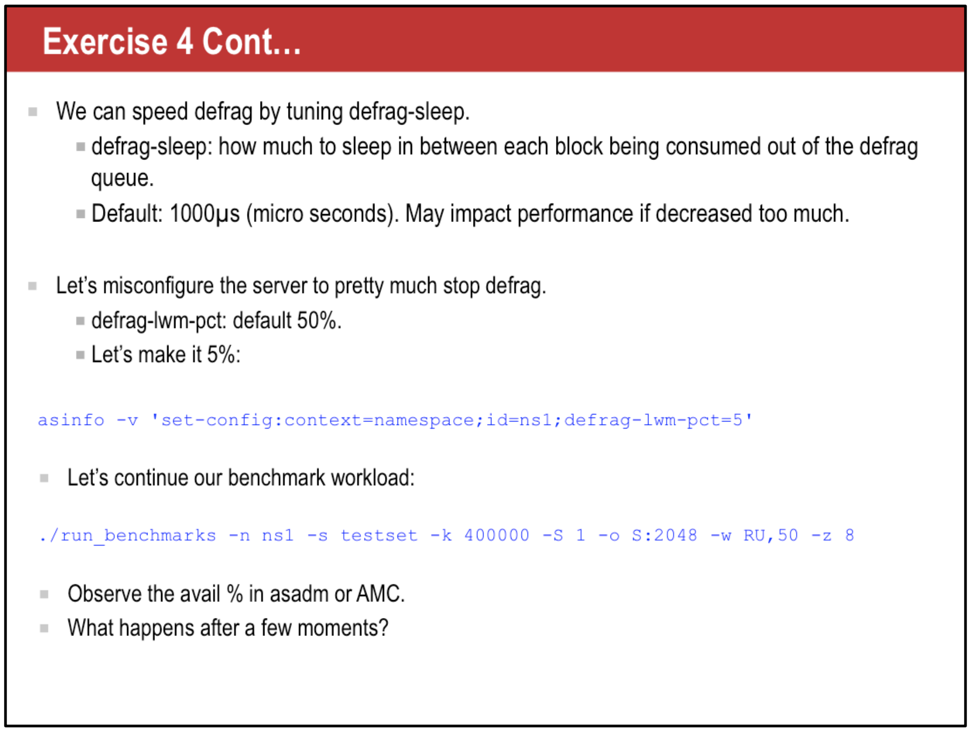
```
./run_benchmarks -n ns1 -s testset -k 400000 -S 1 -o S:2048 -w RU,50 -z 8
```

- Let's look at the logs for speed of writes vs. speed of defrag.

```
May 30 2015 15:14:11 GMT: INFO (drv_ssd): (drv_ssd.c::2436) device /opt/aerospike/data/
bar.dat: used 804317056, contig-free 409M (409 wblocks), swb-free 10, n-w 0, w-q 0 w-tot
2227 (9.2/s), defrag-q 0 defrag-tot 1138 (7.1/s)
```

- At this point in the system this log line is copied from, defrag is not keeping up (9.2 > 7.1).
- Let's stop the benchmark.

In this particular example, defrag may not be keeping up simply because we may not have reached equilibrium yet. So there isn't enough blocks to defrag yet as updates are happening.



Exercise 4 Cont…

- We can speed defrag by tuning defrag-sleep.
  - defrag-sleep: how much to sleep in between each block being consumed out of the defrag queue.
  - Default: 1000µs (micro seconds). May impact performance if decreased too much.

- Let's misconfigure the server to pretty much stop defrag.
  - defrag-lwm-pct: default 50%.
  - Let's make it 5%:

```
asinfo -v 'set-config:context=namespace;id=ns1;defrag-lwm-pct=5'
```

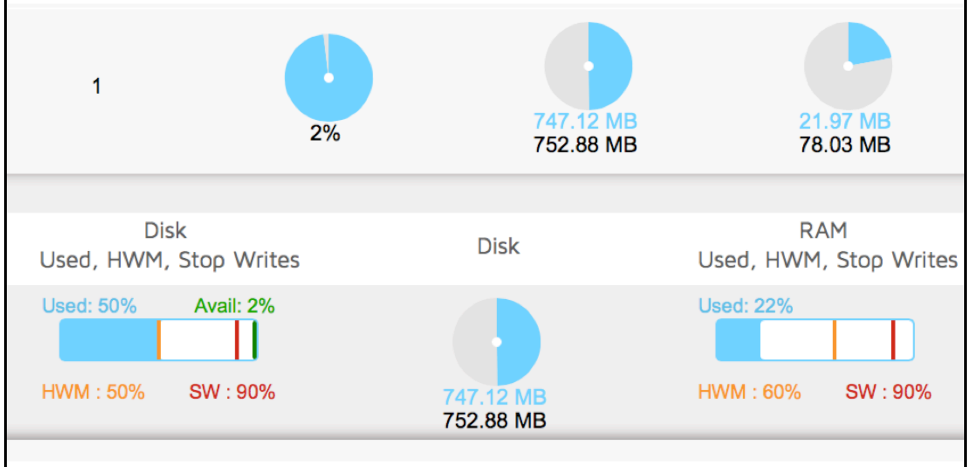
- Let's continue our benchmark workload:

```
./run_benchmarks -n ns1 -s testset -k 400000 -S 1 -o S:2048 -w RU,50 -z 8
```

- Observe the avail % in asadm or AMC.
- What happens after a few moments?

Exercise 5: Stop writes

- We hit stop writes.
- The key value store is now operating in read only mode.
- Notice the errors from the benchmark tool for the writes.



Exercise 5 Cont..

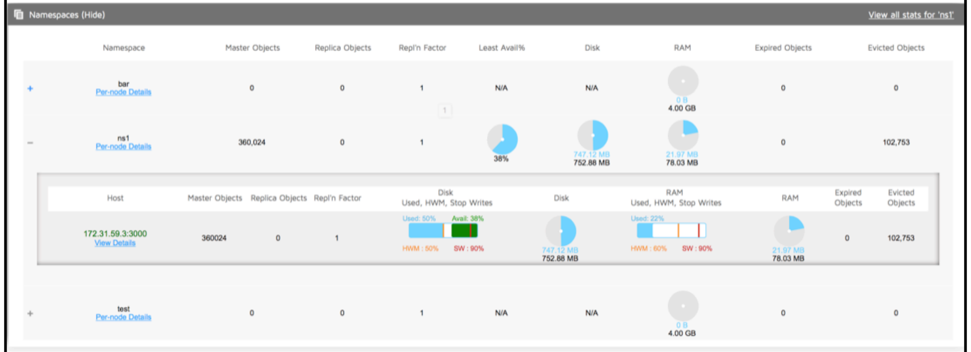
- Same state now from asadm:

```
asadm
Aerospike Interactive Shell, version 0.0.9
Found 1 nodes
Online:  172.31.59.3:3000


Admin> info
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Service Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node   Build   Cluster      Cluster     Cluster   Free   Free   Migrates   Principal     Objects     Uptime
   .       .      Size   Visibility   Integrity   Disk%   Mem%          .           .           .          .
i      3.5.9         1   True         True           50     99   (0,0)      i           360.024 K   14:00:45
Number of rows: 1


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Network Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node               Node                               Fqdn                 Ip   Client     Current       HB        HB
   .                 Id                                  .                  .    Conns        Time     Self   Foreign
i      *BB94FB6A4647106   ip-172-31-59-3.ec2.internal:3000   172.31.59.3:3000       16   170697787   335094         0
Number of rows: 1


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Namespace Information~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Node   Namespace   Avail%   Evictions     Objects     Repl     Stop         Disk    Disk     HWM         Mem     Mem    HWM      Stop
   .           .        .           .           .   Factor   Writes         Used   Used%   Disk%        Used   Used%   Mem%   Writes%
i      bar            N/E           0     0.000            1   false         N/E     N/E      50    0.000 B        0     60        90
i      ns1              2      102753   360.024 K        1   true     747.120 MB      50      50   21.974 MB      22     60        90
i      test           N/E           0     0.000            1   false         N/E     N/E      50    0.000 B        0     60        90
Number of rows: 3
```

Getting out of stop writes

- Let's recover from stop writes
  - Set defrag-lwm-pct back to 50%:

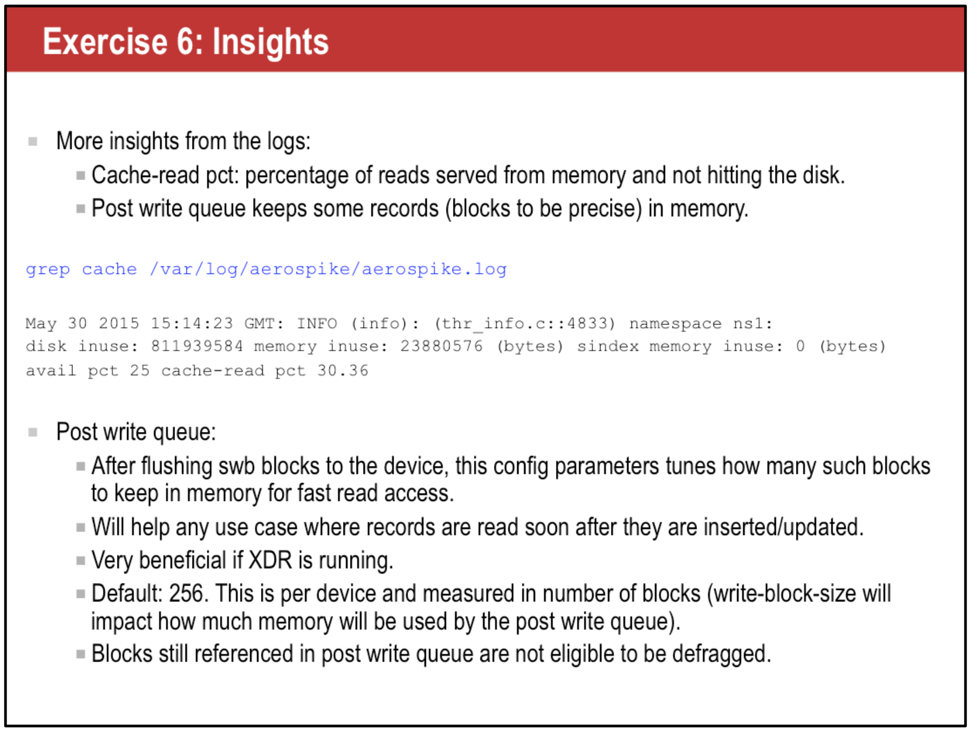
```
asinfo -v 'set-config:context=namespace;id=ns1;defrag-lwm-pct=50'
```



Namespaces (Hide) View all stats for 'ns1'

	Namespace	Master Objects	Replica Objects	Repl'n Factor	Least Avail%	Disk	RAM	Expired Objects	Evicted Objects
+	bar Per-node Details	0	0	1	N/A	N/A	0 B 4.00 GB	0	0
–	ns1 Per-node Details	360,024	0	1	38%	747.12 MB 752.88 MB	21.97 MB 78.03 MB	0	102,753
+	test Per-node Details	0	0	1	N/A	N/A	0 B 4.00 GB	0	0

Host	Master Objects	Replica Objects	Repl'n Factor	Disk Used, HWM, Stop Writes	Disk	RAM Used, HWM, Stop Writes	RAM	Expired Objects	Evicted Objects
172.31.59.3:3000 View Details	360024	0	1	Used: 50% Avail: 38% HWM : 50% SW : 90%	747.12 MB 752.88 MB	Used: 22% HWM : 60% SW : 90%	21.97 MB 78.03 MB	0	102,753



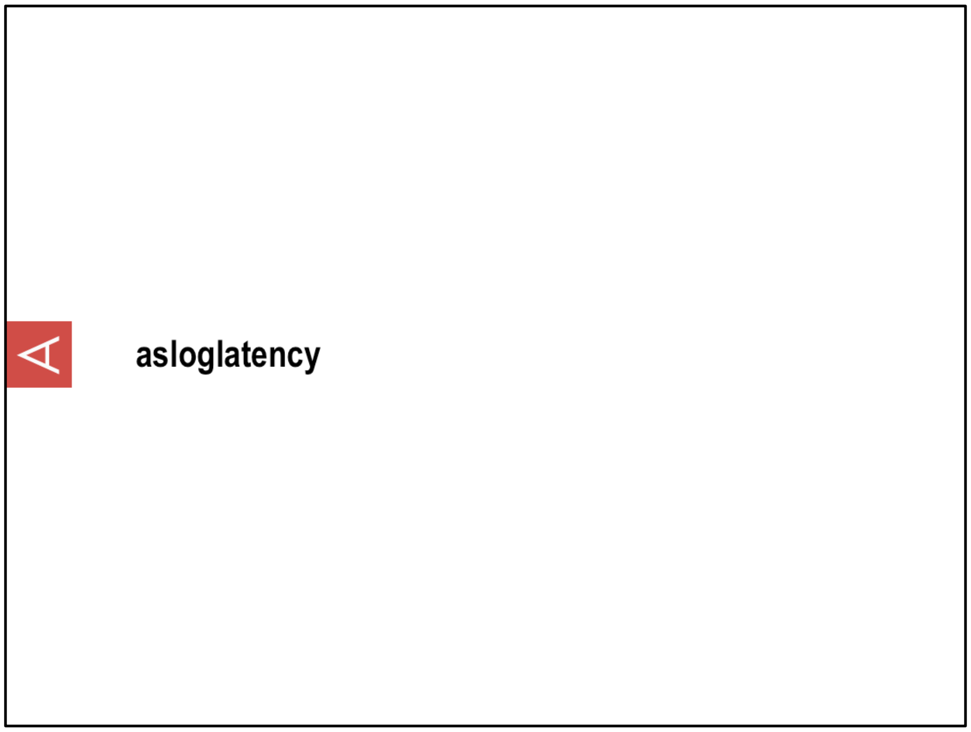
Exercise 6: Insights

- More insights from the logs:
  - Cache-read pct: percentage of reads served from memory and not hitting the disk.
  - Post write queue keeps some records (blocks to be precise) in memory.

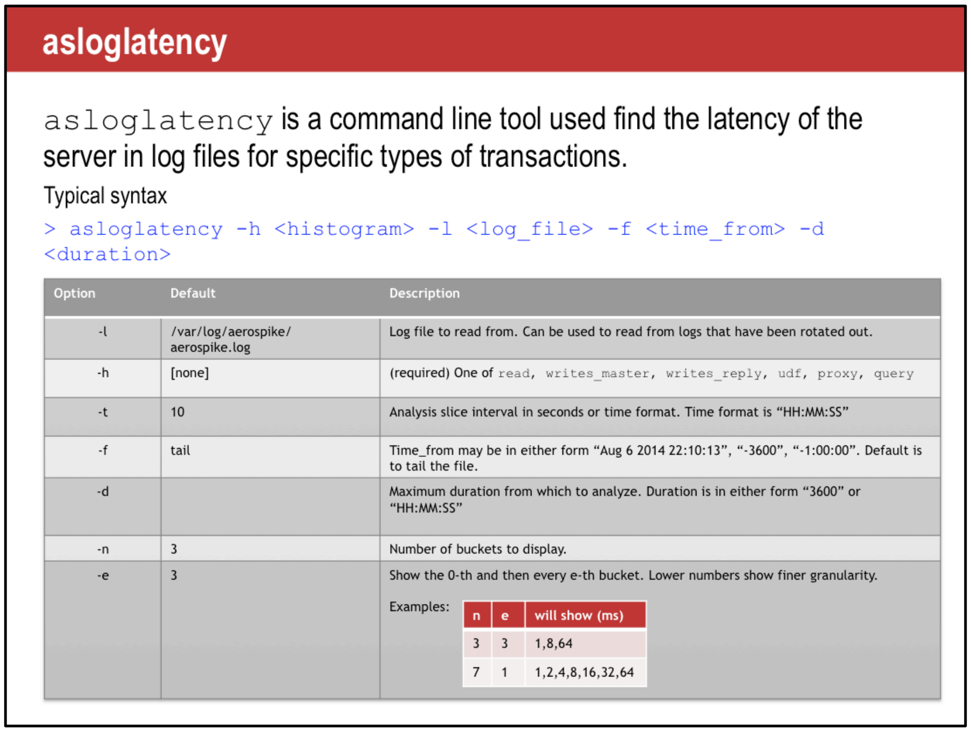
```
grep cache /var/log/aerospike/aerospike.log
```

```
May 30 2015 15:14:23 GMT: INFO (info): (thr_info.c::4833) namespace ns1:
disk inuse: 811939584 memory inuse: 23880576 (bytes) sindex memory inuse: 0 (bytes)
avail pct 25 cache-read pct 30.36
```

- Post write queue:
  - After flushing swb blocks to the device, this config parameters tunes how many such blocks to keep in memory for fast read access.
  - Will help any use case where records are read soon after they are inserted/updated.
  - Very beneficial if XDR is running.
  - Default: 256. This is per device and measured in number of blocks (write-block-size will impact how much memory will be used by the post write queue).
  - Blocks still referenced in post write queue are not eligible to be defragged.



asloglatency



asloglatency

`asloglatency` is a command line tool used find the latency of the server in log files for specific types of transactions.

Typical syntax

```
> asloglatency -h <histogram> -l <log_file> -f <time_from> -d
<duration>
```

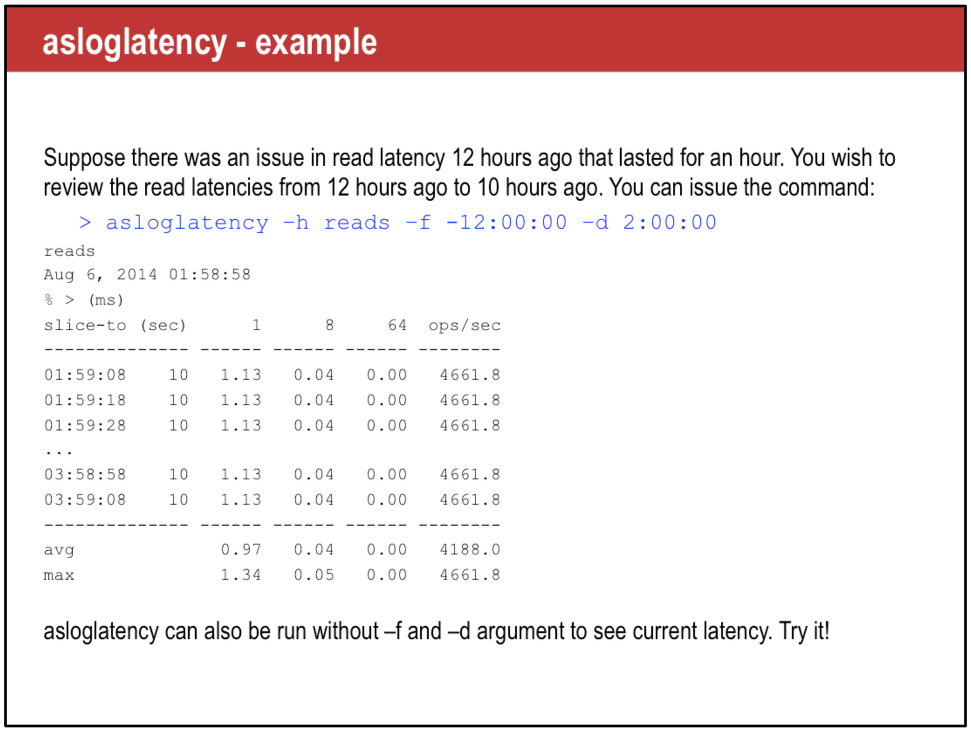
Option	Default	Description
-l	/var/log/aerospike/aerospike.log	Log file to read from. Can be used to read from logs that have been rotated out.
-h	[none]	(required) One of `read, writes_master, writes_reply, udf, proxy, query`
-t	10	Analysis slice interval in seconds or time format. Time format is "HH:MM:SS"
-f	tail	Time_from may be in either form "Aug 6 2014 22:10:13", "-3600", "-1:00:00". Default is to tail the file.
-d		Maximum duration from which to analyze. Duration is in either form "3600" or "HH:MM:SS"
-n	3	Number of buckets to display.
-e	3	Show the 0-th and then every e-th bucket. Lower numbers show finer granularity. Examples: see below

Examples:

n	e	will show (ms)
3	3	1,8,64
7	1	1,2,4,8,16,32,64

asloglatency will show the latencies taken from log files. These may be a considerable time in the past. This is very useful for seeing:

- when a problem started
- did the problem occur suddenly or over a long period of time



asloglatency - example

Suppose there was an issue in read latency 12 hours ago that lasted for an hour. You wish to review the read latencies from 12 hours ago to 10 hours ago. You can issue the command:

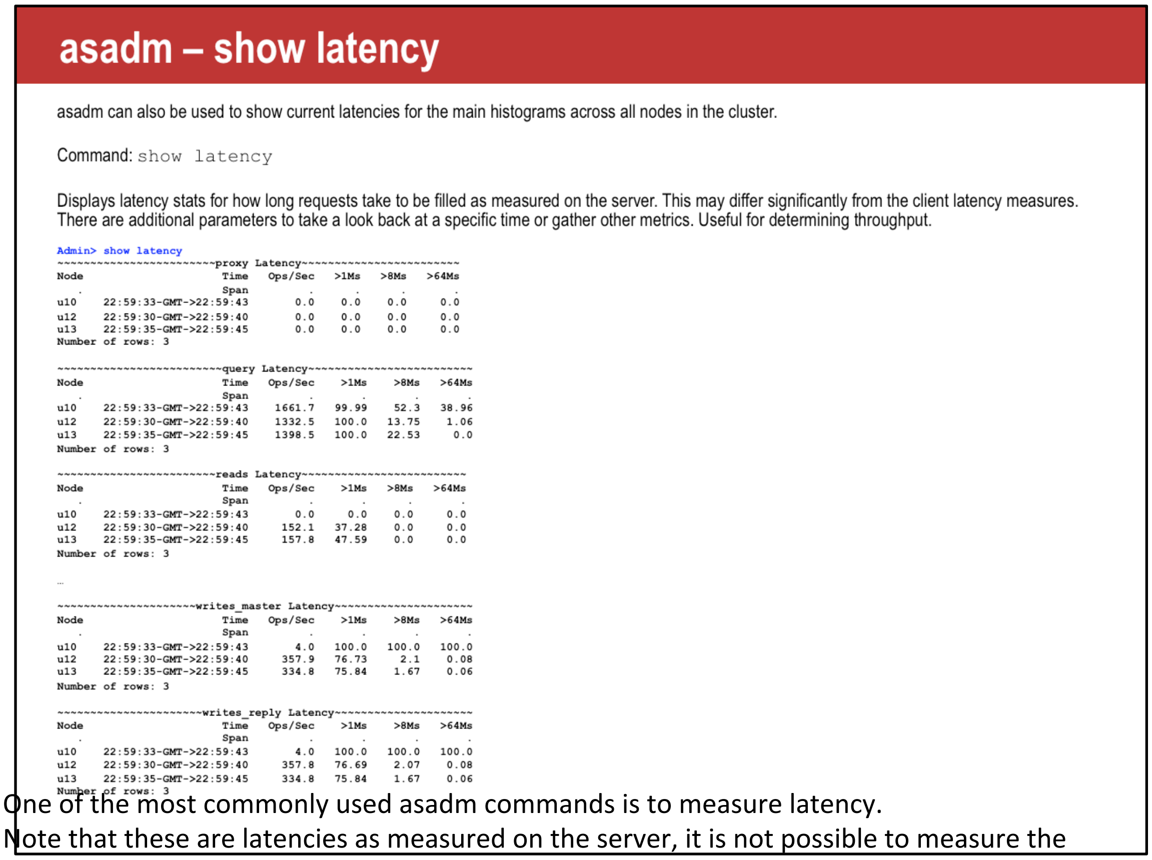
```
   > asloglatency -h reads -f -12:00:00 -d 2:00:00
reads
Aug 6, 2014 01:58:58
% > (ms)
slice-to (sec)      1      8     64  ops/sec
-------------- ------ ------ ------ --------
01:59:08    10   1.13   0.04   0.00   4661.8
01:59:18    10   1.13   0.04   0.00   4661.8
01:59:28    10   1.13   0.04   0.00   4661.8
...
03:58:58    10   1.13   0.04   0.00   4661.8
03:59:08    10   1.13   0.04   0.00   4661.8
-------------- ------ ------ ------ --------
avg              0.97   0.04   0.00   4188.0
max              1.34   0.05   0.00   4661.8
```

asloglatency can also be run without –f and –d argument to see current latency. Try it!

asloglatency – micro / storage benchmarks

Details on our documentation site:

http://www.aerospike.com/docs/tools/asloglatency/



asadm – show latency

asadm can also be used to show current latencies for the main histograms across all nodes in the cluster.

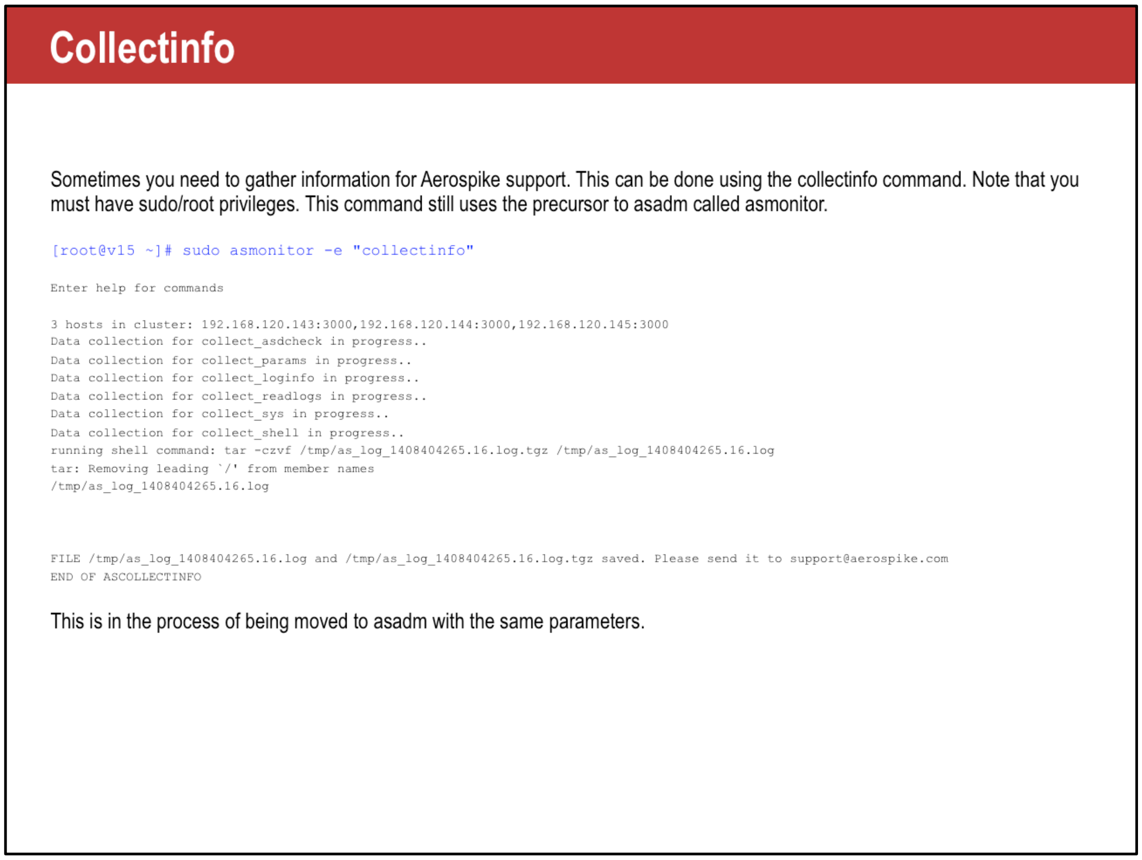
Command: `show latency`

Displays latency stats for how long requests take to be filled as measured on the server. This may differ significantly from the client latency measures. There are additional parameters to take a look back at a specific time or gather other metrics. Useful for determining throughput.

```
Admin> show latency
~~~~~~~~~~~~~~~~~~~~~~~~proxy Latency~~~~~~~~~~~~~~~~~~~~~~~~
Node                        Time   Ops/Sec   >1Ms   >8Ms   >64Ms
   .                        Span         .      .      .       .
u10    22:59:33-GMT->22:59:43        0.0    0.0    0.0     0.0
u12    22:59:30-GMT->22:59:40        0.0    0.0    0.0     0.0
u13    22:59:35-GMT->22:59:45        0.0    0.0    0.0     0.0
Number of rows: 3

~~~~~~~~~~~~~~~~~~~~~~~~~query Latency~~~~~~~~~~~~~~~~~~~~~~~~~
Node                        Time   Ops/Sec    >1Ms    >8Ms   >64Ms
   .                        Span         .       .       .       .
u10    22:59:33-GMT->22:59:43     1661.7   99.99    52.3   38.96
u12    22:59:30-GMT->22:59:40     1332.5   100.0   13.75    1.06
u13    22:59:35-GMT->22:59:45     1398.5   100.0   22.53     0.0
Number of rows: 3

~~~~~~~~~~~~~~~~~~~~~~~~reads Latency~~~~~~~~~~~~~~~~~~~~~~~~~
Node                        Time   Ops/Sec    >1Ms   >8Ms   >64Ms
   .                        Span         .       .      .       .
u10    22:59:33-GMT->22:59:43        0.0     0.0    0.0     0.0
u12    22:59:30-GMT->22:59:40      152.1   37.28    0.0     0.0
u13    22:59:35-GMT->22:59:45      157.8   47.59    0.0     0.0
Number of rows: 3

…

~~~~~~~~~~~~~~~~~~~~writes_master Latency~~~~~~~~~~~~~~~~~~~~
Node                        Time   Ops/Sec    >1Ms    >8Ms   >64Ms
   .                        Span         .       .       .       .
u10    22:59:33-GMT->22:59:43        4.0   100.0   100.0   100.0
u12    22:59:30-GMT->22:59:40      357.9   76.73     2.1    0.08
u13    22:59:35-GMT->22:59:45      334.8   75.84    1.67    0.06
Number of rows: 3

~~~~~~~~~~~~~~~~~~~~~writes_reply Latency~~~~~~~~~~~~~~~~~~~~
Node                        Time   Ops/Sec    >1Ms    >8Ms   >64Ms
   .                        Span         .       .       .       .
u10    22:59:33-GMT->22:59:43        4.0   100.0   100.0   100.0
u12    22:59:30-GMT->22:59:40      357.8   76.69    2.07    0.08
u13    22:59:35-GMT->22:59:45      334.8   75.84    1.67    0.06
Number of rows: 3
```

One of the most commonly used asadm commands is to measure latency.
Note that these are latencies as measured on the server, it is not possible to measure the client latencies from the Aerospike nodes. This command also shows the throughput for each node/type.
This command gives you the latencies for all nodes in the cluster for different measures:

writes_master: These are the latency times for responds to writes from the master. Unless you have actively configured for asynchronous writes, this will be the same as the latency to any replica.
writes_reply: These are the latency times for replica writes. This is normally the same as for writes_master, unless you have configured differently.
reads: These are the latency times for reads. Aerospike does reads from a single node.
udf: The latency times for UDFs to run.
proxy: In cases where the cluster state is dynamic (nodes added/removed) it is possible that the node not have the data. Aerospike will automatically proxy the request for the client. These latency times are just for proxied requests.
query: The latency times for queries using secondary indexes.



Collectinfo

Sometimes you need to gather information for Aerospike support. This can be done using the collectinfo command. Note that you must have sudo/root privileges. This command still uses the precursor to asadm called asmonitor.

```
[root@v15 ~]# sudo asmonitor -e "collectinfo"

Enter help for commands

3 hosts in cluster: 192.168.120.143:3000,192.168.120.144:3000,192.168.120.145:3000
Data collection for collect_asdcheck in progress..
Data collection for collect_params in progress..
Data collection for collect_loginfo in progress..
Data collection for collect_readlogs in progress..
Data collection for collect_sys in progress..
Data collection for collect_shell in progress..
running shell command: tar -czvf /tmp/as_log_1408404265.16.log.tgz /tmp/as_log_1408404265.16.log
tar: Removing leading `/' from member names
/tmp/as_log_1408404265.16.log


FILE /tmp/as_log_1408404265.16.log and /tmp/as_log_1408404265.16.log.tgz saved. Please send it to support@aerospike.com
END OF ASCOLLECTINFO
```

This is in the process of being moved to asadm with the same parameters.