



Support Tools

swarmctl and swarmrestart

DATACORE CONFIDENTIAL

— — — — ● The Authority on Software-Defined Storage

What We Will Cover

- snmp-castor-tool.sh
- swarmctl
- swarmrestart

snmp-castor-tool.sh

- snmp-castor-tool.sh has been around for many years
- uses SNMP for most operations (although not all)
- swiss army knife for many support operations (38 different option flags!)
- can query the cluster AND perform operations against the cluster
- can change cluster settings in real-time
- can rolling reboot a cluster
- can collect snmp oid output
- relies on SNMP which is very slow, especially in larger clusters
- snmp could eventually be removed from Swarm in favor of the better tools

swarmctl overview

- like snmp-castor-tool.sh, swarmctl lives in the support tools bundle (<https://support.cloud.caringo.com/tools/caringo-support-tools.tgz>)
- uses the management API for all operations
- swiss army knife for many support operations
- can query the cluster AND perform operations against the cluster
- can change cluster settings in real-time
- can't rolling reboot a cluster (see the swarmrestart script)
- does not collect snmp oid output
- relies on the mgmt api which is very fast compared to SNMP
- this is the go-forward support tool for the majority of support operations (Swarm 10+)

swarmctl overview

- if SCSP_HOST is set in the environment, -d [ip address] is not necessary
- the -x option is commonly used to get more output, in some cases MUCH more
- the -x option is used to output to a csv file automatically for more analysis
- by default, will only take action or get information from the node specified in SCSP_HOST or -d [ip address] options.
- -a takes action or gets information from ALL nodes in the cluster as seen in the cluster status page
- -a is not necessary to change a value that affects the PSS as the PSS is automatically read by all nodes
- -n is used to take action or query node IPs specified in a file you create called NODES.csv in the directory from which you run the swarmctl script (same note regarding -a applies here- this works the same as in the snmp-castor-tool.sh script)
- use -j to remove the text table lines to declutter the output
- -h is used for help (usage)
- most query output will show you the default value, the current value, whether it is changeable, and scope of the setting's effect (node/ cluster/ etc)

swarmctl -A Announcements

- swarmctl -A [show|clear] shows (default) and allows you to clear announcements. Add -a for all nodes.

```
root@c-csn1:~/tmp>~/Support/swarmctl -A
+-----+-----+-----+
| Node           | Timestamp           | Message                                     |
+-----+-----+-----+
| 192.168.201.85 | 2020-03-20T17:39:42.242303Z | Node is ready for client activity         |
+-----+-----+-----+
| 192.168.201.85 | 2020-03-20T17:39:42.073753Z | Persistent settings propagation succeeded. UUID: |
|                |                      | 03fd85d03e861697e81337eaf2b1af85        |
+-----+-----+-----+
| 192.168.201.85 | 2020-03-20T17:39:42.056902Z | Added new feed id=0                       |
+-----+-----+-----+
| 192.168.201.85 | 2020-03-20T17:39:42.018022Z | Set log.level = 30 (via cluster settings)   |
+-----+-----+-----+
| 192.168.201.85 | 2020-03-20T17:39:42.015807Z | Set feeds.definition = {0: {'destination': {'insertBatchTimeout': |
|                |                      | 1, 'indexAlias': 'c-csn1.enfield.com0', 'insertBatchSize': 100, |
|                |                      | 'host': '192.168.201.203', 'fullMetadata': 1, 'port': 9200}, |
|                |                      | 'type': 'Indexing', 'name': 'IndexerFeed-5.6.12-to-c- |
|                |                      | csn1-indexer2-3', 'lastchanged': 'Tue, 10 Mar 2020 16:03:57 |
|                |                      | GMT'}} (via cluster settings)             |
+-----+-----+-----+
| 192.168.201.85 | 2020-03-20T17:39:42.014194Z | Read new feeds definition from cluster settings stream |
+-----+-----+-----+
| 192.168.201.85 | 2020-03-20T17:39:19.836679Z | Available: 53.65 million index slots, 17.49 GB capacity |
+-----+-----+-----+
| 192.168.201.85 | 2020-03-20T17:39:17.209106Z | Mounted /dev/sdd(mechanical), volumeID is |
|                |                      | a18e8592836fd39d1da925de6a4f1e4c        |
+-----+-----+-----+
root@c-csn1:~/tmp>~/Support/swarmctl -A clear
Success: 10 announcementsgs cleared on 192.168.201.85
root@c-csn1:~/tmp>
```

swarmctl -b Largest streams

- `swarmctl -b` shows the biggest streams and their rep counts. Add `-a` for all nodes. `-x` for much more output to file.
- If the largest stream on a particular disk is a segment, that is noted as “seg”
- In the graphic, `/dev/sda` is retired

```
root@csn1:~/tmp>~/Support/swarmctl -b
```

nodeIPAddress	name	availPercent	streamCount	volErrs	maxSpace	largestStreamSize	largestStreamUuid	largeUuidRepCount
192.168.201.85	/dev/sda	0%	0	0	0.00MB	0		-
192.168.201.85	/dev/sdb	63%	21,560	0	10.12GB	102	7443667259088a0abc171f06b06f41b3	seg
192.168.201.85	/dev/sdc	61%	8,668	0	10.12GB	102	20286435ed16148ec949468876b214fe	seg
192.168.201.85	/dev/sdd	50%	6,196	0	10.12GB	1048	b191f88b765f52ee04293d54402c1101	3

swarmctl -C See and modify configuration

- `swarmctl -C [option]` this shows a particular mgmt api configuration endpoint- think OID in snmp parlance
- By default, this shows a single node's current value... `-a` to see all nodes' values.
- Add `-V [option]` to change the value if Readonly is False.
- By far one of the most commonly used flags.

```
root@c-sn1:~/tmp>swarmctl -C disk.obsoleteTimeout
```

Node	Setting	Value	Changed By	Scope	Readonly
192.168.201.85	disk.obsoleteTimeout	1209600		cluster	False

```
root@c-sn1:~/tmp>swarmctl -C disk.obsoleteTimeout -V 604800
```

Node	Setting	Value	Changed By	Previous Value	Prev Changed By	Scope	Readonly
192.168.201.85	disk.obsoleteTimeout	604800	Management API	1209600		cluster	False

swarmctl -d Specify the node to query

- `swarmctl -d [ip address or DNS name]`
- this option is used in conjunction with other options
- this option specifies the initial IP address to use when querying the cluster.
- `swarmctl` can use this IP address to collect the complete list of IPs in the cluster
- the complete list of IPs can be queried using `-a` once `-d [ip]` has been specified
- this option is not necessary if using the `-n` option
- you can forgo this option if you set `SCSP_HOST` in your environmental variables (which is why the examples in this deck are all missing the `-d [ip]` options)

```
root@c-sn1:~/tmp>echo $SCSP_HOST  
192.168.201.85
```

swarmctl -D Query and control drive lights

- `swarmctl -D [{on,off,1,2,5,10,25,50}]`
- this option is used to turn on and off the drive lights on a particular node/ disk
- the number variables indicate number of minutes to turn the light on, or set “on”
- use with `-V [drive]` to specify a particular drive

```
root@c-csn1:~/tmp>swarmctl -D 1 -V /dev/sda
API reports success setting drive light /dev/sda on 192.168.201.85 to on with timeout: 1
```

swarmctl -E Errors

- `swarmctl -E [show|clear]` shows (default) and allows you to clear errors. Add `-a` for all nodes.
- this is similar to `-A`, except for errors instead of announcements

swarmctl -e HP cycle information

- swarmctl -e shows you the current, ongoing HP cycle information.
- most commonly used with -x to export much more information for analysis

```
root@csn1:~/tmp>swarmctl -e
```

nodeIPAddress	HP Exam queue count	HP Replication queue count	HP State	HP ongoing cycle: Cycle number	HP ongoing cycle: Streams examined	HP last cycle: Streams examined	HP ongoing cycle, whole reps: Trims requested	HP last cycle, whole reps: Trims requested
192.168.201.85	0	0	running (4)	2	1,048	24,425	82	3491

swarmctl -F format stale disks

- `swarmctl -F [stale volume]`
- only applies to disks that have been marked as stale (offline for more than 2 weeks by default)
- leave off the volume option in order to format all stale disks on the node (likely the more common option)
- this prevents you from having to go to the terminal console, stop the storage processes, format the drives, and then reboot the chassis

swarmctl -i Log level

- `swarmctl -i [{0,5,10,15,20,30,40,50}]`
- display the log level (without variable) or use a variable to change the log level for all nodes in the cluster

```
root@c-sn1:~/tmp>swarmctl -i
+-----+-----+-----+-----+-----+-----+
| Node   | Setting | Value | Changed By | Scope | Readonly |
+-----+-----+-----+-----+-----+-----+
| 192.168.201.85 | log.level | 30 | cluster settings | cluster | False |
+-----+-----+-----+-----+-----+-----+
root@c-sn1:~/tmp>swarmctl -i 20
+-----+-----+-----+-----+-----+-----+-----+-----+
| Node   | Setting | Value | Changed By | Previous Value | Prev Changed By | Scope | Readonly |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 192.168.201.85 | log.level | 20 | Management API | 30 | cluster settings | cluster | False |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

swarmctl -l Remount stale volume

- `swarmctl -l [stale-vol-to-mount [stale-vol-to-mount ...]]`
- leave blank to apply this to every stale disk on the chassis
- this option allows you to remount the stale volumes as opposed to -F which allows you to reformat stale volumes
- if you are sure the data on the disks is valid and necessary, use this option to resurrect those streams
- this will likely cause over-replication as the streams likely have already been recovered in the 2+ weeks since the drives were not stale.

swarmctl -k

Kernel modules

- `swarmctl -k` will show the loaded kernel modules
- this is a lot of output- use with `-x` to output to a file, otherwise it is unusable
- partial output below

```
root@csnl:~/tmp>swarmctl -k
Found 511 module descriptions.
```

nodeIPAddress	name	description	filename	version
192.168.201.85	acpi_cpufreq	ACPI Processor P-States Driver	/lib/modules/4.19.56/kernel/drivers/cpufreq/acpi-cpufreq.ko	
192.168.201.85	ahci_platform	AHCI SATA platform driver	/lib/modules/4.19.56/kernel/drivers/ata/ahci_platform.ko	
192.168.201.85	tpm_tis	TPM Driver	/lib/modules/4.19.56/kernel/drivers/char/tpm/tpm_tis.ko	2
192.168.201.85	acpi_power_meter	ACPI 4.0 power meter driver	/lib/modules/4.19.56/kernel/drivers/hwmon/acpi_power_meter.ko	
192.168.201.85	amd_xgbe	AMD 10 Gigabit Ethernet Driver	/lib/modules/4.19.56/kernel/drivers/net/ethernet/amd/xgbe/amd-xgbe.ko	1.0.3

swarmctl -K Display all cluster settings

- `swarmctl -K` shows all currently active configuration parameters in the cluster!
- this shows you the scope of all of the options, where they were set, whether you can change them dynamically
- use with `-x` to export to file for ease of use

```
root@c-snl1:~/tmp>swarmctl -K
Found 93 module descriptions.
```

nodeIPAddress	name	value	scope	modified-by	readonly
192.168.201.85	cache.expirationTime	600	node		writeable
192.168.201.85	cache.maxCacheableSize	1048576	node		writeable
192.168.201.85	cache.percentage	10	node		writeable
192.168.201.85	cache.realmStaleTimeout	600	node		writeable
192.168.201.85	cip.group	224.0.10.100	cluster	config file	readonly
192.168.201.85	cip.queryTimeout	0.030	node		writeable
192.168.201.85	cip.readBufferSize	1048576	node		writeable

swarmctl -L Node log level

- `swarmctl [-L {0,5,10,15,20,30,40,50}]` Allows you to view and change a particular node's log level.
- This is useful when you want to change only a single node's log level instead of EVERY node's log level
- This value does not persist after a reboot

```
root@c-csn1:~/tmp>swarmctl -L 30 -d 192.168.201.85
```

Node	Setting	Value	Changed By	Previous Value	Prev Changed By	Scope	Readonly
192.168.201.85	log.nodeLogLevel	30	Management API	0	Management API	node	False

swarmctl -m detailed running statistics

- `swarmctl [-m {commstats,hpstats,networktest,meminfo,features}]`
- this option includes all kinds of details statistics from a running cluster
- typically used with `-x` to output automatically to a file where you will see much more output

swarmctl -m commstats

- swarmctl -m commstats
- shows bidding and histogram information
- typically used with -a AND -x to output all nodes' information automatically to a file where you will see much more output

```
root@ec-csni:~/tmp>swarmctl -g -a
```

nodeIPAddress	SCSP: Last read bid	SCSP: Last rep bid	SCSP: Last write bid	Response Histogram: Maximum (ms)	Response Histogram: Mean (ms)	Response Histogram: Minimum (ms)	Response Histogram: Tail (ms)
192.168.201.84	19	25	34	95	1	0	70
192.168.201.85	6	27	29	114	1	0	90
192.168.201.86	100	100	255	21	3	1	30
192.168.201.88	19	26	28	136	1	0	140

swarmctl -m hpstats

- swarmctl -m hpstats
- used to show health processor statistics including the current ongoing cycle

```
root@c-snl:~/tmp>swarmctl -m hpstats -a
```

nodeIPAddress	HP Exam queue count	HP Replication queue count	HP State	HP ongoing cycle: Cycle number	HP ongoing cycle: Streams examined	HP last cycle: Streams examined	HP ongoing cycle, whole reps: Trims requested	HP last cycle, whole reps: Trims requested
192.168.201.84	0	0	idle (3); running (1)	158	39,882	41,997	0	0
192.168.201.85	0	0	idle (3); running (1)	113	43,201	44,635	0	0
192.168.201.86	0	0	initializing	0	0	0	0	0
192.168.201.88	0	0	running (2); idle (2)	153	28,467	37,776	0	0

swarmctl -m networktest

- swarmctl -m networktest
- this starts a network connectivity test and can take quite some time so use with caution
- typically used with -a output all nodes' information automatically to a file where you will see much more output.
-x is not required to output to a file with this option as that is the only method of output

```
root@c-csn1:~/tmp>swarmctl -m networktest -a
Note: running a networktest may take a long time.
networktest output for c-csn1.enfield.com is in 2020_0323_1019-networktest.csv in this directory
```

```
root@c-csn1:~/tmp>cat 2020_0323_1019-networktest.csv
nodeIPAddress,nodeId,1:nodeIp,1:tcp:reps,1:tcp:responseTime,1:udp:reps,1:udp:responseTime,2:nodeIp,2:tcp:reps,2:tcp:responseTime,2:udp:reps,2:udp:responseTime,3:nodeIp,3:tcp:reps,3:tcp:responseTime,3:udp:reps,3:udp:responseTime,4:nodeIp,4:tcp:reps,4:tcp:responseTime,4:udp:reps,4:udp:responseTime
192.168.201.84,c038e3e20bd1e422,192.168.201.84,100,2.67790150642395,100,0.05864691734313965,192.168.201.85,100,3.777550220489502,95,0.10914874076843262,192.168.201.86,100,2.612891435623169,100,0.06607747077941895,192.168.201.88,100,2.535768508911133,100,0.06617021560668945
192.168.201.85,b9248f679cadd114,192.168.201.84,100,2.7409722805023193,100,0.08721709251403809,192.168.201.85,100,3.7966408729553223,100,0.05715632438659668,192.168.201.86,100,2.7524921894073486,100,0.0642538070678711,192.168.201.88,100,2.624887466430664,100,0.06181812286376953
192.168.201.86,bc4c1f4dbf1a8805,192.168.201.84,100,2.8358359336853027,100,0.06846165657043457,192.168.201.85,100,4.392239809036255,99,0.0865170955657959,192.168.201.86,100,2.818005084991455,100,0.044764041900634766,192.168.201.88,100,2.6616790294647217,100,0.05435681343078613
192.168.201.88,2ac34b8eb78dad05,192.168.201.84,100,2.9755702018737793,100,0.0733039379119873,192.168.201.85,100,4.481632471084595,99,0.10448074340820312,192.168.201.86,100,2.62528920173645,100,0.07231974601745605,192.168.201.88,100,2.668856382369995,100,0.0669243335723877
```

swarmctl -m meminfo

- swarm -m meminfo
- shows memory information including index and overlay index
- typically used with -a AND -x to output all nodes' information automatically to a file

```
root@c-snl:~/tmp>swarmctl -m meminfo -a
```

nodeIPAddress	accountedMemoryHighwater	accountedMemoryInUse	accountedMemoryUtilization	indexSlotsAvailable	indexUtilization	ioBufferMemory	overlayIndex	overlaySlotsTotal	overlaySlotsUsed	overlayStatus	overlayTotalMem
192.168.201.84	1041890	1041890	0	53.60mil	0	1.42GB	1	53.69mil	15,479	authoritative	2.17GB
192.168.201.85	1045386	1043247	0	53.59mil	0	1.42GB	1	53.69mil	15,830	authoritative	2.17GB

swarmctl -m features

- swarm -m features
- numbers of interesting types of requests like md5, if-match, integrity seal, rename, and versioning
- typically used with -a AND -x to output all nodes' information automatically to a file where you will see much more output

```
root@csnl:~/tmp>swarmctl -m features -a
```

nodeIPAddress	Feature: Number of MD5 requests	Feature: Number of if-match requests	Feature: Number of integrity seal requests	Feature: Number of rename requests	Feature: Number of versioning requests
192.168.201.84	12266	5	0	0	0
192.168.201.85	18921	2	0	0	0
192.168.201.86	1987	2	0	0	0
192.168.201.88	11125	3	0	0	0

swarmctl -O SCSP Response Counts

- `swarmctl -O` shows all of the SCSP response codes that a node has processed
- typically used with `-a` AND `-x` to output all nodes' information automatically to a file
- you can see below that since I have been using `.85` to send these example commands to, that 200 OK is much higher on that node than on other nodes

```
root@c-snl:~/tmp>swarmctl -a -O
```

nodeIPAddress	200	201	202	206	301	304	400	401	404	410	412	4xx	500	503	507	5xx
192.168.201.84	258	0	0	0	334	0	0	0	3	0	0	0	0	0	0	0
192.168.201.85	2,924	6	0	0	12	0	0	1	0	0	0	0	0	0	0	0
192.168.201.86	102	2	0	0	397	0	0	2	5	0	0	0	0	0	0	0
192.168.201.88	296	0	0	0	344	0	0	0	3	0	0	0	0	0	0	0

swarmctl -p specify user/ password for admin access

- `swarmctl -p [user:password]`
- the commands run previously in this deck worked ONLY because we try to use 2 different default passwords if the `-p` option is not specified
- if the admin password for the cluster is “ourpwdofchoicehere” or “caringo”, then `swarmctl` doesn’t require you to include `-p [user:password]` for commands that make changes or otherwise need admin rights
- if you have a non-default password, which is the better strategy, then simply add `-p [user:password]` with your commands for authentication.

swarmctl -P Persistent Settings Stream (PSS)

- `swarmctl -P` export the Persistent Settings Stream to standard out
- requires admin credentials (-p user:pass) - see note on -p option.
- use -x to export to a file
- doesn't support -a - if you need the PSS from multiple nodes, use -d [ip]

```
root@c-csn1:~/tmp>swarmctl -P
Successfully retrieved cluster persistent settings stream from 03fd85d03e861697e81337eaf2b1af85
recovery.volMaintenanceInterval=10800
scsp.autoRecursiveDelete=1
log.forceAudit=0
policy.eCEncoding=unspecified anchored
health.iterTasks=2
cip.queryRetryMultiplier=1.00000
metrics.target=
ec.segmentConsolidationFrequency=10
power.cap=100
policy.eCMinStreamSize=1Mb anchored
```

swarmctl -q Quick summary

- `swarmctl -q` get a quick summary of the cluster status
- what you might see from the cluster status page on port 90

```
root@c-snl:~/tmp>swarmctl -q
```

Type	name	status	availPerc	usedSpace	maxSpace	streamCount	swVer	errCount	volErrs
Cluster	c-csn1.enfield.com	ok	62%	28.89GB	121.43GB	120,035	11.0.3	0	0
Subcluster	default	ok	62%	28.89GB	121.43GB	120,035	11.0.3	0	0
Chassis	192.168.201.85	ok	75%	9.91GB	40.48GB	44,364	11.0.3	0	0
Chassis	192.168.201.84	ok	76%	9.42GB	40.48GB	37,915	11.0.3	0	0
Chassis	192.168.201.86	retired	0%	0.00MB	0.00MB	0	11.0.3	0	0
Chassis	192.168.201.88	ok	76%	9.56GB	40.48GB	37,756	11.0.3	0	0

swarmctl -Q dmesg/ hwinfo/ healthreport

- swarmctl -Q
- allows options for exporting dmesg, hwinfo, and the health report from a single node
- this replaces other scripts like hwinfo-dmesg-grab.sh and collect_health_reports.sh
- without a modifier, assumes “dmesg” by default

```
root@csn1:~/tmp>swarmctl -Q
Node 192.168.201.85 dmesg
[Fri Mar 20 20:33:21 2020] Linux version 4.19.56 (root@ad06e4fd7169) (gcc version 6.3.0 20170516 (Debian 6.3.0-18+deb9u1)) #1 SMP Wed Nov 13 20:02:43 UTC 2019
[Fri Mar 20 20:33:21 2020] Command line: initrd=fsimage/000c292422d5 ramdisk_size=200000 root=/dev/ram0 castor_cfg=http://192.168.201.3:8088/config?mac=000c292422d5 castor_net=balance-alb: BOOT_IMAGE=kerne
L/000c292422d5
[Fri Mar 20 20:33:21 2020] Disabled fast string operations
[Fri Mar 20 20:33:21 2020] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[Fri Mar 20 20:33:21 2020] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[Fri Mar 20 20:33:21 2020] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[Fri Mar 20 20:33:21 2020] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[Fri Mar 20 20:33:21 2020] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[Fri Mar 20 20:33:21 2020] BIOS-provided physical RAM map:
[Fri Mar 20 20:33:21 2020] BIOS-e820: [mem 0x0000000000000000-0x0000000000008dfff] usable
[Fri Mar 20 20:33:21 2020] BIOS-e820: [mem 0x0000000000008e000-0x0000000000009ffff] reserved
```

swarmctl -Q dmesg

- swarmctl -Q dmesg
- export dmesg output
- most commonly used with -x to export output to a file
- can use with -a to grab the dmesg from ALL nodes in the cluster

```
root@c-csn1:~/tmp>swarmctl -Q dmesg -x
Node 192.168.201.85 dmesg
dmesg for 192.168.201.85 written to 2020_0323_1449-dmesg-192.168.201.85.txt
root@c-csn1:~/tmp>swarmctl -Q dmesg -x -a
Node 192.168.201.84 dmesg
dmesg for 192.168.201.84 written to 2020_0323_1450-dmesg-192.168.201.84.txt
Node 192.168.201.85 dmesg
dmesg for 192.168.201.85 written to 2020_0323_1450-dmesg-192.168.201.85.txt
Node 192.168.201.86 dmesg
dmesg for 192.168.201.86 written to 2020_0323_1450-dmesg-192.168.201.86.txt
Node 192.168.201.88 dmesg
dmesg for 192.168.201.88 written to 2020_0323_1450-dmesg-192.168.201.88.txt
```

swarmctl -Q hwinfo

- swarmctl -Q hwinfo
- results in hwinfo output - can take some time to run
- typically run with -x -a to get all nodes' output to a file

```
root@c-csn1:~/tmp>swarmctl -Q hwinfo -x -a
Node 192.168.201.84 hwinfo
hwinfo for 192.168.201.84 written to 2020_0323_1453-hwinfo-192.168.201.84.txt
Node 192.168.201.85 hwinfo
hwinfo for 192.168.201.85 written to 2020_0323_1453-hwinfo-192.168.201.85.txt
Node 192.168.201.86 hwinfo
hwinfo for 192.168.201.86 written to 2020_0323_1453-hwinfo-192.168.201.86.txt
Node 192.168.201.88 hwinfo
hwinfo for 192.168.201.88 written to 2020_0323_1454-hwinfo-192.168.201.88.txt
```

swarmctl -Q healthreport

- swarmctl -Q get the health report in json format from a single node
- use with -x to export to json format
- use with -a also to export all nodes output to json files

```
root@c-csn1:~/tmp>swarmctl -Q healthreport -a -x
Node 192.168.201.84 healthreport
healthreport for 192.168.201.84 written to 2020_0323_1454-healthreport-192.168.201.84.json
Node 192.168.201.85 healthreport
healthreport for 192.168.201.85 written to 2020_0323_1454-healthreport-192.168.201.85.json
Node 192.168.201.86 healthreport
healthreport for 192.168.201.86 written to 2020_0323_1454-healthreport-192.168.201.86.json
Node 192.168.201.88 healthreport
healthreport for 192.168.201.88 written to 2020_0323_1454-healthreport-192.168.201.88.json
```


swarmctl -R, -S Cluster Restart and Shutdown

- `swarmctl -R [chassis]` and `swarmctl -S [chassis]`
- `-R` restarts the whole cluster, `-S` shuts down the whole cluster
- dialog box for “are you sure?”
- requires admin access, so use `-p admin:password` if not using default admin passwords
- can add “chassis”, like “`swarmctl -S chassis -d [chassis-ip]`” to shut down a single chassis
- can add `-n` to operate per chassis against only IP addresses in a `NODES.csv` file located in the script directory: example, if `./NODES.csv` has 2 IP addresses, “`./swarmctl -R -d [chassis-ip] -n`” would only reboot the 2 IPs in `NODES.csv`

```
root@c-sn1:~/tmp>swarmctl -R
Are you sure you want to restart this cluster? Storage will be offline for an extended period until restart has completed. [y/N]: y
API reports restart command success.
```

swarmctl -t Disk related statistics

- `swarmctl -t` gives specific details on each disk on a node
- use `-a -x` to get more disk statistics for all nodes in the cluster
- great for seeing changes over time and tracking potentially bad disks- cron job

```
root@c-csn1:~/tmp>swarmctl -t
```

nodeIPAddress	name	uuid	trappedBytes	trapRatio	usedSpace	availSpace	maxSpace	streamCount	status	largestStreamUuid	largestStreamSize	journalBid
192.168.201.85	/dev/sda	38107195961f9bac73d88062ace250f5	938	0.865	3.15GB	6029	10.12GB	1,038	ok	b191f88b765f52ee04293d54402c1101	1048	0
192.168.201.85	/dev/sdb	c3470745206b23d505c459f1c99a67f6	1537	0.805	2.22GB	6362	10.12GB	21,058	ok		102	1
192.168.201.85	/dev/sdc	aa995b97265dd862386bef9d5dd8d005	1564	0.804	2.16GB	6399	10.12GB	6,935	ok		102	0
192.168.201.85	/dev/sdd	a18e8592836fd39d1da925de6a4f1e4c	1513	0.805	2.38GB	6230	10.12GB	15,348	ok		76	1

swarmctl -u Unretire currently retiring drives

- `swarmctl -u` stops retiring drives that are currently retiring
- does not resurrect drives that have already retired
- use with `-a` to stop retiring on all nodes in the cluster

```
root@c-csn1:~/tmp>swarmctl -u -d 192.168.201.86
Attempting to cancel retire of node 192.168.201.86
Management API reports success in canceling retire of node 192.168.201.86
```



03/23/20 21:09:05 GMT Licensed to: Caringo, Inc.

Cluster Health Report | Feeds | Print Shutdown node Restart node

Node IP	Status	Errors	Streams	Used	Trapped	Available	Capacity	Licensed	Uptime	Version	Actions
192.168.201.86	Ok	0	10723	4.197 GB	708.0 MB	35.57 GB	40.48 GB	20.00 TB	44 mins, 37 secs	11.0.3	Retire Node
Vol 0: /dev/sda ID: 5604f98928da2ba6b2dbb5436414699c	Ok	0	3794	0 bytes	400.0 MB	9.719 GB	10.12 GB				Retire Identify Fail
Vol 1: /dev/sdb ID: fcf18e12a231a1375a92f7bf2a253bb	Ok	0	3439	506.0 MB	133.0 MB	9.480 GB	10.12 GB				Retire Identify Fail
Vol 2: /dev/sdc ID: c12163b211cceaf915ed5954edd391ab	Ok	0	3458	546.0 MB	118.0 MB	9.455 GB	10.12 GB				Retire Identify Fail
Vol 3: /dev/sdd ID: 4fe3a936e3289edf7ae698de4abe9dea	Ok	0	32	3.145 GB	57.00 MB	6.917 GB	10.12 GB				Retire Identify Fail

Node Details

Announcements (Last 10)

Clear Announcements...

- Mar 23, 2020 21:08:20 Canceling retire of node bc4c1f4dbf1a8805 (via Management API)
- Mar 23, 2020 21:08:12 Retire without recovery requested by administrator for volume /dev/sda

swarmctl -U User management

- `swarmctl -U` shows you, and allows you to change, the list of admin users
- run without options shows you the list of admin users
- requires `-p [user:password]` if you are not using default admin credentials
- use `-V [password]` to add the password to a new user, specified like: `swarmctl -U [new admin user] -V [password for new admin user]`
- change password for existing user like: `swarmctl -U [current admin username] -V [new password for admin user]`
- be advised when changing the admin password that other systems might have the admin password set and could potentially break if changed until they are updated

```
root@c-snl:~/tmp>swarmctl -U
API reports ['admin'] as admin users on c-csn1.enfield.com
root@c-snl:~/tmp>swarmctl -U bob -V password
bob:password
API reports success=True setting/changing password for bob
root@c-snl:~/tmp>swarmctl -U
API reports ['admin', 'bob'] as admin users on c-csn1.enfield.com
root@c-snl:~/tmp>swarmctl -U bob -V newpassword
bob:newpassword
API reports success=True setting/changing password for bob
```

swarmctl -V Variables

- `swarmctl -V [option]` adds a variable option
- is not used standalone- this option is used to add information to other parameters
- examples include `-C`, `-D`, `-U`, `-z`

swarmctl -w Recovery Reports

- `swarmctl -w` gets the recovery reports for disks that have been retired or are retiring
- useful with `-x -a` to export all recovery reports to file for further analysis

```
root@c-csn1:~/tmp>swarmctl -w -a -x
recovery output for c-csn1.enfield.com is in 2020_0323_1544-recoveries.csv in this directory
root@c-csn1:~/tmp>head 2020_0323_1544-recoveries.csv
node,localVolumeID,localVolumeName,recoveryType,remoteVolumeID,remoteVolumeIP,remoteVolumeName,state,timeEnded,timeStarted
192.168.201.84,78765d2b5d28fb613d55e688eaf878bf,/dev/sdd,ECR,63319b6940a91895ff3ff748dc67fa71,unknown,unknown,completed,2020-03-23T19:38:59.770148Z,2020-03-20T20:24:04.873824Z
192.168.201.84,78765d2b5d28fb613d55e688eaf878bf,/dev/sdd,FVR,63319b6940a91895ff3ff748dc67fa71,unknown,unknown,completed,2020-03-23T19:38:59.770021Z,2020-03-20T20:24:04.875415Z
192.168.201.84,8886bbaf6685815fbf0e92d2f236dc62,/dev/sda,ECR,63319b6940a91895ff3ff748dc67fa71,unknown,unknown,completed,2020-03-23T19:54:04.193961Z,2020-03-20T20:24:04.910437Z
192.168.201.84,8886bbaf6685815fbf0e92d2f236dc62,/dev/sda,FVR,63319b6940a91895ff3ff748dc67fa71,unknown,unknown,completed,2020-03-23T19:54:04.193776Z,2020-03-20T20:24:04.912178Z
192.168.201.84,78765d2b5d28fb613d55e688eaf878bf,/dev/sdd,ECR,bea15642b51bfc1e9540f0e5ab5ed6be,unknown,unknown,completed,2020-03-23T19:38:59.770204Z,2020-03-20T20:35:53.697240Z
192.168.201.84,78765d2b5d28fb613d55e688eaf878bf,/dev/sdd,FVR,bea15642b51bfc1e9540f0e5ab5ed6be,unknown,unknown,completed,2020-03-23T19:38:59.770092Z,2020-03-20T20:35:53.698169Z
192.168.201.84,8886bbaf6685815fbf0e92d2f236dc62,/dev/sda,ECR,bea15642b51bfc1e9540f0e5ab5ed6be,unknown,unknown,completed,2020-03-23T19:54:04.194020Z,2020-03-20T20:35:53.758803Z
192.168.201.84,8886bbaf6685815fbf0e92d2f236dc62,/dev/sda,FVR,bea15642b51bfc1e9540f0e5ab5ed6be,unknown,unknown,completed,2020-03-23T19:54:04.193901Z,2020-03-20T20:35:53.759541Z
192.168.201.84,78765d2b5d28fb613d55e688eaf878bf,/dev/sdd,ECR,9af0fdab6bb9a7525a34c100f2816301,unknown,unknown,completed,2020-03-23T19:38:59.770239Z,2020-03-20T20:44:53.509530Z
```

swarmctl -x export to csv format

- `swarmctl -x` [other option] allows you to export the output to csv (or json in some cases) format for further analysis
- useful with `-a` to export all nodes' information to file for further analysis
- works with multiple other options (it is not a standalone option)
- if used with `-a`, a zipped bundle will be made of the output and you will be prompted with the option to delete the individual files.
- some flags [`-Q`, `-m`, `-z`] allow multiple options for a single node, in which case the output may be bundled in a zip

```
[root@c-csn1 tmp]# swarmctl -m commstats hpstats -x
commstats output for c-csn1.enfield.com is in 2020_0407_0930-commstats.csv in this directory
hpstats output for c-csn1.enfield.com is in 2020_0407_0930-hpstats.csv in this directory
Compressed files left in 2020_0407_0930-c-csn1.enfield.com-commstats+hpstats.zip.
These files have been included in the zip file and can be deleted:
['2020_0407_0930-commstats.csv', '2020_0407_0930-hpstats.csv']
Delete these files? [y/N]: n
[root@c-csn1 tmp]#
```

swarmctl -X Don't persist sessions

- swarmctl -X
- if you have hundreds of nodes, running cluster wide operations might cause your client to run out of file descriptors
- run this to prevent session persistence while running other swarmctl options
- no need to use except if requested by Caringo Support

swarmctl -z Component Log Level

- `swarmctl -z [component] -V [level]`
- can run with multiple components in the same call: `swarmctl -z component1 component2 [-V [level]]`
- this option allows you to change the log level of a particular component instead of changing the log level of every component
- useful when troubleshooting very specific issues, especially in large environments where debug level produces too much data to sift through
- run with no options to see the default levels
- use like “-z -V 0” to reset all components to their default log level
- use with -a to affect all nodes

```
root@c-snl:~/tmp>swarmctl -z
```

Node	Component	Log Level
192.168.201.85	ADMIN	20
192.168.201.85	ASYNC FILE	20
192.168.201.85	BUFFERS	20
192.168.201.85	CACHE	20
192.168.201.85	COLLECTION	20
192.168.201.85	CONFIG	20

swarmctl -z Component Log Level contd

```
root@c-sn1:~/tmp>swarmctl -z LICENSE
+-----+
| Node      | Component | Log Level |
+-----+
| 192.168.201.85 | LICENSE | 20 |
+-----+
root@c-sn1:~/tmp>swarmctl -z LICENSE -V 10
+-----+
| Node      | Component | Previous Log Level | New Log Level |
+-----+
| 192.168.201.85 | LICENSE | 20 | 10 |
+-----+
root@c-sn1:~/tmp>swarmctl -z -V 0
+-----+
| Node      | Component | Previous Log Level | New Log Level |
+-----+
| 192.168.201.85 | ADMIN | 20 | 20 |
+-----+
| 192.168.201.85 | ASYNC FILE | 20 | 20 |
+-----+
| 192.168.201.85 | BUFFERS | 20 | 20 |
+-----+
| 192.168.201.85 | CACHE | 20 | 20 |
+-----+
```

swarmctl --debug debug mgmt api calls

- `swarmctl -[other flags] --debug {[api args http returns]} [api args http returns]`
- this flag is used to show what's going on under the covers during `swarmctl` runs
- the "api" variable shows all of the mgmt api calls (this is default if no variable specified)
- the "args" variable just shows with args were called
- the "http" variable shows all of the http traffic and headers
- the "returns" variable is verbose

```
[root@csn1 tmp]# swarmctl --debug -q
APICommand: logOperations=True
changeAuth: from None:None to admin:caringo
apiGet: "http://192.168.201.84:91/api/storage/clusters" -> 200
apiGet: "http://192.168.201.84:91/api/storage/clusters/c-csn1.enfield.com/summary" -> 200
apiGet: "http://192.168.201.84:91/api/storage/nodes" -> 200
apiGet: "http://192.168.201.84:91/api/storage/nodes/c038e3e20bd1e422" -> 200
apiGet: "http://192.168.201.85:91/api/storage/nodes/b9248f679cadd114" -> 200
apiGet: "http://192.168.201.88:91/api/storage/nodes/zac34b8eb78dad95" -> 200
apiGet: "http://192.168.201.84:91/api/storage/nodes/_self" -> 200
validateUser: "http://192.168.201.84:91/api/validateUser" user: admin
```

Type	name	status	ava	usedSpace	maxSpace	streamCount	swVer	errC	volE
			l1P					ount	rrs
			erc						
			ent						
Cluster	c-csn1.enfiel d.com	ok	63%	28.88GB	121.43GB	120,033	11.0.3	0	0
Subcluster	default	ok	63%	28.88GB	121.43GB	120,033	11.0.3	0	0
Chassis	192.168.201.8 5	ok	75%	10.11GB	40.48GB	38,217	11.0.3	0	0
Chassis	192.168.201.8 4	ok	76%	9.40GB	40.48GB	39,238	11.0.3	0	0
Chassis	192.168.201.8 8	ok	76%	9.37GB	40.48GB	42,578	11.0.3	0	0

swarmctl --feeds Feeds tables

- `swarmctl --feeds` shows the feeds tables for indexer and replication feeds
- this is the same information as seen in the feeds definition page
- does not show you feed statistics

```
root@c-csn1:~/tmp>swarmctl --feeds -x
No replicationfeeds defined on c-csn1.enfield.com
1 searchfeed defined on c-csn1.enfield.com
searchfeed: IndexerFeed-5.6.12-to-c-csn1-indexer2-3
{ 'destination': { 'fullMetadata': True,
                  'host': '192.168.201.203',
                  'indexAlias': 'c-csn1.enfield.com0',
                  'insertBatchSize': 100,
                  'insertBatchTimeout': 1,
                  'port': 9200},

  'id': 0,
  'isDefault': True,
  'lastchanged': '2020-03-10T16:03:57.000+00:00',
  'latency': 30.0,
  'name': 'IndexerFeed-5.6.12-to-c-csn1-indexer2-3',
  'nodeletes': False,
  'noversioned': False,
  'paused': False,
  'type': 'Search'}
No s3backupfeeds defined on c-csn1.enfield.com
```

swarmctl --feed-control and --node-feed-restart

- --feed-control [action] allows you to perform these operations: pause, resume, restart, setdefault
- --feed-type [type] specifies one of three types of feeds: searchfeeds, replicationfeeds, s3backupfeeds
- --feed-id [value] specifies the particular feed
- --node-feed-restart - restarts the feed ONLY for a particular node
- --feed-type and --feed-id are both required for both --feed-control and --node-feed-restart

```
[root@c-csn1 tmp]# swarmctl --feed-type searchfeeds --feed-id 1 --feed-control pause
Attempting to pause searchfeed #1
Management API reports success with pause of searchfeed #1
[root@c-csn1 tmp]# swarmctl --feed-type searchfeeds --feed-id 1 --feed-control resume
Attempting to resume searchfeed #1
Management API reports success with resume of searchfeed #1
```

swarmctl --license License information

- `swarmctl --license` shows the currently deployed license
- this is the same information as seen on the license page or in the license itself

```
root@c-snl1:~/tmp>swarmctl --license
{ 'clusterDescription': 'Ace Lab Cluster',
  'cn': 'Caringo, Inc.',
  'co': 'USA',
  'expirationDate': None,
  'featureClusterMaxObjects': 0,
  'featureClusterMaxTB': 20.0,
  'featureContentIndexing': True,
  'featureErasureCoding': True,
  'featureHardwareCheck': False,
  'featureHealthReportRequired': False,
  'featureKeys': [],
  'featureMinimumMinReps': 1,
  'featurePlatformId': '',
  'featureVolumeLifetime': 'unlimited',
  'l': 'Austin',
  'licenseFormat': '1.1',
```

swarmctl --policies Policies

- `swarmctl --policies` shows you the policies as currently evaluated by the cluster
- shows the policies for Replicas, ECEncoding, ECMinStream, and SizeVersioning

```
root@c-csn1:~/tmp>swarmctl --policies -V Replicas
4 policies found on c-csn1.enfield.com
Policy: Versioning
{
  'evaluatedValue': 'disabled',
  'headerName': 'Policy-Versioning',
  'id': 1,
  'name': 'Versioning',
  'settingDefValue': 'disallowed',
  'settingMibName': 'policyVersioning',
  'settingName': 'policy.versioning',
  'settingValue': 'disallowed',
  'validValues': ['disallowed', 'suspended', 'allowed']}
Policy: ECMinStreamSize
{
  'evaluatedValue': '1000000',
  'headerName': 'Policy-ECMinStreamSize',
  'id': 2,
  'name': 'ECMinStreamSize',
  'settingDefValue': '1Mb anchored',
  'settingMibName': 'policyECMinStreamSize',
  'settingName': 'policy.eCMinStreamSize',
  'settingValue': '1Mb anchored',
  'validValues': ['20Mb', '1Gb', '20Mb anchored']}
Policy: ECEncoding
{
  'evaluatedValue': 'unspecified',
  'headerName': 'Policy-ECEncoding',
  'id': 3,
  'name': 'ECEncoding',
  'settingDefValue': 'unspecified anchored',
  'settingMibName': 'policyECEncoding',
  'settingName': 'policy.eCEncoding',
  'settingValue': 'unspecified anchored',
  'validValues': ['unspecified', '5:2', 'disabled', '5:2 anchored']}
Policy: Replicas
{
  'evaluatedValue': 'min:2 max:16 default:2',
  'headerName': 'Policy-Replicas',
  'id': 4,
  'name': 'Replicas',
  'settingDefValue': 'min:2 max:16 default:2 anchored',
  'settingMibName': 'policyReplicas',
  'settingName': 'policy.replicas',
  'settingValue': 'min:2 max:16 default:2 anchored',
  'validValues': [
    'min:1 max:15 default:2',
    'min:2 max:16 default:3',
    'min:2 max:10 default:2 anchored']}]
```

swarmrestart

- `swarmrestart [options]`
- `swarmrestart` is a binary script replacement for the `-G` option in `snmp-castor-tool.sh`
- used to rolling restart a cluster
- common options include
 - `-p "user:password"`
 - `-n` - reboots only those nodes in the local `NODES.csv` file (like `-n` in `snmp-castor-tool.sh`)
 - `-m [minutes]` - number of minutes to wait for a booted node to mount the disk. 45 minutes by default
 - `-d [ip address]` - any IP in the cluster from which the script will read all storage IPs
 - `-v [version]` - specify the version of storage nodes to restart. Default will restart any. Useful if you have already upgraded some nodes and want only to rolling reboot the nodes that haven't been upgraded yet.
 - `-x [filename]` - a list of IPs that should NOT be rebooted
 - `-w [boot wait]` - wait this long after rebooting a node until trying to contact it. Faster booting nodes, you can set this lower than the 5 minutes default.

swarmrestart example

```
root@c-csn1:~/tmp>swarmrestart -d 192.168.201.85 -w 2 -p admin:caringo -n -u 0
Preparing for rolling restart of cluster: c-csn1.enfield.com
Collecting cluster summary via 192.168.201.85
Ignoring server at 192.168.201.86 with status retired
3 Servers ready for restart:
192.168.201.84[11.0.3] 4 vols, VMware, Inc. VMware Virtual sn:VMware-56 4d c8 .. da 63, up 1 hour 27 minutes/ok
192.168.201.85[11.0.3] 4 vols, VMware, Inc. VMware Virtual sn:VMware-56 4d 5a .. 22 d5, up 1 hour 27 minutes/ok
192.168.201.88[11.0.3] 4 vols, VMware, Inc. VMware Virtual sn:VMware-56 4d 84 .. 56 4c, up 1 hour 27 minutes/ok

Are you sure you want to restart 3 servers in cluster c-csn1.enfield.com? [y/N]: y
Attempting restart of 192.168.201.84... restart command reports success.
192.168.201.84: waiting█
```

Wrap-Up

There are plenty of options. Spend time in the lab before you need them to know how they work and what might be useful for your environment. Please contact support with any issues or requests.



Questions?

info@datacore.com

www.datacore.com