DATACORE™

Swarm Support Scripts Bundle

DATACORE CONFIDENTIAL

The Authority on Software-Defined Storage

Swarm Support Scripts Bundle installation

- If you already have the support scripts bundle in /root/dist, to update the bundle, you simply type: /root/dist/updateBundle.sh
- If you do NOT already have the bundle in /root/dist:
  - You can download from https://support.cloud.caringo.com/tools/caringo-support-tools.tgz
  - Or if your server has internet access, you can install on a CSN, ES node, or Content Gateway by logging in as root and performing:
    - cd
    - curl -O https://support.cloud.caringo.com/tools/updateBundle.sh
    - bash updateBundle.sh fresh
    - rm updateBundle.sh
- Resulting scripts are placed in /root/dist
- Then, anytime you want to update your scripts (and have internet access), simply type:
  - /root/dist/updateBundle.sh
- You can install an alias / function extension in .bashrc and you may see a notification about that when running updateBundle.sh

Swarm Support Scripts Bundle Notes

- Many scripts require a storage node's IP address to run. Define SCSP_HOST in your environment and many scripts will pick up the IP address without specifying it:
  - export SCSP_HOST=[node ip address]
  - put in ~/.bashrc to persist over sessions
- Easily add the bashrcforcustomers file (for handy aliases and functions) to .bashrc using: /root/dist/add-bashrcforcustomers.sh - this is done by the updateBundle.sh script as well.
  - Type: "alias" to see what all shortcuts are now enabled or just open /root/dist/bashrcforcustomers and see what functions and aliases are now present and may be useful for you
- Many scripts can use a file you create called NODES.csv in the current directory to run the script against only IP addresses listed, one per line, in that file. Most scripts that use NODES.csv enable this functionality using the -n flag. This is good if you only want to affect a subset of IPs in a cluster.
- README.txt shows what all scripts are included and give a brief summary
- REVISION.txt is the date stamp from when the bundle was created and thus uploaded to the support site. If you want to see if the bundle has changed since you last downloaded it, you can compare the local REVISION.txt file with the date in https://support.cloud.caringo.com/tools/REVISION.txt

Swarm Support Scripts Bundle aliases and functions

- To install many handy functions and aliases in your shell, do any of these 3 things:
  - run: /root/dist/updateBundle.sh
  - manually run: /root/dist/add-bashrcforcustomers.sh if you aren't able to run updateBundle.sh because of lack of internet access on this server
  - put: "source /root/dist/bashrcforcustomers" in your .bashrc or .bash_profile
- If you do not want to install these permanently, you can enable them in your current session by typing: "source /root/dist/bashrcforcustomers" in your current shell.
- Some handy aliases:
  - vic[then press tab] on a CSN will vi the cluster.cfg file so you don't have to remember the location
  - vin[then press tab] on a CSN, like above but for the netboot.cfg file
  - lots of Content Gateway aliases for stopping/ starting the service, and also for editing the config files… type: gw[tab] to see the options. Example:
    - root@c-csn1:~/Support>gw

```
gwconfig   gwconfigvi gwidsysvi  gwpolicyvi  gwrestart  gwstart    gwstatus   gwstop
```

Swarm Support Scripts Bundle aliases and functions (cont.)

- alias hg='history | grep -i'   # type hg to grep through your history
- alias eh='echo $SCSP_HOST' #to show your currently set SCSP_HOST
- There is a function called “ch” that changes your current shell’s SCSP_HOST. Type: ch [new storage node ip]. You can then type: “eh” to confirm the change.
- type: “logs” to quickly cd to /var/log/caringo on any system where that exists
- alias xtar='tar zxvf' # to quickly unzip a .tgz extension tarball, type: “xtar blah.tgz”
- type: “uploadtosupport [filename]” on a system connected to the internet to quickly send a file to Swarm support. It will ask for customer name and ticket ID (like: SUP-1234)
- type: nocomment [filename] to quickly output a file removing all of the comment lines that start with the # symbol. This is good if you have a default configuration filled with commented out defaults and only want to see what’s changed.
- There are several functions that do something to the last file (based on time changed) in a directory: vilast, wclast, catlast, lesslast, taillast. These do what you would think based on the prepending characters before the word “last”. For example, type: “vilast” to start editing the file in the current directory that was most recently changed.

Swarm Support Script examples

- swarmctl (mostly replaces snmp-castor-tool.sh)
- swarmrestart (replaces snmp-castor-tool.sh -G)
- snmp-castor-tool.sh
- indexer-enumerator.sh
- test-volumes-network.sh
- collect_health_reports.sh (replaced by swarmctl -Q healthreport)
- platform-read-pss.sh (replaced by swarmctl -P)
- hwinfo-dmesg-grab.sh (replaced by swarmctl -Q hwinfo and swarmctl -Q dmesg)
- techsupport-bundle-grab.sh
- settings_checker.py

swarmctl

- snmp-castor-tool.sh replacement for Swarm 10+
- used to take actions on the nodes like changing settings, clearing errors, reboot cluster, remount stale volumes, change log level
- used to gather information like disk statistics, IOs, dmesg
- by default, only shows information from a single node. Use -a to get info from all nodes
  - Example: swarmctl -a -C policy.ecMinStreamSize -d [node ip]
- -d [node ip] is not necessary if you set SCSP_HOST in your shell
- assumes “caringo” or “ourpwdofchoicehere” as the admin password for the cluster, so use -p [user:password] on calls if you have a different password
- There is a whole separate training deck covering each option for this script

swarmctl (cont.) - common options

- -a to send request to each node (by default, action affects a single node)
- -d [node] to specify a node
- -C [config parm] to see a current value. Use with -V to change the value
- -E clear - to clear errors on a node. Use with -a for all nodes' errors
- -i [log level] - change the log level for the whole cluster
- -L [log level] - to change the log level for a single node
- -P to get the PSS, use with -x to send to a file
- -Q [{dmesg,hwinfo,healthreport}] - to get dmesg, hwinfo, or generate a health report locally
- -R, -S - to restart and shutdown the cluster respectively
- -x used with other options to output the results to a file
- download the complete swarmctl training here:

  https://support.cloud.caringo.com/tools/Tech-Support-Scripts-Bundle-swarmctl.pdf

swarmrestart

- binary that works on CentOS
- Is not currently supported against Swarm 9 and earlier (use snmp-castor-tool.sh -G for Swarm 9 and previous)
- Primary purpose- rolling restart of the cluster using the management API
- Able to include or disclude particular IPs
- Able to only reboot nodes on certain versions
- Ability to resume an interrupted session
- Run normally like: swarmrestart -d [node ip]
- Keeps a log of what’s been done and at what time

snmp-castor-tool.sh

- There are plenty of options- "aAbcC:d:D:eEfFgGhi:Ij:klL:mMno:Op:qrRsStuvV:wxy"
- What options do I possibly need?
- You could use the help (-h)
- You're likely being told to run it by Swarm Support for a particular reason in which case they will give you the option(s)
- Other than that, there are some common uses which will be presented after the overview
- Please make sure you use this script carefully. Remember, only uppercase switches make changes
- Practice in a lab so you are familiar with it
- This script is your friend, but try swarmctl instead, especially with Swarm 10+

snmp-castor-tool.sh overview

- Help! (-h)
- Lowercase options don't make changes to cluster- they are read only
- Uppercase options make changes to the cluster and always requires the read/write password specified
- read/write password is pulled from the CSN's cluster.cfg file so not necessary to enter manually
- avahi vs CSN (ip-assignments --ips) vs NODES.csv
  - by default, snmp-castor-tool.sh will collect IPs to run commands against via the ip-assignments script
  - ip-assignments is not populated if your addresses are configured statically in node config files
  - avahi-browse is available to get a list of all nodes (-d [cluster name] option)
  - use the -n option to specify operations ONLY against IPs specified in NODES.csv
  - using -n is VERY handy if you are only interested in results from a subset of IPs (see createNODEScsv.sh script in the bundle)
- NODES.csv- including its use with the -n option- is common across many support scripts
- MIBS should already be loaded in /usr/share/snmp/mibs- might need updating
- platform-update-mibs.sh (a script in the support bundle) will update the MIBs on a CSN with the latest MIBs
- The -x option is available in some (not all) listing options to output to an Excel friendly output. It is silent for other options that aren't applicable.
- The -x option may output to semicolon separated output if the actual values returned via SNMP include commas.

indexer-enumerator.sh

- Easily enumerate clusters
- All stream types are returned by default
- The benefits of the script are:
  - Easy to remember syntax
  - Enumerate all domains in a single run
  - Output is in easy-to-use (non-json) format which can then be used by other scripts
- Can set SCSP_HOST variable in environment to not include the -a [NODE IP]

indexer-enumerator.sh example options

- -h help
- -o send output to stdout vs (default) file
- -D output only a listing of domain names
- -b [bucket] output only streams in [bucket]
- -B output only bucket names
- -c get a count of matching streams only (don’t enumerate)- can be used with other options
- -d ALL enumerate all domains found
- -d [domain] enumerate a particular domain
- -eEfFgG [#] enumerate streams written prior or since # minutes/ days/ years
- -l and -L [bytes] return streams smaller (l) or larger (L) than the provided value
- -m [value] look for a specific metadata header (use -v to pass the value of the header)
- -p [prefix] only return those streams that match a particular prefix
- -S use HTTPS if fronted by SSL (-U [user] -P [pass])
- -t only enumerate un-tenanted streams
- -v [value] see -m option

For an in-depth walk-through, please see here.

indexer-enumerator.sh examples

- indexer-enumerator.sh -h
- indexer-enumerator.sh -a $SCSP_HOST -D
- indexer-enumerator.sh -a $SCSP_HOST -D -o
- indexer-enumerator.sh -a $SCSP_HOST -d ALL -o *#notice that gatewayadmindomain is the only domain with many streams by default*
- indexer-enumerator.sh -a $SCSP_HOST -d gatewayadmindomain -B -o *#only show buckets from that domain*
- indexer-enumerator.sh -a $SCSP_HOST -d gatewayadmindomain -U -o *#only show unnamed streams*
- indexer-enumerator.sh -a $SCSP_HOST -d gatewayadmindomain -N -o *#only show named streams*
- indexer-enumerator.sh -a $SCSP_HOST -d gatewayadmindomain -N -u -o *#only show named streams with disk usage of single replica reported*
- indexer-enumerator.sh -a $SCSP_HOST -d gatewayadmindomain -b _admin -o *#only show streams from bucket "_admin"*
- indexer-enumerator.sh -a $SCSP_HOST -d gatewayadmindomain -b _admin -P config -o *#only show streams from bucket "_admin" that start with "config"*

test-volumes-network.sh

- This script will test volumes and/or network performance using Swarm's internal test mechanism.
- Can be run with -f [path to NODES.csv] to run against a subset of nodes.
- Typically run at Swarm Support’s request
- The -n option can be replaced by: swarmctl -m networktest -a -x (to run against all nodes and output to file)

collect_health_reports.sh

- This script will collect health reports manually
- This is typically used for sites that have no connectivity to the outside world.
- The results are a tarball that should be sent to Swarm Support
- Replaced by: swarmctl -Q healthreport -x -a (to get all json files). swarmctl is MUCH faster especially in larger environments.

platform-read-pss.sh

- Run this script on a CSN or Swarm Platform Server to get the Persistent Settings Stream for all nodes or a subset of nodes
- This is typically run when there is a password mis-match or you want to back up your PSS
- Good to run if you want to see the parameters in effect currently in your cluster (via CLI). cluster.cfg is potentially overwritten by the PSS so the PSS will have the parameters in effect.
- techsupport-bundle-grab.sh collect this as well.
- Replaced by: swarmctl -P -x # export to file, and may need -p [user:pass]

hwinfo-dmesg-grab.sh

- Run to grab and tarball the hwinfo and dmesg output from each node in the cluster
- Typically run at Swarm Support's request
- Can take a while depending on the size of the cluster
- Can run with -n to force check for NODES.csv to only run against a subset of nodes.
- Replacements: hwinfo and dmesg options can be output using swarmctl: swarmctl -Q dmesg -x -a, or swarmctl -Q hwinfo -x -a

techsupport-bundle-grab.sh

- Run this on a CSN, standalone SCSP Proxy, Indexer (Elasticsearch node), or Content Gateway
- Provides a single step vehicle to collect the most often needed support information
- Use the -s option to run the Swarm 10+ Settings checker to verify your Swarm settings are ready for Swarm 10+. This is only available on the CSN platform.
- Use the -l option if you do not need to collect logs (but only need to collect environmental information about the CSN itself).
- Use the -e option to not grab the entirety of /etc/*
- Use the -H option to not grab hwinfo/ dmesg information from the storage nodes
- Use the -i option to not collect Swarm Search feed (indexer/ Elasticsearch) data
- Use the -p option to manually collect health report data (useful for real time reports or if your cluster is not connected to the Internet)
- Use the -n [integer] to collect integer number of logs. By default, the script grabs the 2 previously rotated logs.
- If the machine it is run on has access to the internet, you may run:
  - uploadtosupport [filename of resulting output]
- That will upload the file to us. You will be asked to put in the case number and the tarball will be attached to the appropriate ticket.
- We will be notified of this new file automatically as a ticket update.

settings_checker.py

- Settings checker on a CSN is called by using the -s option with techsupport-bundle-grab.sh in most cases.
- If no CSN is available, but you have a host with python3, you can run this script from that host.
- Processes config files and the PSS (Persistent Settings Stream) to determine if any changes are recommended before upgrading.
- Having this output analyzed is a requirement for upgrading from 9.x to 10+ and beyond.
- Example: python3 /root/dist/settings_checker.py -a $SCSP_HOST -w caringo -p ./ #where your configuration files are in the local directory (the -p option) and "caringo" is the admin password.

DATACORE™

Questions?

info@datacore.com

www.datacore.com