Pure Storage Array Enhanced Snapshot Facility

Russell Peters – rupeters@coca-cola.com

12/3/2015

Table of Contents

Document Information and Revision History

Document Name	Pure Storage Array Enhanced Snapshot Facility
Original Author(s)	Russell Peters
Current Revision	1.2

Version	Date	Author(s)	Revision Notes
0.9	05/07/15	Russell Peters	Initial Document Creation
0.91	5/24/2015	Russell Peters	Added options to snapsched.ini for replication and expiration
1.0	6/1/2015	Russell Peters	Added -C and -R options to snapcopy. Published document.
1.1	6/13/2015	Russell Peters	Added Enhancement Summary section
1.2	12/3/2015	Russell Peters	Added options to snapsched.ini: noexpirelist and eradicate options.

Executive Summary

The Pure Storage Array Enhanced Snapshot Facility is a customized suite of three basic custom Python based tools (summarized below) enhancing the Protection capability that ships with the Pure Storage array. The facility exploits the REST interface and utilizes the Python API provided by Pure Storage and is currently only tested to run on Unix/Linux platforms (but should work similarly on Windows). Therefore, Python, and the Pure Storage Python API is a prerequisite.

- snapsched
  This is the heart of the facility. The snapsched daemon (a background process) along with the associated snapsched.ini configuration file, operate with the Pure Storage Array Protection Group Snapshot architecture to schedule, replicate, and expire Protection Group snapshots, which are called by this facility as Snapshot Groups. The tool introduces the concept of a Snapshot Group snapshot "level" property providing for better fine-tuning the granularity of snapshot expiration as snapshots age. Each Snapshot Group can contain up to three (3) snapshot levels. Each snapshot level provides its own snapshot interval schedule and local expiration schedule. Optionally each level can be configured to maintain its own customized snapshot command (allowing for pre and post directives for the scheduled snapshot). Additionally snapsched provides an enhanced method to maintain a list of hosts AND volumes which are protected (or excluded) from snapshots. For instance a full host (or set of hosts) can be included in the group along with a volume exclusion string which indicates certain volumes (such as those containing data not requiring protection) that will not be "snapshotted". If a volume is added to the host whose name doesn't match the exclusion string, then it will automatically be added to the Snapshot Group's associated Protection Group before the next scheduled snapshot.

- snaplist
  This tool lists the summarized Snapshot Groups and their properties such as creation date, age, retention schedule, and time-to-live. Any snapshots taken outside the snapsched tool, e.g., a manual Protection Group snapshot, are also listed by snapshot suffix, but will not show certain properties (such as retention schedule and time-to-live). The tool can also be instructed to list volume details of a particular Snapshot Group. Additionally the tool can be used to quickly destroy and/or eradicate all volumes in a Snapshot Group.

- snapcopy
  This is a powerful tool that, with a single command, can quickly copy a Snapshot Group, or a subset of volume snapshots in the Snapshot Group (based on passed arguments) to new or existing volumes (or to recover the snapshots back to the original source volumes). The tool can also be used, via optional argument, to connect the copied volumes (if desired) to an already defined host. Additionally the tool can be used in audit mode. In this mode the tool doesn't actually perform the copies or connections, but displays output as if it were performing the commands. Thus allowing users to preview the work to test and verify that the arguments passed actually execute the expected work.

Summary of Enhancements

1. Enhances granularity in scheduling snapshots by providing ability to fine tune (to the desired minute) when snapshots occur.
   Built in snapshot scheduler simply controls the interval of the snapshot (which begin at configuration time).

2. Enhances granularity in expiring snapshots (both local and replicated … independently) by providing ability to choose up to three different snapshot retention periods (based on the assigned level category) for both local and replicated copies of snapshots.
   Built in snapshot scheduler only provides two different expirations "levels" for short-term and long-term retention.

3. Enhances ability to control and see when a specific snapshot (both local and replicated) is scheduled to expire (based on the assigned snapshot level category).
   Built in snapshot scheduler provides no ability to know when a specific snapshot is going to expire – except in simple schedules (where snapshots aren't retained for additional days).

4. Provides ability to view 1) creation of snapshot, 2) age of each snapshot, 3) retention period assigned to the snapshot, 4) expiration/time left for each snapshot (mentioned above), and 5) number of volumes that are members of the snapshot.
   Built in snapshot mechanism only allows to view the creation of the snapshot and the number of members in the current snapshot protection group (which may not match the number in a previous snapshot). Age is not viewable and retention and time-left are not ascertainable for long-term snapshots (since there is no mechanism to determine which snapshots will be selected to be retained for long-term).

5. Enhances ability to fine tune to the desired minute when snapshots expire (both local and replicated – independently) for each snapshot level category.
   Built in snapshot scheduler only provides fine tuning to the minute for snapshots on a "short-term" schedule. Long term snapshot expiration and only be fine-tuned to the "day".

6. Enhances replication by providing ability to replicate local snapshots.
   Built in snapshot scheduler will not replicate snapshots made from the local scheduler. Replicated snapshots are made using a different schedule (which makes a local snapshot that replicates and is removed when the next replicated snapshot occurs). This creates more snapshot overhead which increases the risk of reaching the 5000 snapshot limit.

7. Provides ability to call out a user-defined script to perform snapshots – which can be fine-tuned as to which snapshot interval (based on the snapshot level) is chosen to call the user-defined script. This allows an administrator the ability to quiesce applications/databases that are running on the member volumes before snapshot execution, allowing for enhanced/finer tuned point in time recovery.
   Built in snapshot mechanism provides no such functionality.

8. Provides ability to exclude volumes (by name substring) from being included in snapshots.
   Built in snapshot mechanism provides no such functionality.

9. Provides ability to automatically include and add volumes (by name substring) to protection groups for future snapshots.
   Built in snapshot mechanism provides no such functionality.

10. Provides ability to control snapshots from multiple arrays in one configuration file.
    Built in snapshot mechanism can only control snapshots for the source array.

11. Enhances ability to make snapshot copies by providing mechanism with a single command to copy all (or a subset, based on several filter options) snapshot volumes to either new or (bu overwriting) existing volumes. The same command can be run in "audit" mode so as to "practice" or to display the CLI or REST commands that can be used to perform the copies manually. Additionally the command provides ability to connect newly copied volumes to an existing host (or host group).
    Built in snapshot mechanism requires each volume to be copied manually (one at a time) which can be time consuming and error prone.

snapsched

Purpose

Starts snapsched daemon and automatically schedules snapshots.

Syntax

snapsched [**-h**] [**-c** CFG] [**-l** LOGDIR]

Description

The snapsched daemon schedules, replicates, and expires Pure Storage Array Protection Group snapshots as defined and configured in the snapsched configuration file, which, if not specified, is **/etc/snapsched.ini**. The Pure Storage array ships with a default embedded scheduler, however the snapsched facility provides enhanced options that are not available with the default scheduler, such as multiple level scheduled interval and local expiration periods per Protection Group. The snapsched daemon also provides for a snapshot script call-out configuration to allow for pre and post directives before and after calling the snapshot command. Additionally the daemon can be configured to include whole hosts, host groups, and/or volumes and to exclude certain volumes using a string match based on volume name.

Note that as of this writing, the snapsched command will not self daemonize. Therefore run snapsched.start.sh (located in the same folder as snapsched) to daemonize snapsched or add to the system startup script.

Every 60 seconds, the daemon scans the snapsched configuration file for changes and updates the current running configuration. For instance, to disable a schedule for a particular Snapshot Group, simply change the “enable” option from True to False for the appropriate Snapshot Group section. Within 60 seconds the snapshot daemon will update and disable snapshots, replication, and expiration performed by the daemon (note that the Pure Flash Array expiration schedule will still apply however). All other options in the configuration file are dynamic as well.

The snapsched daemon uses the Protection Group concept already defined in the Pure Array. If a Snapshot Group has been defined in the /etc/snapsched.ini file that has no associated Protection Group, then snapsched will create and configure a Protection Group on the array. However if the Snapshot Group requires replication, it is advised to set up the Protection Group, along with the replication configuration, prior to creating a Snapshot Group section in the snapsched.ini file.

The snapsched daemon performs snapshots on a specified interval and assigns its own additional suffix based on 1) the “level” of snapshot the snapshot interval (1, 2, or 3) and 2) an incrementing three digit snapshot number. Each snapshot level comprises its own local retention/expiration schedule. A level 1 snapshot occurs at every interval except when it coincides with level 2 or level 3 snapshot intervals (since level 2 and 3 snapshots override level 1 snapshots). Level 1 snapshots will likely have the shortest local retention schedule (although not a requirement). An optional level 2 snapshot occurs less frequently (the snapshot schedule must be a multiple of the level 1 schedule) and will have a local retention schedule that is likely longer than the level 1 but less than that of the level 3. An optional level 3 snapshot occurs the least frequent (must occur as a multiple of a level 2 snapshot) and will likely have the longest local retention schedule.

See the snapsched.ini file documentation for more information.

Options

```
-h, --help
                        show this help message and exit

-c CFG, --cfg CFG
                        Config File Name (default = /etc/snapsched.ini)

-l LOGDIR, --logdir LOGDIR
                        Log Directory (default = /var/snapsched)
```

Files

`/etc/snapsched.ini`

The snapsched configuration file (if not specified by the -c flag).

`/etc/pure.ini`

The Flash Array configuration file. Contains each array name/address and api_key needed for access to the REST interface.

/etc/pure.ini

Purpose

Contains each array name/address and API token needed for access to the REST interface.

Description

The /etc/pure.ini configuration file uses a structure similar to what you would find on Microsoft Windows INI files where sections are defined within square brackets, e.g., [section] and options within a defined are key/value pairs, e.g., option = value (or option: value). For each Pure Flash Array accessed by the tool suite, a corresponding section is required whose name matches the DNS resolvable name of the array. That section will contain one key/value pair. The key = "api-key", and the value is the API token of a valid array administrator ID. To find the API token for an array ***USER*** run the following at the array CLI:

`pureadmin list --api-token` ***USER*** `--expose`

Additionally, the file can consist of a section named [default] containing one key, "array", whose value is the default array used if no array is specified in a utility command.

Examples

A typical /etc/pure.ini file will look like this:

```
[default]
Array: abcpure01

[abcpure01]
api_key: 7636f6ae-e765-897c-638c-ebc6ea8ec48e

[abcpure02]
api_key: 432be9cc-0d29-c03e-3c53-7c07e19c0e8e
```

snapsched.ini

Purpose

Defines the snapshot configuration used by the snapsched daemon.

Description

The snapsched configuration file uses a structure similar to what you would find on Microsoft Windows INI files where sections are defined within square brackets, e.g., [section] and options within a defined are key/value pairs, e.g., option = value. With the exception of the [default] section, each section defined in the snapsched configuration file represents (and matches by name) a Protection Group (also called a Snapshot Group by this utility's documentation) on the associated Pure Storage Flash Array. Each section must be unique, i.e., it is invalid to have multiple sections with the same name. Also option keys within a section must be unique.

Option Keys

debug

Optional and only applies to the [default] section. If set to True, then extra debug information is written to the log files in the /var/snapsched directory.

eradicate

Optional and only applies to the [default] section. If set to True, then all expired snapshots are immediately eradicated by the facility after they are destroyed. If not set (or set to False), then all snapshots are only destroyed upon expiration and remain recoverable for a fixed 24 hour period.

array

Optional - Specifies the array containing the corresponding Section/Protection Group name. If no array is specified, then the default array name (as specified in /etc/pure.ini) is used.

enabled

Optional - Must be "True" or "False". Specifies if this Snapshot Group is actively snapshotting, replicating, and expiring. If not specified, then defaults to True.

hosts

Optional - A comma separated list that specifies the host or hosts whose connected volumes will be protected by the snapsched daemon. The puresched daemon updates the associated Protection Group configuration with the list of volumes associated with the hosts specified in this option. The puresched daemon also constantly monitors this list for changes in configuration (such as new volumes added/removed from a host on the list or new hosts added/removed from the list) and updates the volume list on the Flash Array's Protection Group.

replicate

Optional - Must be "True" or "False". Specifies whether snapshots taken by the snapsched daemon will be replicated to target arrays (required to be preconfigured in the Protection Group configuration on the source and target Flash Arrays). If not specified, then defaults to False. Note that, by design, snapsched will designate all snapshots at the level1_interval to replicate unless, at snapshot time, there is an active replication detected of this protection group. If, at snapshot time, there is an active replication occurring, then the current snapshot is still taken locally, but not queued to replicate.

target_array

Required if replicate = True. Specifies the name of target array for replication. It is assumed that the source and target arrays are connected.

level1_interval

Required - Specifies the time interval, in minutes, between each snapshot. Most snapshots at this interval will have a suffix of level 1, unless overridden by the optional less frequently occurring level 2 or level 3 snapshots.

level1_local_retention

Required - Specifies the time, in minutes, that the level 1 snapshots will be available locally before they are expired (destroyed and eradicated).

level1_target_retention

Optional - Specifies the time, in minutes, that the level 1 snapshots will be available at the target before they are expired (destroyed and eradicated). If replicate = True and not specified, then value defaults to level1_local_retention.

level2_interval

Optional - Specifies the time interval, in minutes, between each level 2 snapshot. The value MUST be a multiple of the level1_interval. Snapshots taken at this interval will have a suffix of level 2 (unless overridden by a level 3 interval) and will override level 1 snapshots.

level2_local_retention

Optional - Specifies the time, in minutes, that the level 2 snapshots will be available locally before they are expired (destroyed and eradicated). Likely you want this time to be significantly longer than the level1_local_retention.

level2_target_retention

Optional - Specifies the time, in minutes, that the level 2 snapshots will be available at the target before they are expired (destroyed and eradicated). If replicate = True and not specified, then value defaults to level2_local_retention.

level3_interval

Optional - Specifies the time interval, in minutes, between each level 3 snapshot. The value MUST be a multiple of the level2_interval. Snapshots taken at this interval will have a suffix of level 3 and will override level 2 snapshots.

level3_local_retention

Optional - Specifies the time, in minutes, that the level 3 snapshots will be available locally before they are expired (destroyed and eradicated). Likely you want this time to be significantly longer than the level2_local_retention.

Level3_target_retention

Optional - Specifies the time, in minutes, that the level 3 snapshots will be available at the target before they are expired (destroyed and eradicated). If replicate = True and not specified, then value defaults to level3_local_retention.

snapscript

Optional - Specifies a custom script to be called instead of performing a snapshot on the array. This allows for customizations such as database preparation before and after a snapshot. It is expected that the script will use an API call to request the snapshot itself.

It is expected that the snapscript have these variable strings as arguments in the command. The strings are case sensitive so must match exactly:

"SnapshotGroup": This string will be replaced by the Snapshot Group/section name associated with this option.

"snapname": This string will be replaced by the suffix name (containing the snapshot level and 3 digit number) that would have been assigned had this script not been called.

snapscript_levels

Optional - A comma separated list of valid snapshot levels 1, 2, or 3. This specifies which snapshot intervals (level 1, level 2 , or level 3 intervals) will be used to call the optional snapscript command (if that option was specified). If not specified, then snapscript will not be called (a normal snapshot at the appropriate level will occur). For example if "2,3" is specified, then the snapsched daemon will perform the snapshots for level 1 intervals, but for level 2 and level 3 intervals, the specified snapscript command will be called instead.

starttime

Required - Specifies the time in HH:MM format for the next occurring highest level snapshot interval. For instance if the highest level snapshot interval specified is level1_interval, then the daemon will not take a snapshot until the specified starttime after which snapshot intervals will begin. However, if the highest level snapshot interval specified is level2_interval, then the daemon will not take a level 2 snapshot until the specified starttime (likewise of level3_interval). NOTE however, if lower level snapshot intervals are available BEFORE the starttime, those snapshots will be taken by the daemon. For example if starttime = 22:00 and level2_interval = 720 (12 hours) and level1_interval = 15, then level 1 snapshots will be taken for all 15 minute intervals between the current time and 22:00. At 22:00 a level 2 snapshot will occur. This allows for fine tuning exactly when snapshots occur.

vol_excludestr

Optional -A string that will specify volume names, that contain this string as a substring, will be excluded from the Protection Group volume list. Even if a volume name also matches the vol_includestr, it will be excluded.

vol_includestr

Optional -A string that will specify volume names (even those not connected to specified hosts, or no hosts at all), that contain this string as a substring, will be included from the Protection Group volume list (unless it also matches the vol_excludestr).

noexpirelist

Optional -A string that will specify a comma (or space) separated list of snaphots that the snapshot facility will not expire (will not destroy) even if the snapshot age is past its expiration date. To get a list of snapshot names that can be added to this option, run snaplist. Note that remote snapshots will be in the colon format *remote-array-name*:*snapshot-name*

Examples

- The following defines a snapshot schedule with an associated Protection Group, AIXOS, on array abcpure02. The schedule will take a daily (every 1440 minutes) snapshot at 22:00 of all Pure Storage volumes that include the string “rootvg”. It retains those snapshots for 7 days (10800 minutes) and does not replicate to any target arrays. If a new volume gets created in the future that contains the string “rootvg” it will get added to future snapshots.

```
[AIXOS]
array = abcpure02
enabled = True
starttime = 22:00
vol_includestr = rootvg
replicate = False

level1_interval = 1440
level1_local_retention = 10800
```

- The following section defines a snapshot schedule on array, abcpure02 for Protection Group, snap_group3.
  - It will populate the Protection Group with all abcpure02 volumes visible to hosts xx11 and xt400 except for those that contain the string, rootvg. If new volumes are added or removed from these hosts, they will be removed and added to the Protection Group
  - Level 1 snapshots will occur every 15 minutes (except when level 2 and 3 snapshots occur) and since the starttime is 12:00 the snapshots will occur at minutes 00, 15, 30, and 45 of the hour.
  - Level 1 snapshots will be retained for 4:15, or 255 minutes.
  - Level 2 and level 3 snapshots run the script called create_pgroup_snapshot and will replace the arguments “SnapshotGroup” with the Protection Group Name, snap_group3, and the “snapname” with the snaphost suffix name (which is dynamically decided depending on the level and incremental number).
  - Level 2 snapshots will occur every 4 hours, or 240 minutes (except when level 3 snapshots override) and since the starttime is 12:00, these snapshots occur at hours 0, 4, 8, 12, and 16 of each day.
  - Level 2 snapshots will be retained for 1 day at 4 hours, or 1680 minutes.
  - Level 3 snapshots will occur every 12 hours, or 720 minutes beginning at noon and also at midnight.
  - Level 3 snapshots will be retained for 30 days, or 43200 minutes.
  - No snapshots are replicated

```
[snap_group3]
array = abcpure02
enabled = True
starttime = 12:00
hosts = xx11,xt400
vol_excludestr = rootvg
replicate = False
snapscript = /utc/bin/purestorage/create_pgroup_snapshot SnapshotGroup snap=True
suffix=\"snapname\"
snapscript_levels = 2,3

level1_interval = 15
level1_local_retention = 255

level2_interval = 240
level2_local_retention = 1680

level3_interval = 720
level3_local_retention = 43200
```

- The following section defines a snapshot schedule on array, abcpure02 for Protection Group, P66_PG.
  - It will populate the Protection Group with all abcpure02 volumes visible to hosts P66ha3. If new volumes are added or removed from P66ha3, they will be removed and added to the Protection Group.
  - The Target Array is xyzpure01
  - Level 1 snapshots will occur every 15 minutes (except when level 2 and 3 snapshots occur) and since the starttime is 12:05 the snapshots will occur at minutes 05, 20, 35, and 50 of the hour.
  - Level 1 snapshots will be retained for 4:15, or 255 minutes both locally and at the target.
  - Level 2 and level 3 snapshots run the script called dbfreeze_snap.sh and will replace the arguments "SnapshotGroup" with the Protection Group Name, P66_PG, and the "snapname" with the snaphost suffix name (which is dynamically decided depending on the level and incremental number).
  - Level 2 snapshots will occur every 4 hours, or 240 minutes (except when level 3 snapshots override) and since the starttime is 12:00, these snapshots occur at hours 0, 4, 8, 12, and 16 of each day at 5 minutes after the hour.
  - Level 2 snapshots will be retained for 1 day and 4 hours, or 1680 minutes locally and remotely.
  - Level 3 snapshots will occur every 12 hours, or 720 minutes beginning at 12:05 and also at 0:05
  - Level 3 snapshots will be retained for 30 days, or 43200 minutes locally but 1 day and 4 hours at the target.
  - All snapshots are replicated. It is assumed that the Protection Group has already been configured and enabled at the target array(s).
  - The Protection Group will be configured to automatically expire all replicated snapshots after 60 minutes, however it will retain 1 of those snapshots for two additional days.

```
[P66_PG]
array = abcpure01
target_array = xyzpure01
enabled = True
starttime = 12:05
hosts = P66ha3,P66ha4
vol_excludestr = tempvg
replicate = True
snapscript = /usr/local/bin/dbfreeze_snap.sh SnapshotGroup snapname
snapscript_levels = 2,3

level1_interval = 15
level1_local_retention = 255
level1_target_retention = 255

level2_interval = 240
level2_local_retention = 1680
level2_target_retention = 1680

level3_interval = 720
level3_local_retention = 43200
level3_target_retention = 1680
```

snaplist

Purpose

To list the summarized or detailed information about Groups of Snapshots. Also used to eradicate a Snapshot Group snapshot.

Syntax

```
snaplist [-h] [-id ID] [[-s SNAPSHOT] [-d | -d -e]] [-p PGROUP]
```

Description

This command, when run without specifying a SNAPSHOT, will list a summary of all Snapshot Groups, or suffixes (volumes that have been taken together) – along with their creation date, age, snapshot name, and count of volumes. Additionally for Snapshot Groups matching a section configured in the optional /etc/snapsched.ini file, three more fields are displayed: Retention, Time Left, and Size_MB.

Additionally -p flag can be supplied that will filter Snapshot Groups by a given string. For instance "-p AIX" will filter and display only Snapshot Groups that contain the string "AIX".

If a SNAPSHOT is specified (with the -s argument), then details about the each snapshot volume in the given Snapshot Group will be displayed: Creation Date, Volume Name, Size_GB, Serial, hosts, hostgroup.

Additionally a give Snapshot Group can be manually destroyed (put in recycle bin) or destroyed and eradicated permanently using the -d and/or -e flags.

Options

```
-h, --help
                show this help message and exit

-id ID, --id ID
                Array Name

-s SNAPSHOT, --snapshot SNAPSHOT
                List all snapshot volumes of the given snapshot

-d, --destroy
                Destroy the given snapshot. Requires -s, --snapshot flag

-e, --eradicate
                Eradicate the given snapshot. Requires -d, --destroy flag

-p PGROUP, --pgroup PGROUP
                Optional -- Protection Group/Snapshot Group to show.
```

Files

`/etc/pure.ini`

The Flash Array configuration file. Contains each array name/address and api_key needed for access to the REST interface.

`/etc/pure.ini`

The Flash Array configuration file. Contains each array name/address and api_key needed for access to the REST interface.

Examples

- To list all Snapshot Groups on abcpure02:
  ```
  snaplist –id abcpure02
  ```
- To list all Snapshot Groups on abcpure02 that contain the string “snap” in their name:
  ```
  snaplist –id abcpure02 –p snap
  ```
- To list all volume snapshots of the Snapshot Group AIXOS.L1-0008:
  ```
  snaplist –s AIXOS.L1–0008
  ```
- To destroy and eradicate all of the Snapshot Group AIXOS.L1-0008:
  ```
  snaplist –s AIXOS.L1–0008 –d –e
  ```

snapcopy

Purpose

To easily copy specified (or all) snapshots of volumes in a Snapshot Group to new or existing volumes and to optionally connect these new volumes to an existing host.

Syntax

```
snapcopy [-h] [-id ID] -s SNAPSHOT [-t TARGET] [-a] [-e] [-p PREFIX] [-f] [-o {host,hgroup,volstr}]
         [-ss SOURCE] [-C] [-R]
```

Description

This command will copy specific volume snapshots in a given Snapshot Group (-s flag) to new or existing volumes with similar names or back to existing volumes with the same name. Optionally (with the -t flag), the new volumes will be connected to an existing target host. The tool can be run in audit mode (-a) that allows for previewing the expected work and does not actually perform the copies or connections. A list of Snapshot Groups can be listed using the snaplist command.

The copy of a snapshot volume to an existing volume will fail unless the -f, --force flag is used. This is to ensure that existing volumes will only be overwritten unless explicitly specified to do so.

The -o and -ss flags are used together to filter a subset of the volume snapshots in the given Snapshot Group. For example “-o volstr -ss rootvg” specifies to only copy snapshot volumes whose names contain the rootvg string – or “-o host -ss xx11” specifies to only copy snapshot volumes whose source volumes are currently connected to host xx11.

Options

```
-h, --help
                    show this help message and exit

-id ID, --id ID
                    Array Name

-s SNAPSHOT, --snapshot SNAPSHOT
                    Snapshot Group name that will be copied
-t TARGET, --target TARGET
                    Name of Target Host or Host Group to which the new volumes will be connected

-a, --audit
                    Only perform audit. Do not copy snapshot or connect copied volume to target

-e, --epochtimestamp
                    Append snapshot epoch timestamp to new volume name

-p PREFIX, --prefix PREFIX
                    Optional Prefix to add to new volume name.

-f, --force
                    Force the copy to overwrite if the new volume name is an existing volume

-o {host,hgroup,volstr}, --objecttype {host,hgroup,volstr}
                    ObjectType of source used, host, hgroup, or volume string. Default=volstr

-ss SOURCE, --source SOURCE
                    Source Host, Host Group, or VolumeString of OBJECTTYPE
```

```
-C, --CMD
                    Show CLI Commands that could be executed to perform the work

-R, --REST
                    Show REST Commands that could be executed to perform the work
```

Files

`/etc/pure.ini`

The Flash Array configuration file. Contains each array name/address and api_key needed for access to the REST interface.

Examples

- To copy all volumes from the Snapshot Group snapgroup3.L1-0806 each with the new prefix of "XXX" and connect to an existing host, xj88:
  snapcopy -s snapgroup3.L1-0806 -p XXX -t xj88

- If the volumes above with prefix "XXX" already exist, then add the -f flag to force and overwrite:
  snapcopy -s snapgroup3.L1-0806 -p XXX -t xj88 -f

- To do the above, but only perform in audit mode (don't actually copy or connect):
  snapcopy -s snapgroup3.L1-0806 -p XXX -t xj88 -a

- To copy only volumes that match the volstr "P66" from the Snapshot Group snapgroup3.L1-0806 each with the new prefix of "XXX" and connect to an existing host, xj88:
  snapcopy -s snapgroup3.L1-0806 -p XXX -t xj88 -o volstr -ss P66

- To copy only volumes currently connected to xx11 from Snapshot Group AIXOS.L1-0008 back to their original volumes (assuming they still exist):
  snapcopy –s AIXOS.L1-0008 -o host -ss xx11 -f

- To do the above in audit mode (don't really overwrite):
  snapcopy –s AIXOS.L1-0008 -o host -ss xx11 -f -a