

# SoftNAS® CLI Guide

The "softnas-cmd" command line interface utility facilitates integration with the REST API as a CLI (command line interface). softnas-cmd is written in CURL and is cross-platform code. It may also be useful as example code for how to use the API, for those who wish to explore it in more detail as a working example. This document has been produced by running `softnas-cmd --help`.

The softnas-cmd CLI is designed to be used from the command line. When running within the Amazon Web Services (AWS) environment, softnas-cli is a convenient tool for use with Cloud Formation templates, which can be combined to automate configuration and setup of SoftNAS storage systems.

As with all SoftNAS API applications, softnas-cmd requires that you first authenticate with the SoftNAS server using the "login" command. Once you are logged in, softnas-cmd maintains a simulated cookie jar that emulates how a browser interacts with a web server to maintain session security (required by the SoftNAS server). Subsequent commands issued via the softnas-cmd CLI make use of the login session (which expires after 30 minutes).

## Usage:

```
softnas-cmd [-h | --help] | [[-b | --base_url URL] [-s | --session_id SESSION_ID] [-i | --insecure] | [-t | --pretty_print] COMMAND [OPTIONS]]
```

## OPTIONS:

```
-h [command name], --help [command name]
```

Display this help message and exit. You can specify the command name to get the help for only one softnas command.

```
-b, --base_url URL
```

Set base url for SoftNAS Application. By default URL=<https://localhost/softnas>

```
-s, --session_id SESSION_ID
```

Set the current session id to be used to used store session information. By default SESSION\_ID=PPID.

```
-i, --insecure
```

Disable SSL verification.

```
-t, --pretty_print
```

Display json result with indentation.

## Commands:

- Usage:
- login
- logout
- resetsessiontimer
- licenseinfo
- licenseactivate
- internallicense
- newlicense
- ackagreement
- checkupdate
- executeupdate
- statusupdate
- meterstatus
- availabledisks
- pools
- pooldetails
- poolcommand
- deletepool
- volumes
- createvolume
- editvolume
- deletevolume
- schedulelist
- snapshotlist
- snapcommand
- iscsitargetlist
- diskdevices
- diskmgmt
- parted\_command
- snapprepcommand
- hacommand
- test\_remote\_address
- get\_s3\_regions
- perfmon
- overview
- netstats
- diskstats
- createpool
- readcache
- writelog
- addspare
- importpool
- backup
- userpassword
- tier\_devices
- tier\_enable
- tier\_disable
- tier\_export

- tier\_import
- tier\_getpolicy
- tier\_setpolicy
- tier\_resize
- tier\_get\_migration\_state
- tier\_set\_migration\_state
- ultrafast\_get\_uuid
- ultrafast\_get\_authorized\_onramps
- ultrafast\_get\_configured\_offramps
- ultrafast\_scheduling
- ultrafast\_authorize\_onramp
- ultrafast\_add\_offramp
- ultrafast\_remove\_offramp
- help

## login

Login into SoftNAS Application.

### COMMAND OPTIONS:

username

password

### EXAMPLE:

```
softnas-cmd login softnas Pass4W0rd
```

## logout

Logout from SoftNAS Application.

### EXAMPLE:

```
softnas-cmd logout
```

## resetsessiontimer

This command is used to keep session active. By default session timeout is set to 30 minutes. if you want to keep session from being expired just call this function.

### EXAMPLE:

```
softnas-cmd resetsessiontimer
```

## licenseinfo

Returns the current license information.

### EXAMPLE:

```
softnas-cmd licenseinfo
```

## licenseactivate

Activate a license key for use with SoftNAS

### COMMAND OPTIONS:

**licencekey:** the licence key; i.e., "softnas"

**register name:** the registered name of the license key

**hardware id:** can be obtained from "licenseinfo" in amazon the hardware id is the instance id

### EXAMPLE:

```
softnas-cmd licenseactivate CEAASA-BESNJA-8MEED6-AHAZZN-XHWB8X-A2NUK3 RBLLC i-0b06fe44
```

## internallicense

Force SoftNAS to use its built-in, default license.

### EXAMPLE:

```
softnas-cmd internallicense
```

## newlicense

Activate a license key for use with SoftNAS

### COMMAND OPTIONS:

**newkey:** the licence key; i.e., "softnas"

**regnew:** the registered name of the license key

**activationType:** manual or online activation type are supported

**activationCode:** manual activation code

## EXAMPLE:

```
softnas-cmd newlicense CEAASA-BESNJA-8MEED6-AHAZZN-XHWP8X-A2NUK3 RBLLC manual AZQWERTY-ACTIV-ATION-CODE
```

## ackagreement

Acknowledge the license agreement (to enable use of the product).

## EXAMPLE:

```
softnas-cmd ackagreement
```

## checkupdate

Check to see if new software updates are available.

## EXAMPLE:

```
softnas-cmd internallicense
```

## executeupdate

Execute and apply software updates.

## EXAMPLE:

```
softnas-cmd executeupdate
```

## statusupdate

Return the status of an update that is in-progress (started by executeupdate).

## EXAMPLE:

```
softnas-cmd statusupdate
```

## meterstatus

Return AWS usage meter status (if available).

### EXAMPLE:

```
softnas-cmd usagemeter
```

## availabledisks

Returns list of available disk devices.

### EXAMPLE:

```
softnas-cmd availabledisks
```

## pools

Lists available storage pools.

### COMMAND OPTIONS:

**start:** position to start from (used for pagination)

**limit:** number of items to get

### EXAMPLE:

```
softnas-cmd pools 1 10
```

## pooldetails

List a storage pool's detailed attributes.

### COMMAND OPTIONS:

**pool name:** the pool name

### EXAMPLE:

```
softnas-cmd pooldetails pool1
```

## poolcommand

Issue a command to control a storage pool.

## COMMAND OPTIONS:

**command name:** zpool command name (startscrub, stopscrub, setonline, setoffline, removedevice, replacedevice)

**pool name:** the pool name

## EXAMPLE:

```
softnas-cmd poolcommand <command> [arguments]
```

## deletepool

Delete pool storage.

## COMMAND OPTIONS:

**pool name:** the pool name

## EXAMPLE:

```
softnas-cmd deletepool pool1
```

## volumes

List available storage volumes.

## COMMAND OPTIONS:

**start:** position to start from (used for pagination)

**limit:** number of items to get

## EXAMPLE:

```
softnas-cmd volumes 1 10
```

## createvolume

Create a volume.

## COMMAND OPTIONS:

**vol\_name:** name of volume

**pool:** selected pool name

**vol\_type:** possible values :

- **filesystem:** Filesystem (NFS, CIFS)
- **blockdevice:** Block Device (iSCSI LUN)

**provisioning:** allocation space type. Possible values :

- **thin:** Thin Provision - dynamically allocate space as it is needed
- **thick:** Thick Provision - preallocate space from storage pool now

**reserve\_space:** volume size (number)

**reserve\_units:** size unit i.e G (Giga)

**compression:** enable compression

**dedup:** enable deduplication

**shareISCS:** enable share ISCSI

**shareCIFS:** enable share CIFS

**exportNFS:** enable export NFS

**enable\_snapshots:** Enable scheduled volume snapshots

**schedule\_name:** Snapshot schedule name

**hourlysnaps:** hourly maximum number of scheduled snapshot

**daily snaps:** daily maximum number of scheduled snapshot

**weeklysnaps:** weekly maximum number of scheduled snapshot

**minimum\_threshold:** Minimum amount of data written to volume before new snapshot is created. If user have dozens of volumes, it prevents high system load caused by creating lot of unnecessary 0-size snapshots and of sending them via SnapReplicate.

**sync:** controls the behavior of synchronous requests



- **standard:** standard is the POSIX specified behavior of ensuring all synchronous requests are written to stable storage and all devices are flushed to ensure data is not cached by device controllers (this is the default).
- **always:** always causes every file system transaction to be written and flushed before its system call returns. This has a large performance penalty.
- **disabled:** this disables synchronous requests. File system transactions are only committed to stable storage periodically. This option will give the highest performance. However, it is very dangerous as ZFS would be ignoring the synchronous transaction demands of applications such as databases or NFS. Administrators should only use this option when the risks are understood.

## EXAMPLE:

```
softnasm-cmd createvolume vol_name=volume5 pool=pool3 vol_type=filesystem provisioning=thin exportNFS=on
shareCIFS=on dedup=on enable_snapshot=on schedule_name=Default hourlysnaps=5 dailysnaps=10 weeklysnaps=0
sync=always
```

## editvolume

Edit a volume.

## COMMAND OPTIONS:

**reserve\_space:** integer value for volume capacity (G,M,T)

**reserve\_units:** unit for volume capacity (G, M, T)

**compression:** ZFS compression type

**dedup:** deduplication

**share via iSCSI** (exclude completely to disable)

**share via CIFS** (exclude completely to disable)

**export via NFS** (exclude completely to disable)

**enable\_snapshot:** enable the identified snapshot

**schedule\_name:** string identifier that represents schedule

## EXAMPLE:

```
softnas-cmd availabledisks
```

## deletevolume

Delete a volume.

### COMMAND OPTIONS:

**volume name:** name of volume

**pool name:** pool name

### EXAMPLE:

```
softnas-cmd deletevolume volume6 pool3
```

## schedulelist

List available schedules.

### EXAMPLE:

```
softnas-cmd schedulelist
```

## snapshotlist

List available snapshots.

### COMMAND OPTIONS:

**pool name:** pool name i.e 'pool3/vol1'

**start:** position to start from (used for pagination)

**limit:** number of items to get

### EXAMPLE:

```
softnas-cmd snapshotlist pool3/vol1 0 10
```

## snapcommand

Issue a volume snapshot control command.

## COMMAND OPTIONS:

**command:** possible values

- **create:** create new snapshot. Related options:

- **pool\_name:** pool name i.e 'pool3/vol1'

- **volume\_name:** volume name

- **clone:** clone snapshot. Related options:

- **pool\_name:** pool name i.e 'pool3/vol1'

- **volume\_name:** volume name

- **snap\_name:** snapshot name to be cloned

- **delete:** delete snapshot list

- **snapshots:** list of snapshot to be deleted in this format "poolname,volumename,snapshotname: poolname1,volumename1,snapshotname1". see example

## EXAMPLE 1 :

Create snapshot.

```
softnas-cmd snapcommand create pool_name=pool1 volume_name=vol1
```

## EXAMPLE 2 :

Clone snapshot.

```
softnas-cmd snapcommand clone pool_name=pool1 volume_name=vol1 snap_name=snapshot1
```

## EXAMPLE 3 :

Delete snapshots.

```
softnas-cmd snapcommand delete 'snapshots=pool1,volumel,snap-20150605-140647-cloned-Jun52015-141106:pool1,volumel,snap-20150606-070607'
```

## iscsitargetlist

Provides a list of available iSCSI targets.

## COMMAND OPTIONS:

start: position to start from (used for pagination)

limit: number of items to get

## EXAMPLE:

```
softnas-cmd iscsitargetlist 0 10
```

## diskdevices

List of available disk devices and their status

## EXAMPLE:

```
softnas-cmd diskdevices
```

## diskmgmt

Issue a disk management command.

## COMMAND OPTIONS:

**command:** possible values

- **createS3disk:** create s3 disk. Related options:

- **AWSAccessKey:** s3 AWS access key id
- **AWSSecretKey:** s3 AWS secret key
- **s3bucket:** s3 bucket name must be unique
- **bucketroot:** s3 bucket root
- **sizeMaxValue:** Disk size
- **sizeMaxUnits:** Size unit TB or GB
- **diskpassword:** disk password protection
- **encrypted:** add it to encrypt disk

- **readahead**: loads file contents into page cache
- **blockCacheDevice**: s3 cache device
- **region**: AWS region. Possible values :
  - **us-west-2, oregon**: oregon
  - **us-west-1, northern\_california**: northern california
  - **eu-west-1, ireland**: ireland
  - **ap-southeast-1, singapore**: singapore
  - **ap-southeast-2, sydney**: sydney
  - **ap-northeast-1, tokyo**: tokyo
  - **ap-northeast-1, seoul**: seoul
  - **sa-east-1, sao\_paulo**: sao paulo
  - **eu-central-1, frankfurt**: frankfurt
  - **us-gov-west-1, govcloud**: govcloud
  - **us-east-1, virginia, default**: virginia
- **deleteS3disk**: delete s3 disk
  - **s3diskname**: s3 disk name i.e (/dev/s3-0)
- **deleteAzureDisk**: delete Azure disk.
  - **diskname**: Name of device (i.e. /dev/sdb)
- **getS3settings**: get list of s3 setting stored in config file
- **bucketlist**: get list of AWS s3 buckets
- **createEBSDisk**: create EBS disk. Related options :
  - **AWSAccessKey**: s3 AWS access key id
  - **AWSSecretKey**: s3 AWS secret key
  - **sizeMaxValue**: Disk size in GB

- **type:** Disk type. Possible values:
  - **gp2** : General Purpose (SSD)
  - **io1:** Provisioned IOPS (SSD). Related option:
    - **iops:** Number of Input/Output Operations Per Second
  - **standard:** Standard
- **numberDisks:** Number of disks to add
- **deleteOnTermination:** Enable delete on termination
- **prewarming:** Enable pre-warming
- **createAzureDisk:** create Azure disk. Related options:
  - **username:** Azure account username
  - **password:** Azure account password
  - **storageAccount:** Azure storage account
  - **sizeMaxValue:** Disk size
  - **sizeMaxUnits:** Size unit TB or GB
  - **numberDisks:** Number of disks to add
  - **sync\_creation** : Waiting for all disks to be created before getting the response
- **createExtenderDisk:** create extender disk. Related options:
  - **type:** establishes disk type. Possible values are: centurylink, cloudian, dunkel, google, hitachi, netapp, azureBlob, custom
  - **accessKey:** access key for Azure account
  - **secretKey:** secret key for Azure account
  - **bucketName:** Azure bucket name
  - **sizeMaxValue:** Disk size
  - **sizeMaxUnits:** Size unit TB or GB

- **encrypted**: password for encrypting the disk
- **awsAuthVersion**: Possible values: aws2, aws4

## EXAMPLE 1:

Create s3 disk.

```
softnas-cmd diskmgmt createS3disk AWSAccessKey=YOUR_AWS_ACCESS_KEY_ID AWSSecretKey=YOUR_AWS_SECRET_KEY  
s3bucket=softnas-test-1 bucketroot=softnas sizeMaxValue=500 sizeMaxUnits=GB region=default encrypted  
diskpassword=password123* readahead blockCacheDevice=/dev/xvdf
```

## EXAMPLE 2:

Get list of buckets.

```
softnas-cmd diskmgmt bucketlist
```

## EXAMPLE 3:

Get list of s3 setting stored in config file.

```
softnas-cmd diskmgmt getS3settings
```

## EXAMPLE 4:

Delete S3 disk.

```
softnas-cmd diskmgmt deleteS3disk "/dev/s3-0"
```

## EXAMPLE 5:

Create EBS Disk General Purpose (SSD).

```
softnas-cmd diskmgmt createEBSdisk AWSAccessKey=XXXXXXXXXXXXXXXXXXXX  
AWSSecretKey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX' sizeMaxValue=10 type=gp2 numberDisks=2 deleteOnTermination  
prewarming -t
```

## EXAMPLE 6:

Create EBS Disk Provisioned IOPS (SSD).

```
softnas-cmd diskmgmt createEBSdisk AWSAccessKey=XXXXXXXXXXXXXXXXXXXX  
AWSSecretKey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX' sizeMaxValue=100 type=io1 iops=3000 numberDisks=2  
deleteOnTermination prewarming -t
```

## EXAMPLE 7 :

Create Azure Disk.

```
softnas-cmd diskmgmt createAzureDisk username=softnas password=Pass4W0rd storageAccount=SoftnasStorage  
sizeMaxValue=100 sizeMaxUnits=GB -t
```

## parted\_command

Issue a disk partitioning command.

## COMMAND OPTIONS:

**command:** possible values

- **partition\_all:** partition all disks that are not partitioned
- **add\_partition:** adds a partition to specified disk/s
- **remove\_partition:** removes a partition from specified disk/s
- **disk name:** selected disk device name

## EXAMPLE:

```
softnas-cmd parted_command add_partition dev/xvdr
```

## snaprepcommand

Issue SnapReplicate command.

## COMMAND OPTIONS:

**command:** possible values

- **snapverify:** verify the remote node. Related options:
  - **remotenode:** ip address of remote node
  - **userid:** username for remote node
  - **password:** password of remote node
- **initsnapreplicate:** begin initial replicate with remote node
  - **type:** type of source node possible values source or target
  - **remotenode:** ip address of remote node
  - **userid:** username for remote node
  - **password:** password of remote node
- **keyexchange:** Exchange servers keys and allow remote servers ssh connection for each node
  - **keyfile:** public key file path (/var/www/softnas/keys/SoftNAS-PrimaryPublic.pem)
  - **pubEncoded:** Keyfile content base64 encoded



- **serverEncoded**: rsa public host key content file base 64 encoded (/etc/ssh/ssh\_host\_rsa\_key.pub)
- **serverIP**: server node ip (sender)
- **snapreplicatetestatus**: get the current status of replication
  - **start**: position to start from (used for pagination)
  - **limit**: number of items to get
- **snapreplug**: get snapreplicate log
  - **start**: position to start from (used for pagination)
  - **limit**: number of items to get
- **snapreplicatetasks**: get snapreplicate tasks
  - **start**: position to start from (used for pagination)
  - **limit**: number of items to get
- **forcesync**: force full synchronization of all volumes image or specific volume between nodes
  - **volume\_path**: optional, you can specify a volume (poolname/volumename) to force full synchronization
- **replicatenow**: get snapreplicate log
- **activate**: activate replication
- **deactivate**: deactivate replication
- **takeover**: takeover as replication source
- **giveback**: giveback replication source duties
- **deletereplication**: remove the replication
- **getsettings**: get replication settings
- **modifysettings**: edit replication settings
- **loglevel**: Logging Level. Possible values
  - **info**: Informational, warning and error messages (default)
  - **debug**: Debug, informational, warning and error messages (all messages)

- **warn**: Warning and error messages
- **error**: Error messages only
- **fatal**: Fatal messages only
- **off**: No messages (not recommended)
  
- **transportcmd**: Transport Command
- **transportflags**: Transport Flags
- **cipherspec**: Cipher Spec
- **compressenabled**: Compress data stream (consumes additional CPU)
- **throttleenabled**: enable bandwidth Throttle (per stream)
- **throttlelimit**: limit size
- **throttleunits**: size unit possible values
  - **b** => bits/sec
  - **k** => Kbits/sec
  - **m** => Mbits/sec
  - **B** => Bytes/sec
  - **K** => Kbytes/sec
  - **M** => MBytes/sec

## EXAMPLE 1:

Verify the SnapReplicate remote node.

```
softnas-cmd snaprepcommand snapverify remotenode=10.227.57.227 userid=softnas password=Pass4W0rd -t
```

## EXAMPLE 2:

Initiate snapreplicate.

```
softnas-cmd snaprepcommand initsnapreplicate remotenode=10.227.57.227 userid=softnas password=Pass4W0rd  
type=source -t
```

## EXAMPLE 3:

Get snapreplicate status.

```
softnas-cmd snaprepcommand snapreplicatetest status start=0 limit=10 -t
```

#### EXAMPLE 4:

Get snapreplicate log

```
softnas-cmd snaprepcommand snapreplog start=0 limit=10 -t
```

#### EXAMPLE 5:

Delete snapreplication.

```
softnas-cmd snaprepcommand deletereplication -t
```

#### EXAMPLE 6:

Get replication settings.

```
softnas-cmd snaprepcommand getsettings -t
```

#### EXAMPLE 7:

Force full synchronization of all volumes images between 2 nodes

```
softnas-cmd snaprepcommand forcesync -t
```

#### EXAMPLE 8:

Force full synchronization volume image between 2 nodes. You can specify a volume (poolname/volumename) to force full synchronization.

```
softnas-cmd snaprepcommand forcesync poolname/volumename -t
```

#### EXAMPLE 9:

Start replication cycle.

```
softnas-cmd snaprepcommand replicatenow -t
```

#### EXAMPLE 10:

Activate replication.

```
softnas-cmd snaprepcommand activate -t
```

#### EXAMPLE 11:

Deactivate replication

```
softnas-cmd snaprepcommand deactivate -t
```

#### EXAMPLE 12:

Change role to primary node.

```
softnas-cmd snaprepcommand takeover -t
```

### EXAMPLE 13:

Change role to secondary node.

```
softnas-cmd snaprepcommand giveback -t
```

### EXAMPLE 14:

Modify replication settings

```
softnas-cmd snaprepcommand modifysettings loglevel=info transportcmd=ssh transportflags="-o ConnectTimeout=30"  
cipherspec="aes128-cbc,blowfish-cbc,3des-cbc,cast128-cbc,aes192-cbc,aes256-cbc" compressenabled=on  
throttleenabled=on throttlelimit=200 throttleunits=K throttleflags= -t
```

## hacommand

Issue ha command.

### COMMAND OPTIONS:

**command:** possible values:

- **checklicense:** check for valid ha license
  - **haLicenseKey:** Ha license key (required)
  - **regname:** license registration name (optional). If not specified it will use softnas.ini registration name.
- **checkHAcontroller:** Check for valid HA Controller IP address
  - **haControllerIP:** Controller IP address
- **install:** Begin install HA from the SnapReplicate "source" node
  - **haLicenseKey:** Ha license key (required)
  - **regname:** license registration name (optional). If not specified it will use softnas.ini registration name.
- **add:** Begin Add configuration of SNAP HA from the SnapReplicate "source" node
  - **AWSAccessKey:** AWS Access Key
  - **AWSSecretKey:** AWS Secret Key

- **ha\_type**: Possible values :
  - **VIP**: for private virtual ip
  - **EIP**: for AWS elastic IP
- **vip**: virtual IP (AWS Elastic IP)
- **haControllerIP**: Controller IP address (required only on Virtual machine)
- **bypassHAchecks** : If given, bypass pre-checks for HA installation
- **del**: Delete HA from SnapReplicate
- **azure\_install**: Begin install Azure HA from the SnapReplicate "source" node
  - **azureUsername**: The main azure username credential (required)
  - **azurePassword**: The main azure password credential (required)
  - **vip**: A virtual IP address (VIP or VIPA) is a private IP address that doesn't correspond to an actual physical network interface (port). ie 192.168.1.1 (required)
  - **ha\_mon\_secs**: Timeout interval for HA health check (required)
  - **ha\_num\_retries**: Number of retries for HA health check (required)
  - **tenant\_uuid**: Service principal UUID (if using service principal)
- **azure\_delete**: Delete Azure HA from SnapReplicate

## EXAMPLE 1:

Check HA license.

```
softnas-cmd hacommand checklicense XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX --pretty_print
```

## EXAMPLE 2:

Install HA.

```
softnas-cmd hacommand install XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX --pretty_print
```

## EXAMPLE 3:

Add HA.

```
softnas-cmd hacommand add YOUR_AWS_ACCESS_KEY YOUR_AWS_SECRET_KEY EIP 54.84.87.120 --pretty_print
```

### EXAMPLE 4:

check HA controller

```
softnas-cmd hacommand checkHAcontroller 50.15.14.15 --pretty_print
```

### EXAMPLE 5:

Delete HA.

```
softnas-cmd hacommand del --pretty_print
```

### EXAMPLE 6:

Azure install HA.

```
softnas-cmd hacommand azure_install azureUsername azurePassword 192.168.1.1 30 4 --pretty_print
```

### EXAMPLE 7:

Azure install HA with Service Principal.

```
softnas-cmd hacommand azure_install https://USERNAME.onmicrosoft.com/UUID azurePassword 192.168.1.1 30 4  
TENANT_ID --pretty_print
```

### EXAMPLE 8:

Azure delete HA.

```
softnas-cmd hacommand azure_delele --pretty_print
```

## test\_remote\_address

Check if address with port is reachable from instance

### COMMAND OPTIONS:

**remote\_address** : ip address or domain trying to reach

**remote\_port** : port on that address (optional)

### EXAMPLE:

```
softnas-cmd test_remote_address 192.168.0.5 8081
```

## get\_s3\_regions

get s3 array of regions

### COMMAND OPTIONS:

**include\_all** :get all regions, not just AWS regions

**gov** : get also AWS gov region

### EXAMPLE:

```
get_s3_regions
get_s3_regions include_all=true
get_s3_regions gov=true
get_s3_regions include_all=true gov=true
```

## perfmon

Get performance monitoring status information.

**IMPORTANT:** This command gives 30 records. You should call this command in series between each call less than 10 sec to get performance monitoring status information. For the first call this command will return all values as false and each next call with new record values.

### EXAMPLE:

```
softnas-cmd perfmon
```

## overview

Get NAS overview status information.

### EXAMPLE:

```
softnas-cmd overview
```

## netstats

Get network performance status information.

### EXAMPLE:

```
softnas-cmd netstats
```

## diskstats

Get disk performance status information.

### EXAMPLE:

```
softnas-cmd diskstats
```

## createpool

Creates new storage pool.

### COMMAND OPTIONS:

**disk devices:** list of diskdevices separated by comma ":"

**-n pool name:** the pool name

**-r raid level:** possible values:

0 : No RAID, JBOD

1 : RAID 1/10 (mirror, striped mirrors)

5 : RAID-Z (single parity)

6 : RAID-Z2 (dual parity)

7 : RAID-Z3 (triple parity)

**-f forced creation :** possible values on, off

**-le use LUKS encryption :** possible values on, off, unset (off)

**-lp LUKS passphrase :** must be provided when LUKS encryption is on

**-sync sync mode :** possible values standard, always, disabled

**-cs case sensitive :** possible values on, off

**-shared shared :** possible values on, off

### EXAMPLE:

```
softnas-cmd createpool /dev/xvdf:/dev/xvdg -n=pool1 -r=1 -f=on -t
```

```
softnas-cmd createpool /dev/xvdf:/dev/xvdg -n=pool1 -r=1 -f=on -le=on -lp=luksPassword.123 -t
```

```
softnas-cmd createpool /dev/xvdf:/dev/xvdg -n=pool1 -r=1 -f=on -le=on -lp=Pas4Wrd -sync=standard -cs=off -t
```



```
softnas-cmd createpool /dev/s3-1:/dev/s3-2 -n=pool1 -r=1 -f=on -sync=standard -cs=off -shared=on -t
```

## readcache

Create cache disk devices for pool storage.

### COMMAND OPTIONS:

**disk devices:** list of diskdevices separated by comma ":"

**pool name:** the pool name

**force cache:** possible values on, off

### EXAMPLE:

```
softnas-cmd readcache /dev/xvdr pool3 on
```

## writelog

Create log disk devices for pool storage.

### COMMAND OPTIONS:

**disk devices:** list of diskdevices separated by comma ":"

**pool name:** the pool name

**raid level:** possible values 0, 1

0 : No RAID, JBOD

1 : RAID 1/10 (mirror, striped mirrors)

**force cache:** possible values on, off

### EXAMPLE:

```
softnas-cmd writelog /dev/xvdk pool3 0 on
```

## addspare

Add spare disk to storage pool.

## COMMAND OPTIONS:

**disk devices:** list of diskdevices separated by comma ":"

**pool name:** the pool name

**forced spare:** possible values on, off

## EXAMPLE:

```
softnas-cmd addspare /dev/xvdc pool3 on
```

## importpool

Import deleted or foreign pools.

## COMMAND OPTIONS:

**pool name:** the pool name

**pool type:** possible values deleted, foreign

**force:** possible values on, off

## EXAMPLE:

```
softnas-cmd importpool pool1 deleted on
```

## backup

Issue a EBS snapshots backup command.

## COMMAND OPTIONS:

**command:** possible values

- **list:** list ebs backup snapshots. Related options:

- **start:** position to start from (used for pagination) (optional)

- **limit:** number of items to get (optional)

- **create:** create new ebs backup snapshot. Related options:

- **access\_key:** AWS S3 access key id (required)

- **secret\_key:** AWS S3 secret key (required)

- **restore:** restore ebs snapshot. Related options:

- **SnapshotID:** Snapshot ID to restore (required)

- **access\_key:** AWS S3 access key id (required)

- **secret\_key:** AWS S3 secret key (required)

- **delete:** clone snapshot. Related options:

- **SnapshotID:** Snapshot ID to delete (required)

## EXAMPLE 1:

List ebs backup snapshots.

```
softnas-cmd backup list 0 10 --pretty_print
```

## EXAMPLE 2:

Clone snapshot.

```
softnas-cmd backup create 'yourAWSAccessKey' 'yourAWSecretKey' --pretty_print
```

## EXAMPLE 3:

Restore ebs snapshot.

```
softnas-cmd backup restore 'SnapshotID' 'yourAWSAccessKey' 'yourAWSecretKey' --pretty_print
```

## EXAMPLE 4:

Delete ebs snapshot.

```
softnas-cmd backup delete 'SnapshotID' --pretty_print
```

## userpassword

change password for specified user

## COMMAND OPTIONS:

**user** : username (ex. softnas)  
**oldpassword** : old user password  
**newpassword** : new user password

### EXAMPLE:

```
softnas-cmd userpassword user=softnas
```

## tier\_devices

List tiered storage devices.

### EXAMPLE:

```
softnas-cmd tier_devices
```

## tier\_enable

Mark tiered storage device as enabled (but do not import it).

### COMMAND OPTIONS:

**name:** human-readable tier name (ex. tierA)

**device:** device node (ex. sdtiera)

**uuid:** tier uuid

### EXAMPLE:

```
softnas-cmd tier_enable name=tierA
```

### EXAMPLE:

```
softnas-cmd tier_enable device=sdtiera
```

### EXAMPLE:

```
softnas-cmd tier_enable uuid=5B074FFFFFF23FFFF56FFFFFF
```

## tier\_disable

Mark tiered storage device as disabled (but do not export it)

## COMMAND OPTIONS:

**name:** human-readable tier name (ex. tierA)

**device:** device node (ex. sdtiera)

**uuid:** tier uuid

## EXAMPLE:

```
softnas-cmd tier_disable name=tierA
```

## EXAMPLE:

```
softnas-cmd tier_disable device=sdtiera
```

## EXAMPLE:

```
softnas-cmd tier_disable uuid=5B074FFFFFF23FFFF56FFFFFF
```

## tier\_export

Stop tiered storage device if it is not in use without marking as disabled (will auto-import on reboot or takeover)

## COMMAND OPTIONS:

**name:** human-readable tier name (ex. tierA)

**device:** device node (ex. sdtiera)

**uuid:** tier uuid

## EXAMPLE:

```
softnas-cmd tier_export name=tierA
```

## EXAMPLE:

```
softnas-cmd tier_export device=sdtiera
```

## EXAMPLE:

```
softnas-cmd tier_export uuid=5B074FFFFFF23FFFF56FFFFFF
```

## tier\_import

Activate an exported tier storage device.

## COMMAND OPTIONS:

**name:** human-readable tier name (ex. tierA)

**uuid:** tier uuid

## EXAMPLE:

```
softnas-cmd tier_import name=tierA
```

## EXAMPLE:

```
softnas-cmd tier_import uuid=5B074FFFFFF23FFFF56FFFFFF
```

## tier\_getpolicy

Display current storage migration policy JSON document

## COMMAND OPTIONS:

**name:** human-readable tier name (ex. tierA)

**device:** device node (ex. sdtiera)

**uuid:** tier uuid

## EXAMPLE:

```
softnas-cmd tier_getpolicy name=tierA
```

## EXAMPLE:

```
softnas-cmd tier_getpolicy device=sdtiera
```

## EXAMPLE:

```
softnas-cmd tier_getpolicy uuid=5B074FFFFFF23FFFF56FFFFFF
```

## tier\_setpolicy

Apply new storage migration policy.

## COMMAND OPTIONS:

**name:** human-readable tier name (ex. tierA)

**device:** device node (ex. sdtiera)

**uuid:** tier uuid

**policy:** JSON string encapsulated in " (retrieve using tier\_getpolicy)

## EXAMPLE:

```
softnas-cmd tier_setpolicy {name=tierA|device=sdtiera|uuid=5B074FFFFFF23FFFF56FFFFFF} policy='[{"hit_collecttime": "23250", "max_age": "46401", "device": "zd0", "order": "0"}, {"hit_collecttime": "43200", "max_age": "86400", "device": "zd16", "order": "1"}]'
```

## tier\_resize

Expand tiered storage device when underlying volumes have grown.

## COMMAND OPTIONS:

**name:** human-readable tier name (ex. tierA)

**device:** device node (ex. sdtiera)

**uuid:** tier uuid

## EXAMPLE:

```
softnas-cmd tier_resize name=tierA
```

## EXAMPLE:

```
softnas-cmd tier_resize device=sdtiera
```

## EXAMPLE:

```
softnas-cmd tier_resize uuid=5B074FFFFFF23FFFF56FFFFFF
```

## tier\_get\_migration\_state

return the current data migration status for the selected tier

## COMMAND OPTIONS:

**name:** human-readable tier name (ex. tierA)

**device:** device node (ex. sdtiera)

**uuid:** tier uuid

\* If no device is given, all tier states will be returned

### EXAMPLE:

```
softnas-cmd tier_get_migration_state name=tierA
softnas-cmd tier_get_migration_state device=sdtiera
softnas-cmd tier_get_migration_state uuid=5B074FFFFFF23FFFF56FFFFFF
```

## tier\_set\_migration\_state

enable or disable background tier storage migration - when re-enabled, migration begins immediately.

### COMMAND OPTIONS:

**name:** human-readable tier name (ex. tierA)

**device:** device node (ex. sdtiera)

**uuid:** tier uuid

**state :** enabled or disabled (default: enabled)

\* If no device is given, all tier states will be returned

### EXAMPLE:

```
softnas-cmd tier_set_migration_state state=enabled name=tierA
softnas-cmd tier_set_migration_state state=disabled device=sdtiera
softnas-cmd tier_set_migration_state uuid=5B074FFFFFF23FFFF56FFFFFF
```

## ultrafast\_get\_uuid

enable or disable background tier storage migration - when re-enabled, migration begins immediately.

### EXAMPLE:

```
softnas-cmd ultrafast_get_uuid
```



## ultrafast\_get\_authorized\_onramps

return the list of onramps (sources) authorized to connect to this offramp (target)

### EXAMPLE:

```
softnas-cmd ultrafast_get_authorized_onramps
```

## ultrafast\_get\_configured\_offramps

return the list of offramps (targets) this onramp (source) was configured to connect to

### EXAMPLE:

```
softnas-cmd ultrafast_get_uuid
```

## ultrafast\_scheduling

return the list of schedule containing hourly bandwidth throttles that can be set (or is currently set) to a connection

### EXAMPLE:

```
softnas-cmd ultrafast_scheduling
```

## ultrafast\_authorize\_onramp

authorize an onramp (source) to connect to this offramp (target)

### COMMAND OPTIONS:

**onramp uuid** : uuid of the onramp

**onramp secret** : onramp secret string

**offramp secret** : offramp secret string

### EXAMPLE:

```
softnas-cmd ultrafast_authorize_onramp 62f0d535-167c-4c34-9a00-002893c12a89 9397b58923c595ee 3ecf7d521bc8068c
```

## ultrafast\_add\_offramp

configure an offramp (target) this onramp (source) will automatically connect to

### COMMAND OPTIONS:

**offramp uuid** : uuid of the offramp

**offramp dns** : dns/ip of the offramp

**offramp udp port**: configured offramp udp port used for the tunnel communication

**onramp secret** : onramp secret string

**offramp secret** : offramp secret string

**max upstream** : maximum upstream bandwidth throttle

**max downstream** : maximum downstream bandwidth throttle

**schedule uuid** : uuid of the schedule info containing hourly bandwidth throttles (optional)

### EXAMPLE:

```
softnas-cmd ultrafast_add_offramp 62f0d535-167c-4c34-9a00-002893c12a89 172.16.0.148 8888 9397b58923c595ee
3ecf7d521bc8068c 1000 1000
softnas-cmd ultrafast_add_offramp 62f0d535-167c-4c34-9a00-002893c12a89 172.16.0.148 8888 9397b58923c595ee
3ecf7d521bc8068c 1000 1000 62f0d535-167c-4c34-9a00-002893c12a8a
```

## ultrafast\_remove\_offramp

remove an offramp (target) this onramp (source) will automatically connect to (or is currently connected with)

### COMMAND OPTIONS:

**offramp uuid** : uuid of the offramp

### EXAMPLE:

```
softnas-cmd ultrafast_remove_offramp 62f0d535-167c-4c34-9a00-002893c12a89
```

## help

Display help text.

### COMMAND OPTIONS:

**command name** : Optional. If you wish to know the help content for a particular command, add the command name.

### EXAMPLE:

softnas-cmd help snaprepcommand