SoftNAS Cloud® API and CLI Developers Guide

User guide
2016

Table of Contents

Introduction to the SoftNAS Cloud® Rest API

This manual describes a web services programming interface to the **SoftNAS Cloud®** server.

The interface includes the following:
• Licensing
• Storage Pools
• Volumes
• Disk Devices
• Snapshot Management
• Performance Monitoring

Introduction to the API

The **SoftNAS Cloud® Rest API** provides access to the PHP-based **SoftNAS Cloud®** admin server. This provides access to SSD and disk device management on the ESXi, and EC2 hosts. The **SoftNAS Cloud® API** can be programmed in any language that supports HTTPS requests and responses, including Javascript with Ajax, PHP, cURL, PERL, .NET, Java, etc.

The **SoftNAS Cloud® Rest API** uses **GET**, **POST**, **PUT** and **DELETE** requests sent over HTTPS connections to the Apache web server running on Linux, which in turn are processed by the PHP-based **SoftNAS Cloud®** admin server. The **SoftNAS Cloud®** admin server returns its responses as JSON-formatted strings via the HTTP response. The **SoftNAS Cloud® Rest API** offers a way for third party systems to access the same API that is used by the **SoftNAS StorageCenter** administration GUI. This provides access to the **SoftNAS Cloud®** admin server, which manages the **SoftNAS Cloud®** run-time environment.

In addition,softnas-cmd is a command-line utility written in cURL, provides access to the same API calls from the Linux command line. The command line operations are defined in the **Command Reference** section.

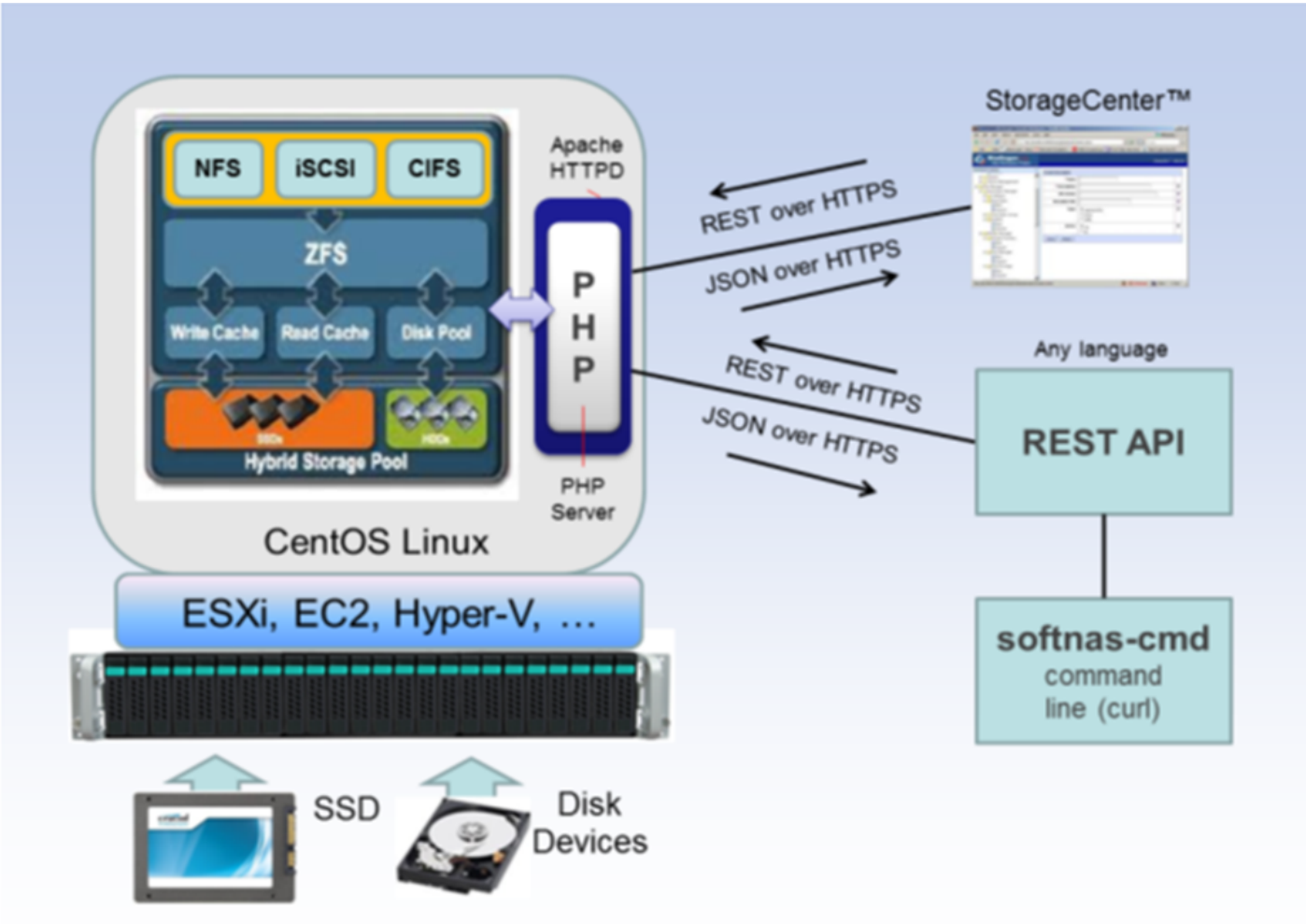


Figure 1 – SoftNAS API Architecture

List of API Operations

See section **Methods** for a complete list of API Operations.

REST API Structure

Each REST API call is structured as an **opcode**, followed by one or more **operands**. This is conceptually similar to how the **von Neumann CPU architecture** organizes CPU instructions and operands.

REST Verb: all commands are sent over HTTPS using a REST verb: GET, POST, PUT, DELETE.
Opcode: this is a string value specifying the API operation code, which determines which admin processor is called to process the request
Command: this (optional) string specifies an additional command to be performed by the admin processor
Operands: these additional string arguments (e.g., GET arguments, POST parameters) vary based upon the particular Opcode and Command passed and provide additional parameters to the admin processor.
JSON Response – the admin processor returns the result of the operation, along with any parameters in a JSON formatted response string.

For a definition of the REST API arguments and parameters, see section **Properties**.

Parameters

This section includes the definitions for the arguments and parameters which make up the Opcodes and Commands of the REST API.

General SoftNAS Cloud API Parameters

Parameter	Description	Note/Data Type
username	username (e.g., **softnas**) Linux user name that will be used to log into **SoftNAS Cloud®** via the API	Alphanumeric String
password	The user's password for the username used to log into **SoftNAS Cloud®** via the API.	Alphanumeric String
-s (number)	Session ID for the client session with **SoftNAS Cloud®**. Every new login to **SoftNAS Cloud®** via the API will receive a unique session ID. The Session ID is returned from the **SoftNAS Cloud®** server and indicates the session ID which will be used for subsequent requests, and for storing information (eg. cookies). It is also possible to specify the session ID for a request using this parameter. If session ID is left unspecified, **SoftNAS Cloud®** will use the PPID (Parent Process ID) as Session ID.	Integer
--session_id (number)	Session ID for the client session with **SoftNAS Cloud®**. Every new login to **SoftNAS Cloud®** via the API will receive a unique session ID. The Session ID is returned from the **SoftNAS Cloud®** server and indicates the session ID which will be used for subsequent requests, and for storing information (eg. cookies). It is also possible to specify the session ID for a request using this parameter. If session ID is left unspecified, **SoftNAS Cloud®** will use the PPID (Parent Process ID) as Session ID.	Integer
-b (url)	Base URL: Same as the location of the **SoftNAS StorageCenter** administrative user interface. `-b https://nas01.mydomain.com/softnas` By default, base url is `https://localhost/softnas`	URL String
--base_url (url)	Base URL: Same as the location of the **SoftNAS StorageCenter** administrative user interface. `-b https://nas01.mydomain.com/softnas` By default base url is “https://localhost/softnas”	URL String
-t	Pretty Print: Causes the JSON return string from the API to be formatted with newlines so that it word wraps nicely on a display.	CLI Command Line Switch.
--pretty_print	Pretty Print: Causes the JSON return string from the API to be formatted with newlines so that it word wraps nicely on a display.	CLI Command Line Switch
success	Returns true or false depending on success or failure of the API request for issued command.	Boolean
start	Start position for the query results. This property can be used when results must be returned using multiple pages. The start position instructs the server to begin returning additional results from a specific item number. **Default: 0**	Integer

limit	For paged query results, this parameter is used to limit request results to a specific number of items. For example, when obtaining list of storage pools, it is possible to limit the amount of results that are returned. **Default:999**	Integer
returnGlobals	-obtain definition	

Licensing

Licensing Parameters

Parameter	Description	Note/Data Type
actual-storage-GB	Actual storage in use, measured in gigabytes.	String
actual-expiration	Expiration date of license including grace period.	Date String
capabilities	Internal use only.	String
capacityGB	Licensed maximum capacity in gigabytes.	String
currentkey	License key.	String
Msg	Message appropriate to the request, based on success or failure.	String
expiration	Expiration date of the license.	Date String
gracedays	Number of grace days following license expiration that product continues to operate.	Integer String
graceperiod	If a **SoftNAS Cloud®** license expires in a production environment, it will enter the grace period. “1” means grace period is in effect.	Boolean String
graceremaining	Amount of days remaining in the grace period, before the grace period expires and **SoftNAS Cloud®** functionality is impacted.	Boolean String
hardware_id	The hardware ID used for license activation. On AWS, the instance ID; otherwise, IP address of admin interface.	String
ipmode	IP address mode, how the default IP address used for licensing was assigned “STATIC” or “DHCP”	String
is_activated	Reports if the license is active or not. True or False.	Boolean String
is_perpetual	Reports if the license is a perpetual license or not. True or false.	Boolean String
is_subscription	Reports of the license type is subscription-based (monthly or annual).	Boolean String
istrial	Reports if the product is currently running in trial mode.	Boolean String
license-version	Licensed software version of **SoftNAS Cloud®**.	String
licensetype	Reports the license type: monthly, annual, subscription, trial etc.	String
maint_expiration	Reports the date at which the maintenance package expires.	Date String
maint_expired	Reports if the status of the maintenance package is expired. True or False.	Boolean String
model	The license model: “key” (licensed by key) or “utility”. (Licensed by the hour).	String
platform	Platform on which **SoftNAS Cloud®** is running, i.e., Amazon Web Services, Microsoft Azure, VMWare vSphere, VMware vCloud Air	String
product-id	**SoftNAS Cloud®** product ID	String
producttype	**SoftNAS Cloud®** product type.	String
regname	Registered name – customer name associated with an activated license key.	String
sig	internal use only.	
status	Reports of the current license is valid. Possible values: valid license or an error message	String

storage-capacity-GB	Reports total storage capacity applicable to the current license in gigabytes.	String
today	Reports today's date.	Date String
totalstorageGB	Total storage consumed currently by storage pools.	String
valid	True if the license is valid.	Boolean String

Avail Disks

Parameters for Available Disks

Parameter	Description	Data Type
disk_avail	Disk's availability for use: • Needs disk partition • Available for use • In use by pool <poolname>	String
disk_name	Disk device name	String

Storage Pools

Storage Pool Parameters

Parameter	Description	Note/Data Type
Available	GB's of storage available	String
used	Percentage of space used; e.g., "5"	Integer String
Compression	Compression feature "on" or "off"	String
dedup	Deduplication feature "on" of "off"	String
dedupfactor	Deduplication factor	Float String
free_numeric	Numeric percentage free space; e.g., "99.5"	Float String
free-space	Free space in GB; e.g., "99.5G"	String
no_disks	Number of disks in the storage pool; e.g. "5"	Integer String
optimizations	Optimizations display string	String
pct_used	Percentage of space used; "5%"	String percentage
pool_name	Name of the storage pool; e.g., "naspool1"	String
provisioning	Type of provisioning: "Thin" or "Thick"	String
status	Status of the pool; e.g., "ONLINE"	String
time_updated	Last date/time pool stats updated	Date String
total_numeric	Total space; "99.5"	Integer String
total_space	Total space in GB; e.g., "99.5G", "120T"	String
used_numeric	Used space numeric; e.g., "10.5"	Float String
used_space	Used space in GB; e.g., "10.5G"	String
pool type	Type of pool; "filesystem" or "blockdevice"	String
force	Set to true "1" to force pool creation	Boolean String
disk devices	Array of disk devices; e.g., "/dev/sdb", "/dev/xvf10"	String
force cache	Set to true "1" to force cache device creation	Boolean String
raid_abbr	RAID level: "0" for no RAID/RAID0, "1": for "mirror", "5" for "raidz", "6" for "raidz2", or "7" for "raidz3"	String
forced spare	Set to true "1" to force space device addition to pool	Boolean String

Volumes

Storage Volume Parameters

Parameter	Description	Note / Data Type
Available	Available volume space; “97.9” (GB)	Float String
Snapshots	Volume space consumed by snapshots; “25.2” (GB)	Float String
Used	Used volume space numeric; “136” (GB)	Float String
cbSnapshotEnabled	Enable snapshots. Set to “1” if snapshots are to be enabled (otherwise do not provide a value and default is disabled)	Boolean String
compression	Enable compression. Set to “1” if compression is to be enabled (otherwise do not provide a value and default is disabled)	Boolean String
dailysnaps	Daily maximum number of scheduled snapshots to retain before pruning snapshots; “10”.	Integer String
dedup	Enable deduplication. Set to “1” if deduplication is to be enabled (otherwise do not provide a value and default is disabled)	Boolean String
enable_snapshots	Enable scheduled volume snapshots: “1” to enable, “0” to disable.	Boolean String
free_numeric	Numeric free space; e.g., “97.9” (GB)	Float String
free_space	Free space in GB; e.g., “97.9G”	String
hourlysnaps	Hourly maximum number of scheduled snapshots to keep before pruning snapshots	Integer String
nfs_export	Set true to export via NFS using default export	Boolean String
optmizations	List of optimizations	String
pct_used	Percentage of space used; “5%”	String
pool	Storage pool name associated with this volume	String
provisioning	Allocation space type; “Thin” or “Thick”	String
	Thin: Thin Provision. Dynamically allocate space as it is needed.	
	Thick: Thick Provision. Pre-allocate space from storage pool.	
reserve_space	Reserved space size (number). Amount of space, in “reserve_units”, to reserve for Thick provisioned volume.	String
reserve_units	Reserved space size unit. ie. M(mega), G (giga), T (tera)	String
schedule_name	Snapshot schedule name to be associated with this volume; e.g., “Business”	String
status	Volume status; e.g., “ONLINE”, “DEGRADED”	String
time_updated	Last time record updated	Date String
total_numeric	Total space; “99.5”	Float String
total_space	Total space in GB; e.g., “99.5G”, “120T”	String
used_numeric	Used space numeric; e.g., “10.5”	Float String
used_space	Used space in GB; e.g., “10.5G”	String
usedbydataset	Space used by the dataset; e.g., “106K”, “10T”	String
usedbysnapshots	Space used by snapshots; ., “10GB”, “2.5T”	String
vol_name	Name of the volume; e.g., “myvolumename”	String

vol_path	Mount path for volume; e.g. “/mypool/myvol” (filesystem), “/dev/zvol/apool1/avol" (blockdevice)	String
vol_type	Possible values:	String
	“filesystem”: Filesystem (NFS, CIFS)	
	“blockdevice”: Block device (iSCSI LUN)	
weeklysnaps	Number of weekly snapshots to retain before pruning	Integer String
shareISCSI	True “1” if shared via iSCSI	Boolean String
shareCIFS	True “1” if share via CIFS	Boolean String
exportNFS	True “1” if share via NFS	Boolean String
enable snapshots	True “1” to enable scheduled snapshots	Boolean String
schedule name	Name of the schedule to use	String

Snapshot

Snapshot Parameters

Parameter	Description	Note / Data Type
creation	When snapshot was created	Date String
datesort	Internal use only	Date String
pool_name	Storage pool name	String
refer	Amount of data referenced in snapshot; e.g., “10.5G”	String
snapshot_name	Name of the snapshot	String
volume_name	Volume name associated with Snapshot	String

iSCSI Target

iSCSI Target Parameters

Parameter	Definition	Note / Data Type
incominguser	Username/password	String
initiators	Comma-separate list of hosts or IP's of allowed initiators	String
lunXX	Lun name	String
lundevice	Device or file path associated with the LUN	String
password	Password for access	String
target	iSCSI target name	String
username	Username for password-protected access	String
writecache	"1" to enable write-caching, "0" to disable	Boolean Strings

Disk

Disk Parameters

Parameter	Definition	Data Type
disk_name	Name of disc device.	String
disk_size	Size of disk; e.g., “100GB”	String
partitions	Number of partitions	Integer String.
poolmember	Message; name of storage pool or needs partitioning	String

Partition

Partition Parameters

Parameter	Definition	Data Type
partition_all	Command to partition all unpartitioned devices	String
add_partition	Command to partition a specific device	String
remove_partition	Command to remove partition from specific device	String
disk_name	Name of disc device.	String

Performance Monitoring

Performance Monitoring Parameters

Parameter	Definition
arc_hitpercent	Percentage of ARC cache hits
arc_hits	Number of ARC cache hits
arc_miss	Number of ARC cache misses
arc_read	Number of ARC cache reads
arc_size	Size of ARC cache (bytes)
arc_target	Maximum size of ARC cache (bytes)
cpu	Percentage of CPU consumed
io_diskreads	Disk reads in MB/sec
io_diskwrites	Disk writes in MB/sec
io_netreads	Network reads in MB/sec
io_netwrites	Network writes in MB/sec
iops_cifs	Unavailable
iops_iscsi	Unavailable
iops_nfs	Unavailable
latency_cifs	Unavailable
latency_iscsi	Unavailable
latency_nfs	Unavailable
time	Date/time last reading was taken

Overview

Dashboard Display Bar Chart Return Parameters

Parameter	Definition
storage_data	Percentage free/used
storage_name	"6.3T Free\n(63%)"
memory_data	Percent free/used.
memory_name	"2.4 GB\nCache Used (xx%)

Network Stats

Network Statistics Parameters

Parameter	Definition
readMB	MB/sec read from network
writeMB	MB/sec written to network

Methods

List of API Operations

login – log into **SoftNAS Cloud®** and create an active API session
logout – log out of **SoftNAS Cloud®** and deactivate the API session
resetsessiontimer – for long-running commands, keeps the session from automatically logging out
licenseinfo – returns the current license information
licenseactivate – activate a license key for use with **SoftNAS Cloud®**
newlicense – install a new license key for **SoftNAS Cloud®**
internallicense – force **SoftNAS Cloud®** to use its built-in, default license
ackagreement – acknowledge the license agreement (to enable use of the product)
checkupdate – check to see if new software updates are available
executeupdate – execute and apply software updates
statusupdate – return status of update that is in-progress (started by executeupdate)
availabledisks – returns list of available disk devices
pools – lists available storage pools
pooldetails – list a storage pool's detailed attributes
createpool – creates a storage pool
deletepool – delete a storage pool
expandpool – expand a storage pool with additional devices
readcache – add a read cache device to a storage pool
writelog – add a write log cache device to a storage pool
addspare – add a spare device to a storage pool
volumes – list available storage volumes
createvolume – create a new volume and optionally share it via NFS, CIFS or iSCSI
editvolume – edit a volume's options
deletevolume – delete a volume
schedulelist – list available snapshot schedules
schedulecommand – issue a snapshot schedule control command
snapshotlist – list available snapshots
snapcommand – issue a snapshot control command
iscsitargetlist – list of available iSCSI targets
iscsicommand – issue an iSCSI control command
diskdevices – list of available disk devices and their status
diskmgmt – issue a disk management command
parted_command – issue a disk partitioning command
perfmon – get performance monitoring status information
overview – get NAS overview status information
netstats – get network performance status information
diskstats – get disk performance status information

Standard Request Structure

Each request to the **SoftNAS Cloud®** API is sent as a **REST** call over HTTP.

The requests will typically require a username, password, and potentially one or more of the following optional parameters:

- Session ID
- Base URL
- Pretty Print

For parameter definitions, consult section **Parameters**.

Standard Response Structure

Each response from the **SoftNAS Cloud®** includes a .json-formatted javascript object with the following values:

- Result json object: json-formatted response object.
- Session ID
- Success

See section **Parameters** for more information about the listed properties of the standard response structure.

Login

Description

POST a login request to the **SoftNAS Cloud® API**, creating an active session with the **SoftNAS Cloud® Admin Server.**

```
POST login username password
```

Request

Login is based on the **Standard Request Structure**.

For specific parameter definitions, see section **Parameters**

Parameter	Required?
username	Y
password	Y
Session ID	N
Base URL	N
pretty_print	N

Example

Login using PHP.

```
<?php

    $mypath = getcwd();

    $mypath = preg_replace('/\\\\/', '/', $mypath);

    $rand = rand(1, 15000);

    $cookie_file_path = "$mypath/cookies/cookie$rand.txt";

    $data = array(

     'username' => 'softnas',

     'password' => 'Pass4W0rd'

     );

    $ch = curl_init();

    curl_setopt($ch, CURLOPT_URL, "https://example.com/softnas/
login.php");

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
```

```
    curl_setopt($ch, CURLOPT_POST, true);

    curl_setopt($ch, CURLOPT_POSTFIELDS, $data);

    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);

    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE );

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE );

    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);   // Cookie
management.

    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);

    curl_setopt($ch, CURLOPT_COOKIESESSION, TRUE);


    $output = curl_exec($ch);

    $info = curl_getinfo($ch);

    curl_close($ch);

    echo $info;
```

Return Values

Standard Response Structure

Example Output

```
{
"result": {},
"session_id": 8062,
"success": true
}
```

Logout

Description

Log out of **SoftNAS Cloud®** and deactivate the API session.

```
GET logout
```

Request

Based on **Standard Request Structure**

Example

```
<?php

    $mypath = getcwd();

    $mypath = preg_replace('/\\\\/', '/', $mypath);

    $cookie_file_path = "$mypath/cookies/cookie.txt";

    $ch = curl_init();

    curl_setopt($ch, CURLOPT_URL, "https://example.com/softnas/
logout.php");

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);

    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE );

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE );

    curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);   // Cookie
management.

    curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);

    curl_setopt($ch, CURLOPT_COOKIESESSION, TRUE);

    $output = curl_exec($ch);

    $info = curl_getinfo($ch);

    curl_close($ch);

    echo $info;

?>
```

Return Values

Consult section **Standard Response Structure**.

Example

```
{
    "result": {},
    "session_id": 8062,
    "success": true
}
```

resetsessiontimer

Description

Keep a session active beyond the default timeout value of 30 minutes.

```
POST resetsessiontimer
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example

```
{
"result": {
"msg": "",
"records": null,
"success": true,
"total": 0
},
"session_id": 29660,
"success": true
}
```

licenseinfo

Description

Return the current license information.

```
POST licenseinfo
```

Request

Consult section **Standard Request Structure**,

Response

The following properties are returned in addition to the **Standard Response Structure**

Property	Description	Note/Data Type
actual-storage-GB	Actual storage in use, measured in gigabytes.	String
actual-expiration	Expiration date of license including grace period.	Date String
capabilities	Internal use only.	String
capacityGB	Licensed maximum capacity in gigabytes.	String
currentkey	License key.	String
Msg	Message appropriate to the request, based on success or failure.	String
expiration	Expiration date of the license.	Date String
gracedays	Number of grace days following license expiration that product continues to operate.	Integer String
graceperiod	If a **SoftNAS Cloud®** license expires in a production environment, it will enter the grace period. “1” means grace period is in effect.	Boolean String
graceremaining	Amount of days remaining in the grace period, before the grace period expires and **SoftNAS Cloud®** functionality is impacted.	
hardware_id	The hardware ID used for license activation. On AWS, the instance ID; otherwise, IP address of admin interface.	String
ipmode	IP address mode, how the default IP address used for licensing was assigned “STATIC” or “DHCP”	String
is_activated	Reports if the license is active or not. True or False.	Boolean String
is_perpetual	Reports if the license is a perpetual license or not. True or false.	Boolean String
is_subscription	Reports of the license type is subscription-based (monthly or annual).	Boolean String
istrial	Reports if the product is currently running in trial mode.	Boolean String
license-version	Licensed software version of **SoftNAS Cloud®**.	String
licensetype	Reports the license type: monthly, annual, subscription, trial etc.	String
maint_expiration	Reports the date at which the maintenance package expires.	Date String

maint_expired	Reports if the status of the maintenance package is expired. True or False.	Boolean String
model	The license model: "key" (licensed by key) or "utility". (Licensed by the hour).	String
platform	Platform on which **SoftNAS Cloud®** is running, i.e., Amazon Web Services, Microsoft Azure, VMWare vSphere, VMware vCloud Air	String
product-id	**SoftNAS Cloud®** product ID	String
producttype	**SoftNAS Cloud®** product type.	String
regname	Registered name – customer name associated with an activated license key.	String
sig	internal use only.	
status	Reports of the current license is valid. Possible values: valid license or an error message	String
storage-capacity-GB	Reports total storage capacity applicable to the current license in gigabytes.	String
today	Reports today's date.	Date String
totalstorageGB	Total storage consumed currently by storage pools.	String
valid	True if the license is valid.	Boolean String

Example


```
{
 "result": {
      "msg": "",
        "records": {
            "actual-storage-GB": 0,
            "actual_expiration": "License does not expire",
            "capabilities": "2:1:100:00/00/0000:2:0",
            "capacityGB": "100 GB",
            "currentkey": "Built-in License",
            "errMsg": "Valid License",
            "expiration": "License does not expire",
            "gracedays": "2",
            "graceperiod": "0",
            "graceremaining": "2",
            "hardware_id": "i-0b06fe44",
            "ipmode": "DHCP",
            "is_activated": false,
            "is_perpetual": false,
            "is_subscription": true,
            "istrial": "0",
            "license-version": "1",
            "licensetype": "Subscription",
            "maint_expiration": "No maintenance expiration",
            "maint_expired": false,
            "model": "key",
            "platform": "amazon",
            "product-id": "2",
            "producttype": "SoftNAS&trade; Essentials",
            "regname": "Unregistered, Built-in License",
            "sig": "556B7E03",
```

```
            "status": "Valid License",
            "storage-capacity-GB": "100",
            "today": "12/25/2013",
            "totalStorageGB": "0 GB",
            "valid": true
        },
        "success": true,
        "total": 31
    },
    "session_id": 29660,
    "success": true
}
```

licenseactivate

Description

Activate a license key for use with **SoftNAS Cloud®**.

Note: **SoftNAS Cloud®** does not require license keys or activation and is recommended product for on-demand, dynamic use on AWS.

```
POST
licenseactivate
```

Request

The following parameters are requested in addition to the **Standard Request Structure**,

For more information, consult the **Parameters** section.

Parameter	Required?
currentkey	Y
regname	Y
hardware_id	Y

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {

 "msg": "Activation proxy: Activation attempt failed. Details: This
license key has been previously activated and cannot be activated
again.<br><br>Please obtain a new license key or contact
Support for activation assistance if you have moved or reinstalled.",
        "success": false
    },
    "session_id": 11255,
    "success": true
}
```

newlicense

Description

Install a new **SoftNAS Cloud®** license key.

```
POST newlicense
```

Request

Consult section **Standard Request Structure**,

Response

The following properties are returned in addition to the **Standard Response Structure**

Property	Description	Note/Data Type
actual-storage-GB	Actual storage in use, measured in gigabytes.	String
actual-expiration	Expiration date of license including grace period.	Date String
capabilities	Internal use only.	String
capacityGB	Licensed maximum capacity in gigabytes.	String
currentkey	License key.	String
Msg	Message appropriate to the request, based on success or failure.	String
expiration	Expiration date of the license.	Date String
gracedays	Number of grace days following license expiration that product continues to operate.	Integer String
graceperiod	If a **SoftNAS Cloud®** license expires in a production environment, it will enter the grace period. “1” means grace period is in effect.	Boolean String
graceremaining	Amount of days remaining in the grace period, before the grace period expires and **SoftNAS Cloud®** functionality is impacted.	
hardware_id	The hardware ID used for license activation. On AWS, the instance ID; otherwise, IP address of admin interface.	String
ipmode	IP address mode, how the default IP address used for licensing was assigned “STATIC” or “DHCP”	String
is_activated	Reports if the license is active or not. True or False.	Boolean String
is_perpetual	Reports if the license is a perpetual license or not. True or false.	Boolean String
is_subscription	Reports of the license type is subscription-based (monthly or annual).	Boolean String
istrial	Reports if the product is currently running in trial mode.	Boolean String
license-version	Licensed software version of **SoftNAS Cloud®**.	String
licensetype	Reports the license type: monthly, annual, subscription, trial etc.	String
maint_expiration	Reports the date at which the maintenance package expires.	Date String

maint_expired	Reports if the status of the maintenance package is expired. True or False.	Boolean String
model	The license model: “key” (licensed by key) or “utility”. (Licensed by the hour).	String
platform	Platform on which **SoftNAS Cloud®** is running, i.e., Amazon Web Services, Microsoft Azure, VMWare vSphere, VMware vCloud Air	String
product-id	**SoftNAS Cloud®** product ID	String
producttype	**SoftNAS Cloud®** product type.	String
regname	Registered name – customer name associated with an activated license key.	String
sig	internal use only.	
status	Reports of the current license is valid. Possible values: valid license or an error message	String
storage-capacity-GB	Reports total storage capacity applicable to the current license in gigabytes.	String
today	Reports today's date.	Date String
totalstorageGB	Total storage consumed currently by storage pools.	String
valid	True if the license is valid.	Boolean String

Example


```
{
    "result": {
        "data": {
            "actual-storage-GB": 29.8,
            "actual_expiration": "01/26/2014",
            "capabilities": "1:1:2048:01/26/2014:2:1",
            "capacityGB": "2,048 GB",
            "currentkey": "CEAASA-BESNJA-8MEED6-AHAZZN-XHWB8X-A2NUK3",
            "errMsg": "Valid License",
            "expiration": "01/28/2014",
            "gracedays": "2",
            "graceperiod": "0",
            "graceremaining": "2",
            "hardware_id": "i-0b06fe44",
            "hwlock": "",
            "is_activated": true,
            "is_perpetual": false,
            "is_subscription": true,
            "istrial": "1",
            "license-version": "1",
            "licensetype": "TRIAL",
            "maint_expiration": "01/28/2014",
            "maint_expired": false,
            "model": "key",
            "product-id": "1",
            "producttype": "SoftNAS&trade; Professional Edition",
            "regname": "RBLLC",
            "sig": "1F842C0F",
            "status": "Valid License",
            "storage-capacity-GB": "2048",
```

```
            "today": "01/12/2014",
            "totalStorageGB": "30 GB",
            "valid": true
        },
        "msg": "",
        "success": true,
        "total": 30
    },
    "session_id": 11255,
    "success": true
}
```

internallicense

Description

Instruct **SoftNAS Cloud®** to use the default built-in license.

```
POST internallicense
```

Request

Consult section **Standard Request Structure**,

Response

The following properties are returned in addition to the **Standard Response Structure**.

Property	Description	Note/Data Type
actual-storage-GB	Actual storage in use, measured in gigabytes.	String
actual-expiration	Expiration date of license including grace period.	Date String
capabilities	Internal use only.	String
capacityGB	Licensed maximum capacity in gigabytes.	String
currentkey	License key.	String
Msg	Message appropriate to the request, based on success or failure.	String
expiration	Expiration date of the license.	Date String
gracedays	Number of grace days following license expiration that product continues to operate.	Integer String
graceperiod	If a **SoftNAS Cloud®** license expires in a production environment, it will enter the grace period. “1” means grace period is in effect.	Boolean String
graceremaining	Amount of days remaining in the grace period, before the grace period expires and **SoftNAS Cloud®** functionality is impacted.	
hardware_id	The hardware ID used for license activation. On AWS, the instance ID; otherwise, IP address of admin interface.	String
ipmode	IP address mode, how the default IP address used for licensing was assigned “STATIC” or “DHCP”	String
is_activated	Reports if the license is active or not. True or False.	Boolean String
is_perpetual	Reports if the license is a perpetual license or not. True or false.	Boolean String
is_subscription	Reports of the license type is subscription-based (monthly or annual).	Boolean String
istrial	Reports if the product is currently running in trial mode.	Boolean String
license-version	Licensed software version of **SoftNAS Cloud®**.	String
licensetype	Reports the license type: monthly, annual, subscription, trial etc.	String
maint_expiration	Reports the date at which the maintenance package expires.	Date String
maint_expired	Reports if the status of the maintenance package is expired. True or False.	Boolean String
model	The license model: “key” (licensed by key) or “utility”. (Licensed by the hour).	String

platform	Platform on which **SoftNAS Cloud®** is running, i.e., Amazon Web Services, Microsoft Azure, VMWare vSphere, VMware vCloud Air	String
product-id	**SoftNAS Cloud®** product ID	String
producttype	**SoftNAS Cloud®** product type.	String
regname	Registered name – customer name associated with an activated license key.	String
sig	internal use only.	
status	Reports of the current license is valid. Possible values: valid license or an error message	String
storage-capacity-GB	Reports total storage capacity applicable to the current license in gigabytes.	String
today	Reports today's date.	Date String
totalstorageGB	Total storage consumed currently by storage pools.	String
valid	True if the license is valid.	Boolean String

Example


```
{
  "success": true,
  "msg": "",
  "records": {
    "valid": true,
    "errMsg": "Valid License",
    "status": "Valid License",
    "capabilities": "2:1:100:00/00/0000:2:0",
    "model": "key",
    "sig": "556B7E03",
    "today": "12/17/2013",
    "hardware_id": "i-0b06fe44",
    "product-id": "2",
    "license-version": "1",
    "storage-capacity-GB": "100",
    "actual_expiration": "License does not expire",
    "expiration": "License does not expire",
    "gracedays": "2",
    "istrial": "0",
    "graceperiod": "0",
    "graceremaining": "2",
    "is_perpetual": false,
    "is_subscription": true,
    "maint_expired": false,
    "maint_expiration": "No maintenance expiration",
    "actual-storage-GB": 0,
    "regname": "Unregistered, Built-in License",
    "licensetype": "Subscription",
    "capacityGB": "100 GB",
    "totalStorageGB": "0 GB",
    "is_activated": false,
    "currentkey": "Built-in License",
    "producttype": "SoftNAS™ Essentials",
    "ipmode": "DHCP",
    "platform": "amazon"
  },
```

```
  "total": 31
}
```

ackagreement

Description

Acknowledge the license agreement to enable use of the product.

```
POST ackagreement
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "Agreement accepted.",
        "records": {
            "msg": "Agreement accepted."
        },
        "success": true,
        "total": 1
    },
    "session_id": 20012,
    "success": true
}
```

checkupdate

Description

Check if new software updates are available.

```
POST checkupdate
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "A newer version is available",
        "records": {
            "msg": "A newer version is available",
            "newversion": "2.0.13.el6.x86_64",
            "updateavailable": true,
            "version": "2.0.9.el6.x86_64"
        },
        "success": true,
        "total": 4
    },
    "session_id": 20012,
    "success": true
}
```

executeupdate

Description

Execute and apply software updates.

```
POST executeupdate
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "Update was started successfully. Version
'2.0.13.el6.x86_64' installation is underway...",
        "records": {
            "msg": "Update was started successfully. Version
'2.0.13.el6.x86_64' installation is underway...",
            "newversion": "2.0.13.el6.x86_64"
        },
        "success": true,
        "total": 2
    },
    "session_id": 20012,
    "success": true
}
```

statusupdate

Description

Return the status on an update that is currently in progress.

Check the status of an update started by **executeupdate**.

```
POST statusupdate
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "OK. Update to version  2.0.13.el6.x86_64 completed at
Fri Jan 3 14:45:48 EST 2014",
        "records": {
            "msg": "OK. Update to version  2.0.13.el6.x86_64 completed
at  Fri Jan 3 14:45:48 EST 2014"
        },
        "success": true,
        "total": 1
    },
    "session_id": 20012,
    "success": true
}
```

availabledisks

Description

Return a list of available disk devices.

```
POST availabledisks
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "disk_avail": "Available for Use",
                "disk_name": "/dev/xvdf"
            },
            {
                "disk_avail": "Available for Use",
                "disk_name": "/dev/xvdg"
            }
        ],
        "success": true,
        "total": 2
    },
    "session_id": 20012,
    "success": true
}
```

pools

Description

List the available storage pools.

```
POST pools
```

Request

In addition to the **Standard Request Structure** the following parameters are required.

For more information, consult section **Parameters**

Parameter	Required?
start	Y
limit	Y

Response

The following parameters are returned in addition to the **Standard Response Structure**

Parameter	Description	Note/Data Type
Available	GB's of storage available	String
used	Percentage of space used; e.g., "5"	Integer String
Compression	Compression feature "on" or "off"	String
dedup	Deduplication feature "on" of "off"	String
dedupfactor	Deduplication factor	Float String
free_numeric	Numeric percentage free space; e.g., "99.5"	Float String
free-space	Free space in GB; e.g., "99.5G"	String
no_disks	Number of disks in the storage pool; e.g. "5"	Integer String
optimizations	Optimizations display String	String
pct_used	Percentage of space used; "5%"	String percentage
pool_name	Name of the storage pool; e.g., "naspool1"	String
provisioning	Type of provisioning: "Thin" or "Thick"	String
status	Status of the pool; e.g., "ONLINE"	String
time_updated	Last date/time pool stats updated	Date String
total_numeric	Total space; "99.5"	Integer String
total_space	Total space in GB; e.g., "99.5G", "120T"	String
used_numeric	Used space numeric; e.g., "10.5"	Float String
used_space	Used space in GB; e.g., "10.5G"	String
pool type	Type of pool; "filesystem" or "blockdevice"	String
force	Set to true "1" to force pool creation	Boolean String
disk devices	List of disk devices; e.g., "/dev/sdb", "/dev/xvf10"	String
force cache	Set to true "1" to force cache device creation	Boolean String
raid level	RAID level; "" empty for no RAID/RAID0, "mirror", "raidz", "raidz2", or "raidz3" for ZFS striped RAID.	String

forced spare	Set to true “1” to force space device addition to pool	Boolean String

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "Available": 9.9,
                "Used": 0,
                "compression": "off",
                "dedup": "off",
                "dedupfactor": "1.00x",
                "free_numeric": 9.9,
                "free_space": "9.9G",
                "no_disks": 5,
                "optimizations": "none",
                "pct_used": "0%",
                "pool_name": "pool1",
                "provisioning": "Thin",
                "status": "ONLINE",
                "time_updated": "Jan 12, 2014 06:55:23",
                "total_numeric": 9.9,
                "total_space": "9.9G",
                "used_numeric": 0,
                "used_space": "0.0G"
            },
            {
                "Available": 9.9,
                "Used": 0,
                "compression": "off",
                "dedup": "off",
                "dedupfactor": "1.00x",
                "free_numeric": 9.9,
                "free_space": "9.9G",
                "no_disks": 5,
                "optimizations": "none",
                "pct_used": "0%",
                "pool_name": "pool2",
                "provisioning": "Thin",
                "status": "ONLINE",
                "time_updated": "Jan 12, 2014 06:55:23",
                "total_numeric": 9.9,
                "total_space": "9.9G",
                "used_numeric": 0,
                "used_space": "0.0G"
            },
            {
                "Available": 9.9,
                "Used": 0,
                "compression": "off",
                "dedup": "off",
                "dedupfactor": "1.00x",
```

```
                "free_numeric": 9.9,
                "free_space": "9.9G",
                "no_disks": 5,
                "optimizations": "none",
                "pct_used": "0%",
                "pool_name": "pool3",
                "provisioning": "Thin",
                "status": "ONLINE",
                "time_updated": "Jan 12, 2014 06:55:23",
                "total_numeric": 9.9,
                "total_space": "9.9G",
                "used_numeric": 0,
                "used_space": "0.0G"
            }
        ],
        "success": true,
        "total": 3
    },
    "session_id": 11255,
    "success": true
}
```

pooldetails

Description

List a storage pool's detailed attributes.

```
POST pooldetails
```

Request

The following parameters are required in addition to the **Standard Request Structure**

For more information, consult section **Parameters**

Parameter	Required?
pool_name	**Y**

Response

The following parameters are returned in addition to the **Standard Response Structure**

Parameter	Description	Note/Data Type
Available	GB's of storage available	String
used	Percentage of space used; e.g., "5"	Integer String
Compression	Compression feature "on" or "off"	String
dedup	Deduplication feature "on" of "off"	String
dedupfactor	Deduplication factor	Float String
free_numeric	Numeric percentage free space; e.g., "99.5"	Float String
free-space	Free space in GB; e.g., "99.5G"	String
no_disks	Number of disks in the storage pool; e.g. "5"	Integer String
optimizations	Optimizations display string	String
pct_used	Percentage of space used; "5%"	String percentage
pool_name	Name of the storage pool; e.g., "naspool1"	String
provisioning	Type of provisioning: "Thin" or "Thick"	String
status	Status of the pool; e.g., "ONLINE"	String
time_updated	Last date/time pool stats updated	Date String
total_numeric	Total space; "99.5"	Integer String
total_space	Total space in GB; e.g., "99.5G", "120T"	String
used_numeric	Used space numeric; e.g., "10.5"	Float String
used_space	Used space in GB; e.g., "10.5G"	String
pool type	Type of pool; "filesystem" or "blockdevice"	String
force	Set to true "1" to force pool creation	Boolean String
disk devices	List of disk devices; e.g., "/dev/sdb", "/dev/xvf10"	String
force cache	Set to true "1" to force cache device creation	Boolean String
raid level	RAID level; "" empty for no RAID/RAID0, "mirror", "raidz", "raidz2", or "raidz3" for ZFS striped RAID.	String
forced spare	Set to true "1" to force space device addition to pool	Boolean String

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "Available": 9.9,
                "Used": 0,
                "compression": "off",
                "dedup": "off",
                "dedupfactor": "1.00x",
                "free_numeric": 9.9,
                "free_space": "9.9G",
                "no_disks": 5,
                "optimizations": "none",
                "pct_used": "0%",
                "pool_name": "pool1",
                "provisioning": "Thin",
                "status": "ONLINE",
                "time_updated": "Jan 12, 2014 06:55:23",
                "total_numeric": 9.9,
                "total_space": "9.9G",
                "used_numeric": 0,
                "used_space": "0.0G"
            },
            {
                "Available": 9.9,
                "Used": 0,
                "compression": "off",
                "dedup": "off",
                "dedupfactor": "1.00x",
                "free_numeric": 9.9,
                "free_space": "9.9G",
                "no_disks": 5,
                "optimizations": "none",
                "pct_used": "0%",
                "pool_name": "pool2",
                "provisioning": "Thin",
                "status": "ONLINE",
                "time_updated": "Jan 12, 2014 06:55:23",
                "total_numeric": 9.9,
                "total_space": "9.9G",
                "used_numeric": 0,
                "used_space": "0.0G"
            },
            {
                "Available": 9.9,
                "Used": 0,
                "compression": "off",
                "dedup": "off",
                "dedupfactor": "1.00x",
                "free_numeric": 9.9,
```

```
                "free_space": "9.9G",
                "no_disks": 5,
                "optimizations": "none",
                "pct_used": "0%",
                "pool_name": "pool3",
                "provisioning": "Thin",
                "status": "ONLINE",
                "time_updated": "Jan 12, 2014 06:55:23",
                "total_numeric": 9.9,
                "total_space": "9.9G",
                "used_numeric": 0,
                "used_space": "0.0G"
            }
        ],
        "success": true,
        "total": 3
    },
    "session_id": 11255,
    "success": true
}
```

createpool

Description

Create a new storage pool.

```
POST createpool
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters**

Parameter	Required?
selectedItems	Y
pool_name	Y
raid_abbr	Y
forcedCreation	N

selectedItems - this is an array of devices; e.g., {**/dev/sdb, /dev/sdc**}
raid_abbr - the RAID type:

- "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)

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "Create pool 'pool1' was successful.",
        "records": [],
        "success": true,
        "total": 0
    },
    "session_id": 27088,
    "success": true
}
```

deletepool

Description

Delete a storage pool.

```
DELETE deletepool
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters**

Parameter	Required?
Pool name	Y

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "Pool 'pool2' deleted.",
        "records": [],
        "success": true,
        "total": 0
    },
    "session_id": 11255,
    "success": true
}
```

expandpool

Description

Import deleted or foreign pools.

```
POST expandpool
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters**

Parameter	Required?
pool_name	Y
pool type	Y
force	Y

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "Import deleted pool 'pool1' was successful.",
        "records": [],
        "success": true,
        "total": 0
    },
    "session_id": 11255,
    "success": true
}
```

readcache

Description

Create cache disk devices for pool storage.

```
POST readcache
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters**

Parameter	Parameter	Required?
disk devices	disc_devices	Y
pool name	pool_name	Y
force cache	force_cache	Y

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "Read Cache pool 'pool3' was successful.",
        "records": [],
        "success": true,
        "total": 0
    },
    "session_id": 11255,
    "success": true
}
```

writelog

Description

Create log disk devices for pool storage.

```
POST writelog
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters**

Parameter	Required?
disk_devices	Y
pool_name	Y
raid_level	Y
force_cache	

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "Write log pool 'pool3' was successful.",
        "records": [],
        "success": true,
        "total": 0
    },
    "session_id": 11255,
    "success": true
}
```

addspare

Description

Add a spare disk to a storage pool.

```
POST addspare
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters**

Parameter	Required?
disk_devices	Y
pool_name	Y
forced_spare	Y

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "Add Spare pool 'pool3' was successful.",
        "records": [],
        "success": true,
        "total": 0
    },
    "session_id": 11255,
    "success": true
}
```

volumes

Description

List the available storage volumes.

```
POST volumes
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
start	Y
limit	Y

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "Available": 29.3,
                "Snapshots": null,
                "Used": 136,
                "cbSnapshotEnabled": "1",
                "compression": "off",
                "dailysnaps": 2,
                "dedup": "off",
                "free_numeric": 29.3,
                "free_space": "29.3G",
                "hourlysnaps": 5,
                "nfs_export": null,
                "optimizations": "none",
                "pct_used": "0%",
                "pool": "pool3",
                "provisioning": "Thin",
                "reserve_space": null,
                "reserve_units": null,
                "schedule_name": "Default",
                "status": "ONLINE",
                "time_updated": "Jan 12, 2014 16:34:02",
                "total_numeric": 29.8,
                "total_space": "29.8G",
```

```
                "used_numeric": 0,
                "used_space": "0.0G",
                "usedbydataset": "136K",
                "usedbysnapshots": "0",
                "vol_name": "vol1",
                "vol_path": "/dev/zvol/pool3/vol1",
                "vol_type": "filesystem",
                "weeklysnaps": 0
            },
            {
                "Available": 29.3,
                "Snapshots": null,
                "Used": 136,
                "cbSnapshotEnabled": "1",
                "compression": "off",
                "dailysnaps": 2,
                "dedup": "off",
                "free_numeric": 29.3,
                "free_space": "29.3G",
                "hourlysnaps": 5,
                "nfs_export": null,
                "optimizations": "none",
                "pct_used": "0%",
                "pool": "pool3",
                "provisioning": "Thin",
                "reserve_space": null,
                "reserve_units": null,
                "schedule_name": "Default",
                "status": "ONLINE",
                "time_updated": "Jan 12, 2014 16:34:02",
                "total_numeric": 29.8,
                "total_space": "29.8G",
                "used_numeric": 0,
                "used_space": "0.0G",
                "usedbydataset": "136K",
                "usedbysnapshots": "0",
                "vol_name": "vol2",
                "vol_path": "/dev/zvol/pool3/vol2",
                "vol_type": "filesystem",
                "weeklysnaps": 0
            }
        ],
        "success": true,
        "total": 2
    },
    "session_id": 11255,
    "success": true
}
```

createvolume

Description

Create a volume.

```
POST createvolume
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
vol_name	Y
pool	Y
vol_type	Y
provisioning	Y
reserve_space	Y
reserve_units	Y
compression	Y
dedup	Y
shareISCSI	Y
shareCIFS	Y
exportNFS	Y
enable_snapshots	Y
schedule_name	Y
hourlysnaps	Y
dailysnaps	Y
weeklysnaps	Y

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "Volume 'volume5' created.",
        "records": [],
        "success": true,
        "total": 0
```

```
    },
    "session_id": 11255,
    "success": true
}
```


Example 2


```
{
    "result": {
        "msg": "Volume 'volume6' created.",
        "records": [],
        "success": true,
        "total": 0
    },
    "session_id": 11255,
    "success": true
}
```

editvolume

Description

Edit a volume.

```
POST editvolume
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters**

Parameter	Required?
vol_name	Y
pool	Y
vol_type	Y
provisioning	Y
reserve_space	Y
reserve_units	Y
compression	Y
dedup	Y
shareISCSI	Y
shareCIFS	Y
exportNFS	Y
enable_snapshots	Y
schedule_name	Y
hourlysnaps	Y
dailysnaps	Y
weeklysnaps	Y

Response

Consult section **Standard Response Structure.**

deletevolume

Description

Delete a volume.

```
POST deletevolume
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
vol_name	Y
pool	Y

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "Volume 'volume6' deleted.",
        "records": [],
        "success": true,
        "total": 0
    },
    "session_id": 11255,
    "success": true
}
```

schedulelist

Description

List the available schedules.

```
POST schedulelist
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "cbDailyFri": "1",
                "cbDailyMon": "1",
                "cbDailySat": "0",
                "cbDailySun": "0",
                "cbDailyThu": "1",
                "cbDailyTue": "1",
                "cbDailyWed": "1",
                "daily_schedule": "Mon, Tue, Wed, Thu, Fri",
                "hour0": "0",
                "hour1": "0",
                "hour10": "0",
                "hour11": "0",
                "hour12": "1",
                "hour13": "0",
                "hour14": "0",
                "hour15": "1",
                "hour16": "0",
                "hour17": "0",
                "hour18": "1",
                "hour19": "0",
                "hour2": "0",
                "hour20": "0",
                "hour21": "0",
                "hour22": "0",
                "hour23": "0",
                "hour3": "0",
                "hour4": "0",
                "hour5": "0",
                "hour6": "1",
                "hour7": "0",
```

```
                "hour8": "0",
                "hour9": "1",
                "hourly_map":
"6&amp;nbsp;&amp;nbsp;9&amp;nbsp;&amp;nbsp;12&amp;nbsp;&amp;nbsp;15&amp;nbsp;&amp;nbsp;18&amp;nbsp;&amp;nbsp;",
                "schedule_name": "Default",
                "weekly_schedule": "Sun",
                "weekly_val": "7"
            },
            {
                "cbDailyFri": "1",
                "cbDailyMon": "1",
                "cbDailySat": "1",
                "cbDailySun": "1",
                "cbDailyThu": "1",
                "cbDailyTue": "1",
                "cbDailyWed": "1",
                "daily_schedule": "Mon, Tue, Wed, Thu, Fri, Sat, Sun",
                "hour0": "1",
                "hour1": "0",
                "hour10": "1",
                "hour11": "0",
                "hour12": "1",
                "hour13": "0",
                "hour14": "1",
                "hour15": "0",
                "hour16": "1",
                "hour17": "0",
                "hour18": "1",
                "hour19": "0",
                "hour2": "1",
                "hour20": "1",
                "hour21": "0",
                "hour22": "1",
                "hour23": "0",
                "hour3": "0",
                "hour4": "1",
                "hour5": "0",
                "hour6": "1",
                "hour7": "0",
                "hour8": "1",
                "hour9": "0",
                "hourly_map":

"0&amp;nbsp;&amp;nbsp;2&amp;nbsp;&amp;nbsp;4&amp;nbsp;&amp;nbsp;6&amp;nbsp;&amp;nbsp;8&amp;nbsp;&amp;nbsp;10&amp;nbsp;&amp;nbsp;12&amp;nbsp;&amp;nbsp;
14&amp;nbsp;&amp;nbsp;16&amp;nbsp;&amp;nbsp;18&amp;nbsp;&amp;nbsp;20&amp;nbsp;&amp;nbsp;22&amp;nbsp;&amp;nbsp;",
                "schedule_name": "24x7",
                "weekly_schedule": "Sun",
                "weekly_val": "7"
            },
            {
                "cbDailyFri": "1",
                "cbDailyMon": "1",
                "cbDailySat": "1",
```

```
                "cbDailySun": "1",
                "cbDailyThu": "1",
                "cbDailyTue": "1",
                "cbDailyWed": "1",
                "daily_schedule": "Mon, Tue, Wed, Thu, Fri, Sat, Sun",
                "hour0": "1",
                "hour1": "1",
                "hour10": "1",
                "hour11": "1",
                "hour12": "1",
                "hour13": "1",
                "hour14": "1",
                "hour15": "1",
                "hour16": "1",
                "hour17": "1",
                "hour18": "1",
                "hour19": "1",
                "hour2": "1",
                "hour20": "1",
                "hour21": "1",
                "hour22": "1",
                "hour23": "1",
                "hour3": "1",
                "hour4": "1",
                "hour5": "1",
                "hour6": "1",
                "hour7": "1",
                "hour8": "1",
                "hour9": "1",
                "hourly_map":

"0&amp;nbsp;&amp;nbsp;1&amp;nbsp;&amp;nbsp;2&amp;nbsp;&amp;nbsp;3&amp;nbsp;&amp;nbsp;4&amp;nbsp;&amp;nbsp;5&amp;nbsp;&amp;nbsp;6&amp;nbsp;&amp;nbsp;7&amp;n
8&amp;nbsp;&amp;nbsp;9&amp;nbsp;&amp;nbsp;10&amp;nbsp;&amp;nbsp;11&amp;nbsp;&amp;nbsp;12&amp;nbsp;&amp;nbsp;13&amp;nbsp;&amp;nbsp;14&amp;nbsp;&amp;nbsp
16&amp;nbsp;&amp;nbsp;17&amp;nbsp;&amp;nbsp;18&amp;nbsp;&amp;nbsp;19&amp;nbsp;&amp;nbsp;20&amp;nbsp;&amp;nbsp;21&amp;nbsp;&amp;nbsp;22&amp;nbsp;&amp;nb
                "schedule_name": "MaximumSnapshots",
                "weekly_schedule": "Sun",
                "weekly_val": "7"
            },
            {
                "cbDailyFri": "1",
                "cbDailyMon": "1",
                "cbDailySat": "0",
                "cbDailySun": "0",
                "cbDailyThu": "1",
                "cbDailyTue": "1",
                "cbDailyWed": "1",
                "daily_schedule": "Mon, Tue, Wed, Thu, Fri",
                "hour0": "1",
                "hour1": "0",
                "hour10": "0",
                "hour11": "0",
                "hour12": "1",
                "hour13": "0",
```

```
                "hour14": "0",
                "hour15": "1",
                "hour16": "0",
                "hour17": "0",
                "hour18": "1",
                "hour19": "0",
                "hour2": "0",
                "hour20": "0",
                "hour21": "1",
                "hour22": "0",
                "hour23": "0",
                "hour3": "0",
                "hour4": "0",
                "hour5": "0",
                "hour6": "1",
                "hour7": "0",
                "hour8": "0",
                "hour9": "1",
                "hourly_map":

"0&amp;nbsp;&amp;nbsp;6&amp;nbsp;&amp;nbsp;9&amp;nbsp;&amp;nbsp;12&amp;nbsp;&amp;nbsp;15&amp;nbsp;&amp;nbsp;18&amp;nbsp;&amp;nbsp;21&amp;nbsp;&amp;nbsp
                "schedule_name": "Business",
                "weekly_schedule": "Sun",
                "weekly_val": "7"
            }
        ],
        "success": true,
        "total": 4
    },
    "session_id": 11255,
    "success": true
}
```

schedulecommand

Description

Issue a schedule control command.

```
POST schedulecommand
```

Request

Consult section **Standard Request Structure**.

Response

Consult section **Standard Response Structure.**

snapshotlist

Description

List the available snapshots.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
pool_name	Y
start	Y
limit	Y

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "creation": "Sun Jan 12 19:00 2014 ",
                "datesort": 1389574800,
                "pool_name": "pool3",
                "refer": "136K",
                "snapshot_name": "hourly1",
                "volume_name": "vol1"
            }
        ],
        "success": true,
        "total": 1
    },
    "session_id": 11255,
    "success": true
}
```

snapcommand

Description

Issue a snapshot control command.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
pool_name	Y
start	Y
limit	Y

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "creation": "Sun Jan 12 19:00 2014 ",
                "datesort": 1389574800,
                "pool_name": "pool3",
                "refer": "136K",
                "snapshot_name": "hourly1",
                "volume_name": "vol1"
            }
        ],
        "success": true,
        "total": 1
    },
    "session_id": 11255,
    "success": true
}
```

iscsitargetlist

Description

List the available iSCSI targets.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
start	Y
limit	Y
returnGlobals	Y

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "incominguser": "No authentication required",
                "initiators": "Unrestricted network access",
                "lun1": "/pool3/LUN_vol3/lundata.dat",
                "lundevice": "/pool3/LUN_vol3/lundata.dat",
                "password": "",
                "target": "iqn.2013-02.com.softnas:storage.target1",
                "username": "",
                "writecache": "on"
            }
        ],
        "success": true,
        "total": 2
    },
    "session_id": 11255,
    "success": true
}
```

iscsicommand

Description

Issue an iSCSI control command.

```
POST iscsicommand
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
start	Y
limit	Y
returnGlobals	

Response

Consult section **Standard Response Structure.**

diskdevices

Description

List the available disk devices and their status

```
POST diskdevices
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "brand": "Xen Virtual Block Device (xvd)",
                "disk_name": "/dev/xvdf",
                "disk_size": "10.7GB",
                "partitions": 1,
                "poolmember": "Used in pool pool3"
            },
            {
                "brand": "Xen Virtual Block Device (xvd)",
                "disk_name": "/dev/xvdg",
                "disk_size": "10.7GB",
                "partitions": 1,
                "poolmember": "Used in pool pool3"
            },
            {
                "brand": "Xen Virtual Block Device (xvd)",
                "disk_name": "/dev/xvdi",
                "disk_size": "10.7GB",
                "partitions": 1,
                "poolmember": "Used in pool pool3"
            },
            {
                "brand": "Xen Virtual Block Device (xvd)",
                "disk_name": "/dev/xvdu",
                "disk_size": "10.7GB",
                "partitions": 1,
                "poolmember": "Used in pool pool3"
            },
            {
                "brand": "Xen Virtual Block Device (xvd)",
                "disk_name": "/dev/xvdm",
```

```
            "disk_size": "10.7GB",
            "partitions": 1,
            "poolmember": "Used in pool pool3"
        },
        {
            "brand": "Xen Virtual Block Device (xvd)",
            "disk_name": "/dev/xvdl",
            "disk_size": "10.7GB",
            "partitions": 1,
            "poolmember": "Used in pool pool3"
        },
        {
            "brand": "Xen Virtual Block Device (xvd)",
            "disk_name": "/dev/xvdj",
            "disk_size": "10.7GB",
            "partitions": 1,
            "poolmember": "Used in pool pool3"
        },
        {
            "brand": "Xen Virtual Block Device (xvd)",
            "disk_name": "/dev/xvdk",
            "disk_size": "10.7GB",
            "partitions": 0,
            "poolmember": "Used in pool pool3"
        },
        {
            "brand": "Xen Virtual Block Device (xvd)",
            "disk_name": "/dev/xvdn",
            "disk_size": "10.7GB",
            "partitions": 0,
            "poolmember": "Used in pool pool3"
        },
        {
            "brand": "Xen Virtual Block Device (xvd)",
            "disk_name": "/dev/xvdo",
            "disk_size": "10.7GB",
            "partitions": 1,
            "poolmember": "Used in pool pool3"
        },
        {
            "brand": "Xen Virtual Block Device (xvd)",
            "disk_name": "/dev/xvdr",
            "disk_size": "10.7GB",
            "partitions": 1,
            "poolmember": "Used in pool pool3"
        }
    ],
    "success": true,
    "total": 11
},
"session_id": 11255,
"success": true
}
```

diskmgmt

Description

Issue a disk management command for dynamically added disk devices (e.g., S3 Cloud Disks).

```
POST diskmgmt
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Parameter Value	Required?
command	createS3disk, deleteS3disk, deleteS3disk, getS3settings, bucketlist	Y
start	start	Y
limit	limit	Y

Command parameters

```
#  createS3disk - creates an S3 disk devices. Related arguments:
## # # 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
## # # cb_encrypted    : add it to encrypt disk

#  deleteS3disk - delete an S3 disk
## # # s3diskname      : S3 disk name; i.e (/dev/s3-0)

#  getS3settings - get list of S3 disk devices settings

#  bucketlist - get list of S3 buckets
```

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example 1 - create S3 disk device

```
$ 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
diskpassword=password123* -t

JSON Response:
     {
          "result": {
              "msg": "s3-0 created successfully.",
              "records": {
                  "msg": "s3-0 created successfully."
              },
              "success": true,
              "total": 1
          },
          "session_id": 10123,
          "success": true
}
```

Example 2 - get list of available S3 buckets

```
$ softnas-cmd diskmgmt bucketlist -t

JSON Response:
     {
          "result": {
              "msg": "",
              "records": [
                  {
                      "bucketname": "softnas-test-1"
                  },
                  {
                      "bucketname": "softnas-test-2"
                  },
                  {
                      "bucketname": "softnas-test-3"
                  },
              ],
              "success": true,
              "total": 3
          },
          "session_id": 10123,
          "success": true
```

Example 3 - Get list of S3 settings

```
$ softnas-cmd diskmgmt getS3settings -t

JSON Response:
```

```
    {
        "result": {
            "msg": "",
            "records": {
                "awsAccessKey": "XXXXXXXXXXXXXXXX",
                "awsSecretKey": "XXXXXXXXXXXXXXXX",
                "blocksize": "1M",
                "bucketroot": "softnas",
                "cachesize": "10",
                "cachethreads": "10",
                "mountroot": "/mnt",
                "nDevices": "3",
                "nDisks": "3",
                "timeout": "15"
            },
            "success": true,
            "total": 10
        },
        "session_id": 10123,
        "success": true
}
```


Example 4 - Delete S3 disk device

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

JSON Response:
    {
        "result": {
            "msg": "/dev/s3-0 deleted.",
            "records": {
                "msg": "/dev/s3-0 deleted."
            },
            "success": true,
            "total": 1
        },
        "session_id": 10123,
        "success": true
}
```

parted_command

Description

Issue a disk partitioning command.

```
POST parted_command
```

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
partition_all	N
add_partition	N
remove_partition	N
disk name	N

Response

Consult section **Standard Response Structure.**

Example

```
{
    "result": {
        "msg": "/dev/xvdr has been created successfully.",
        "records": [],
        "success": true,
        "total": 0
    },
    "session_id": 11255,
    "success": true
}
```

Notes

On AWS, EBS disk devices always show as raw devices, even after partitioning. This is a known issue and peculiar to EBS disk device partitioning. Even though the devices may display as needing to be partitioned, they can be used within a pool.

perfmon

Description

Obtain performance monitoring status information.

```
POST perfmon
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "arc_hitpercent": false,
                "arc_hits": false,
                "arc_miss": false,
                "arc_read": false,
                "arc_size": false,
                "arc_target": false,
                "cpu": 0,
                "io_diskreads": 0,
                "io_diskwrites": 0,
                "io_netreads": 0,
                "io_netwrites": 0,
                "iops_cifs": 0,
                "iops_iscsi": 0,
                "iops_nfs": 0,
                "latency_cifs": 0,
                "latency_iscsi": 0,
                "latency_nfs": 0,
                "time": "25:38"
            },
            {
                "arc_hitpercent": false,
                "arc_hits": false,
                "arc_miss": false,
                "arc_read": false,
                "arc_size": false,
                "arc_target": false,
                "cpu": false,
                "io_diskreads": false,
                "io_diskwrites": false,
                "io_netreads": false,
```

```
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
```

```
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
```

```
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
```

```
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
```

```
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
```

```
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
```

```
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
```

```
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
```

```
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
```

```
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
        "arc_hitpercent": false,
        "arc_hits": false,
        "arc_miss": false,
        "arc_read": false,
        "arc_size": false,
        "arc_target": false,
        "cpu": false,
        "io_diskreads": false,
        "io_diskwrites": false,
        "io_netreads": false,
        "io_netwrites": false,
        "iops_cifs": false,
        "iops_iscsi": false,
        "iops_nfs": false,
        "latency_cifs": false,
        "latency_iscsi": false,
        "latency_nfs": false,
        "time": " "
    },
    {
```

```
                "arc_hitpercent": false,
                "arc_hits": false,
                "arc_miss": false,
                "arc_read": false,
                "arc_size": false,
                "arc_target": false,
                "cpu": false,
                "io_diskreads": false,
                "io_diskwrites": false,
                "io_netreads": false,
                "io_netwrites": false,
                "iops_cifs": false,
                "iops_iscsi": false,
                "iops_nfs": false,
                "latency_cifs": false,
                "latency_iscsi": false,
                "latency_nfs": false,
                "time": " "
            },
            {
                "arc_hitpercent": false,
                "arc_hits": false,
                "arc_miss": false,
                "arc_read": false,
                "arc_size": false,
                "arc_target": false,
                "cpu": false,
                "io_diskreads": false,
                "io_diskwrites": false,
                "io_netreads": false,
                "io_netwrites": false,
                "iops_cifs": false,
                "iops_iscsi": false,
                "iops_nfs": false,
                "latency_cifs": false,
                "latency_iscsi": false,
                "latency_nfs": false,
                "time": " "
            }
        ],
        "success": true,
        "total": 30
    },
    "session_id": 11255,
    "success": true
}
```

overview

Description

Obtain NAS overview status information

```
POST overview
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure**.

Example


```
{
    "result": {
        "msg": "",
        "records": [
            {
                "storage_data": 99.992133285506,
                "storage_name": "9.9G Free\n(100.0%)"
            },
            {
                "storage_data": 0.0078673333949007,
                "storage_name": "820.0K Used\n(0.0%)"
            },
            {
                "memory_data": 0.23388061317604,
                "memory_name": "707.4K\nCache Used\n(0.2%)"
            },
            {
                "memory_data": 99.533330223782,
                "memory_name": "294.7M\nCache Free\n(99.5%)"
            }
        ],
        "success": true,
        "total": 4
    },
    "session_id": 11255,
    "success": true
}
```

netstats

Description

Obtain network performance status information.

```
POST netstats
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure.**

Example


```
{
    "result": {
        "msg": "",
        "records": {
            "readMB": 7.6505872938368e-05,
            "writeMB": 0.00035296546088325
        },
        "success": true,
        "total": 2
    },
    "session_id": 11255,
    "success": true
}
```

diskstats

Description

Obtain network performance status information

```
POST diskstats
```

Request

Consult section **Standard Request Structure**,

Response

Consult section **Standard Response Structure**.

Example


```
{
    "result": {
        "msg": "",
        "records": {
            "readMB": 0.00066332547169811,
            "writeMB": 0.014150943396226
        },
        "success": true,
        "total": 2
    },
    "session_id": 11255,
    "success": true
}
```

Command Reference

Command Reference Operations

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.

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 Cloud®** storage systems.

As with all **SoftNAS Cloud® API** applications, **softnas-cmd** requires initial authentication with the **SoftNAS Cloud®** server using the **login** command. Upon login, **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 Cloud®** server). Subsequent commands issued via the **softnas-cmd** CLI make use of the login session (which expires after 30 minutes).

Obtain softnas-cmd CLI program

The **softnas-cmd** CLI is located in the **SoftNAS Cloud®** install directory **/var/www/softnas/api**. It is also installed by default into **/usr/local/bin** on the **SoftNAS Cloud®** storage server for local use.

The **softnas-cmd** CLI is also available for **download from the product documentation page here**. Download the zip file, then unzip the **softnas-cmd** file. Ensure cURL is installed locally..

Installing cURL

Linux: To install cURL on CentOS/RHEL: **yum install curl** (or installer for the local version of Linux)

Download and Install cURL for all platforms

For more information on cURL, consult **the cURL website**.

Available CLI commands

This section provides basic information of the **SoftNAS Cloud®** command line interface, and also provides the list of available commands that correspond to related API calls.

Use the **Pretty Print** command line option to format the JSON String responses for better human readability (the raw JSON is printed by default).

Login

Login to **SoftNAS Cloud®**.

Parameters

Parameter	Required?
username	Y
password	Y
Session ID	N
Base URL	N
Pretty Print	N

For more information, consult section **Parameters.**

Command Options

Command	Possible Values
login	softnas
Password	Pass4wOrd

Example

```
$ softnas-cmd login softnas Pass4W0rd--base_url https://example.com/softnas --pretty_print
```

Logout

Description

Log out of **SoftNAS Cloud®** and deactivate the API session.

Parameters

Parameter	Required?
username	Y
password	Y
Session ID	N
Base URL	N
Pretty Print	N

Consult section **Standard Request Structure**,

Command Options

Command	Possible Values
login	softnas
Password	Pass4wOrd

Example 1

```
$ softnas-cmd --session_id 8062 logout
```

Example 2

```
$ softnas-cmd -s 8062 logout
```

For more information, consult section **Parameters.**

resetsessiontimer

Description

This command is used to keep session active. By default session timeout is set to 30 minutes. If you want to keep a session from expiring just call this function.

Properties

Parameter	Required?
Session ID	Y
Pretty Print	N

Consult section **Standard Request Structure**.

Command Options

Not applicable

Example 1

```
$ softnas-cmd resetsessiontimer --session_id 29660  --pretty_print
```

Example 2

```
$ softnas-cmd resetsessiontimer -s 29660 -t
```

licenseinfo

Description

This command returns the current license information.

Parameters

Parameter	Required?
Session ID	N
Pretty Print	N

Consult section **Standard Request Structure**.

Command Options

Not applicable.

Example

```
$ softnas-cmd --session_id 29660 licenseinfo
```

licenseactivate

Description

Activate a license key for use with **SoftNAS Cloud®**.

Parameters

In addition to the **Standard Request Structure**, the below parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
licensekey	Y
regname	Y
hardware_id	Y

Command Options

Command	Possible Values
license_key	the licence key; i.e. CEAASA-BESNJA-8MEED6-AHAZZN-XHWB8X-A2NUK3 RBLLC
register_name	the registered name of the license key
hardware_id	The 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
```

newlicense

Description

Install a new **SoftNAS Cloud®** license key.

Propertie

In addition to the **Standard Request Structure**, the below parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
licensekey	Y

Command Options

Command	Possible Values
license_key	the licence key; i.e. CEAASA-BESNJA-8MEED6-AHAZZN-XHWB8X-A2NUK3 RBLLC

Example

```
$ softnas-cmd newlicense CEAASA-BESNJA-8MEED6-AHAZZN-XHWB8X-A2NUK3RBLLC online --
pretty_print
```

internallicense

Description

Instruct **SoftNAS Cloud®** to use the default built-in license.

Properties

No required Parameters.

Consult section **Standard Request Structure**

Command Options

Not applicable

Example

```
$ softnas-cmd internallicense --pretty_print
```

ackagreement

Description

Acknowledge the license agreement to enable use of the product.

Properties

No required parameters.

Consult section **Standard Request Structure**.

Command Options

Not applicable.

Example

```
$ curl --cookie cookies.txt --cookie-jar cookies.txt --data
"opcode=resetsessiontimer"
https://example.com/softnas/snserver/snserv.php
```

checkupdate

Description

Check if new software updates are available.

Properties

No Required Parameters.

Consult section **Standard Request Structure**.

Command Options

Not Applicable.

Example

```
$ softnas-cmd checkupdate
```

executeupdate

Description

Execute and apply software updates.

Properties

No required parameters.

Consult section **Standard Request Structure**.

Command Options

Not applicable.

Example

```
$ softnas-cmd executeupdate
```

statusupdate

Description

Return the status on an update that is currently in progress, initially started by **executeupdate**.

Properties

No required parameters.

Consult section **Standard Request Structure**.

Command Options

Not Applicable.

Example

```
$ softnas-cmd statusupdate
```

availabledisks

Description

Return a list of available disc devices.

Properties

No required parameters

Consult section **Standard Request Structure**.

Command Options

Not Applicable.

Example

```
$ softnas-cmd availabledisks
```

pools

Description

List the available storage pools.

Properties

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
start	Y
limit	Y

COMMAND OPTIONS

Command	Possible Values
start	position to start from (used for pagination)
limit	number of items to get

Example

```
$ softnas-cmd pools 0 10 --pretty_print
```

pooldetails

Description

List a storage pool's detailed attributes.

Properties

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
pool_name	Y

Command Options

Command	Possible Values
pool_name	the pool name

Example

```
$ softnas-cmd pooldetails pool2 --pretty_print
```

poolcommand

Description

Issue a command to control a storage pool

Properties

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
pool_name	Y

Command Options

Command	Possible Values
pool_name	the pool name

Example

```
$ softnas-cmd poolcommand pool2 --pretty_print
```

createpool

Description

Create a new storage pool.

Properties

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
disk devices	Y
pool_name	Y
raid level	Y
compression	Y

Command Options

Command	Possible Values
disk_devices	The physical or virtual disks that make up the pool
pool_name	the pool name
raid_level	Raid Level used for the storage pool. Possible Values include: 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)
enable_compression	possible values: on/off

Example

```
$ softnas-cmd createpool /dev/xvdf:/dev/xvdg pool1 1 on -t
```

deletepool

Description

Delete a storage pool.

Properties

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
pool_name	Y

Command Options

Command	Possible Values
pool_name	the pool name

Example

```
$ softnas-cmd deletepool pool1 -t
```

importpool

Description

Import deleted or foreign pools.

Properties

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
pool_name	Y
pool_type	Y
force	Y

Command Options

Command	Possible Values
pool_name	the pool name
pool_type	type of pool
force	whether to force import

Example

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

readcache

Description

Delete a storage pool.

Properties

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
disk_devices	Y
pool_name	Y
force_cache	Y

Command Options

Command	Possible Values
disk_devices	list of diskdevices separated by comma ":"
pool_name	the name of the pool
force_cache	possible values: on/off

Example

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

writelog

Description

Create log disk devices for pool storage.

Properties

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
disk_devices	Y
pool_name	Y
raid_level	Y
force_cache	Y

Command Options

Command	Possible Values
disk_devices	list of diskdevices separated by comma ":"
pool_name	name of the pool
raid_level	The raid level of the writelog. Possible values: 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 -t
```

addspare

Description

Add a spare disk to a storage pool.

Properties

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
disk_devices	Y
pool_name	Y
forced_spare	Y

Command Options

Command	Possible Values
disk_devices	list of disk devices separated by comma ":"
pool_name	the pool name
force_spare	possible values: on/off

Example

```
$ softnas-cmd addspare /dev/xvdn pool3 on -t
```

volumes

Description

List the available storage volumes.

Properties

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
start	Y
limit	Y

Command Options

Command	Possible Values
start	position to start from (used for pagination)
limit	number of items to get

Example

```
$ softnas-cmd volumes 0 10 -t
```

createvolume

Description

Create a volume.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Parameter	Required?
volume name	vol_name	Y
pool name	pool	Y
volume type	vol_type	Y
provisioning	provisioning	Y
reserve space	reserve_space	Y
reserve units	reserve_units	Y
compression	compression	Y
dedup	dedup	Y
shareISCSI	shareISCSI	Y
shareCIFS	shareCIFS	Y
exportNFS	exportNFS	Y
enable snapshots	enable_snapshots	Y
schedule name	scedule_name	Y
hourlysnaps	hourlysnaps	Y
dailysnaps	dailysnaps	Y
weeklysnaps	weeklysnaps	Y

Command Options

Command	Possible Values
volume_name	name of volume
pool_name	selected pool name
volume_type	possible values include:
	• **File system:** (NFS, CFS)
	• **blockdevice:** Block Device (iSCSI LUN)

provisioning	allocation space type - Possible values include:
	• **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
ShareISCSI	enable sharing iscsi
ShareCIFS	enable share CIFS
exportNFS	enable export of NFS
enable_snapshots	Enable scheduled volume snapshot
schedule_name	Snapshot schedule name
hourlysnaps	hourly maximum number of scheduled snapshots
dailysnaps	daily maximum number of scheduled snapshots
weeklysnaps	weekly maximum number of scheduled snapshots

Example 1

```
softnas-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

Description

Edit a volume.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Parameter	Required?
volume name	vol_name	Y
pool name	pool	Y
volume type	vol_type	Y
provisioning	provisioning	Y
reserve space	reserve_space	Y
reserve units	reserve_units	Y
compression	compression	Y
dedup	dedup	Y
shareISCSI	shareISCSI	Y
shareCIFS	shareCIFS	Y
exportNFS	exportNFS	Y
enable snapshots	enable_snapshots	Y
schedule name	scedule_name	Y
hourlysnaps	hourlysnaps	Y
dailysnaps	dailysnaps	Y
weeklysnaps	weeklysnaps	Y

Command Option

Command	Possible Values
volume_name	name of volume
pool_name	selected pool name
volume_type	possible values include:
	• **File system:** (NFS, CFS)
	• **blockdevice:** Block Device (iSCSI LUN)

provisioning	allocation space type - Possible values include:
	• **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 sharing iscsi
ShareCIFS	enable share CIFS
exportNFS	enable export of NFS
enable_snapshots	Enable scheduled volume snapshot
schedule_name	Snapshot schedule name
hourlysnaps	hourly maximum number of scheduled snapshots
dailysnaps	daily maximum number of scheduled snapshots
weeklysnaps	weekly maximum number of scheduled snapshots

Example

```
$ softnas-cmd editvolume volume6 pool3
filesystem thin exportNFS=on shareCIFS=on dedup=on enable_snapshot=on
schedule_name=Default hourlysnaps=5 dailysnaps=10 weeklysnaps=0 -t
```

deletevolume

Description

Delete a volume.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
vol_name	Y
pool	Y

Command Option

Command	Possible Values
Volume name	name of the volume, ie: volume6
Pool name	name of the pool, ie: pool3

Example

```
$ softnas-cmd deletevolume volume6 pool3 -t
```

schedulelist

Description

List the available schedules.

Request

No required parameters.

Consult the section called **Standard Request Structure**.

Command Options

Not Applicable.

Example

```
$ softnas-cmd schedulelist -t
```

schedulecommand

Description

Issue a schedule control command.

Request

Consult section **Standard Request Structure**,

Example

```
$ softnas-cmd schedulecommand -t
```

snapshotlist

Description

List the available snapshots.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
pool_name	Y
start	Y
limit	Y

Command Options

Command	Possible Values
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 -t
```

snapcommand

Description

Issue a volume snapshot control command

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
pool_name	Y
start	Y
limit	Y

Command Options

Command	Possible Values
CREATE:	Create New Snapshot
Related Options:	
pool_name	pool name i.e 'pool3/vol1'
volume_name	volume name

Example 1

Create snapshot.

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

Example 2

Delete snapshots.

```
$ softnas-cmd snapcommand delete
'snapshots=pool1,volume1,snap-20150605-140647-cloned-
Jun52015-141106:pool1,volume1,snap-20150606-070607'
```

snapclone

Description

Issue a Clone Snapshot (SnapClone) control command.

Requests

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
pool_name	Y
start	Y
limit	Y

Command Options

Command	Possible Values
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 examples below.

Example

Add SnapClone.

```
$ softnas-cmd snapcommand clone pool_name=pool1 volume_name=vol1
snap_name=snaphot1
```

iscsitargetlist

Description

List the available iSCSI targets.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
start	Y
limit	Y
returnGlobals	Y

Command Options

Command	Possible Values
start	position to start from (used for pagination)
limit	number of items to get
returnGlobals	

Example

```
$ softnas-cmd iscsitargetlist  0 10 returnGlobals -t
```

iscsicommand

Description

Issue an iSCSI control command.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
start	Y
limit	Y
returnGlobals	

Command Options

Command	Possible Values
start	position to start from (used for pagination)
limit	number of items to get
returnGlobals	

Example

```
$ softnas-cmd iscsicommand  0 10 returnGlobals -t
```

diskdevices

Description

List the available disk devices and their status.

Request

No required parameters.

Consult section **Standard Request Structure**.

Command Options

Not Applicable

Example

```
$ softnas-cmd diskdevices -t
```

Y

diskmgmt

Description

Issue a disk management command.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Parameter	Required?
CreateS3disk	CreateS3disk	Y
awsAccessKey	awsAccessKey	Y
awsSecretKey	awsSecretKey	Y
s3bucket	s3bucket	Y
bucketroot	bucketroot	Y
sizeMaxValue	sizeMaxValue	Y
sizeMaxUnits	sizeMaxUnits	Y
diskpassword	diskpassword	Y
cb_encrypted	cb_encrypted	Y
cb_readahead	cb_readahead	Y
deleteS3disk	deleteS3disk	Y
s3diskname	s3diskname	Y
getS3settings	getS3settings	Y
bucketlist	bucketlist	Y

Command Options

Command	Possible Values
CreateS3disk	Create an 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
cb_encrypted	add it to encrypt disk
cb_readahead	
deleteS3disk	delete s3 disk
s3diskname	s3 disk name i.e (/dev/s3-0)
getS3settings	get list of s3 setting stored in config file
bucketlist	get list of aws s3 buckets

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 sizeMaxVal
ue=500 sizeMaxUnits=GB diskpassword=password123*
```

Example 2

Get a list of available buckets.

```
$ softnas-cmd diskmgmt bucketlist
```

Example 3

Get list of s3 settings stored in a config file.

```
$ softnas-cmd diskmgmt getS3settings
```

Example 4

Delete S3 Disk.

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

parted_command

Description

Issue a disk partitioning command.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
partition_all	N
add_partition	N
remove_partition	N
disk name	N

Command Options

Command	Possible Values
partition_all	partion all disks not currently partioned
add_partition	adds a partition to selected disk
remove_partition	removes a partition from selected disk
disk name	selected disk device game

Example

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

snaprepcommand

Description

Issue a SnapReplicate command.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters**.

Parameter	Required?
snapverify	Y
remotenode	Y
userid	Y
password	Y
initsnapreplicate	Y
type	Y
keyexchange	Y
keyfile	Y
pubEncoded	Y
serverEncoded	Y
serverIP	Y
snapreplicatestatus	Y
start	Y
limit	Y
snapreplog	Y
snapreplicatetasks	Y
forcesync	Y
replicatenow	Y
activate	Y
deactivate	Y
takeover	Y
giveback	Y
deletereplication	Y
getsettings	Y
modifysettings	Y
transportcmd	Y

transportflags	Y
cipherspec	Y
compressenabled	Y
throttleenabled	Y
throttlelimit	Y
throttleunits	Y

Command Options

Command	Possible Values
snapverify	verify the remote node.
Related Options:	
remotenode	ip adress of remote node
userid	username for remote node
password	password of remote node
initsnapreplicate	begin initial replicate with remote node
type	type of remote node possible values source or target
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, base 64 encoded
serverEncoded	RSA public host key content file, base 64 encoded (/etc/ssh/ssh_host_rsa_key.pub)
serverIP	server node ip (sender)
snapreplicatestatus	get the current status of replication
start	position to start from (used for pagination)
limit	number of items to get
snapreplog	get snapreplicate log
snapreplicatetasks	get snapreplicate tasks
forcesync	forces full synchronization between nodes
replicatenow	replicates changed content/files
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
Related Options:	
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	These are the possible size unit values: • b => bits/sec • k => Kbits/sec • m => Mbits/sec • B => Bytes/sec • K => Kbytes/sec • M => MBytes/sec

Example 1

Initiate SnapReplicate.

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

Example 2

SnapReplicate - verify remote node

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

Example 3

SnapReplicate - get snapreplicate status.

```
$ softnas-cmd snaprepcommand snapreplicatestatus start=0 limit=10 -t
```

Example 4

SnapReplicate - get snapreplicate logs.

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

Example 5

Delete SnapReplication.

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

Example 6

SnapReplicate - get replication settings.

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

Example 7

SnapReplicate - force synchronization between 2 nodes.

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

Example 8

SnapReplicate - start replication cycle.

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

Example 9

Activate replication.

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

Example 10

Deactivate Replication.

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

Example 11

SnapReplicate - Change role to primary node.

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

Example 12

SnapReplicate - Change role to secondary node.

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

Example 13

SnapReplicate - 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
```

perfmon

Description

Obtain performance monitoring status information.

IMPORTANT: This command gives 30 records. The command should be run several times in order to gain valuable information over a set interval. If attempting to get a baseline of performance for 20 minutes of runtime (as an example), you should run the command, wait for completion, and run it again immediately, or failing that, within ten seconds of completion, in order to avoid missing important performance data. Also, please note that for the first call, this command will return all values as false. Each subsequent call will report new record values.

Request

No parameters required.

Consult section **Standard Request Structure**.

Example

```
$ softnas-cmd perfmon -t
```

overview

Description

Obtain NAS overview status information.

Request

No parameters required.

Consult section **Standard Request Structure**.

Example

```
$ softnas-cmd overview -t
```

netstats

Description

Obtain network performance status information.

Request

No required parameters.

Consult section **Standard Request Structure**.

Example

```
$ softnas-cmd netstats -t
```

diskstats

Description

Obtain disk status information.

Request

No required parameters.

Consult section **Standard Request Structure**,

Example

```
$ softnas-cmd diskstats -t
```

procarcstatus

Description

Issue a disk management command.

Request

Consult section **Standard Request Structure**,

Example

```
$ softnas-cmd diskmgmt -t
```

hacommand

Description

Issue an HA command.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
checklicense	N
haLicenseKey	Y
checkHAcontroller	N
regname	N
install	N
add	N
del	N

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
add	Begin Add configuration of SNAP HA from the SnapReplicate "source" node
awsAccessKey	aws Access Key
awsSecretKey	aws Secret Key
vip	virtual IP (AWS Elastic IP)
hacontrollerip	Controller IP address (required only on Virtual machine)
del	delete 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 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
```

help

Description

Display help text.

Request

In addition to the **Standard Request Structure**, the following parameters are required.

For more information, consult section **Parameters.**

Parameter	Required?
commandname	N

Command Options

Command	Possible Values
Commandname	By adding the name of a given softnas command, you will be provided help text on that topic. Optional.

Example

```
$ softnas-cmd diskstats -t
```

AWS CloudFormation Templates

About the SoftNAS Cloud Formation Templates

SoftNAS Cloud® supports Amazon Web Services CloudFormation templates, which leverage the CLI for fully automating storage deployment. Deployment and setup automation is especially helpful for DevOps to create repeatable processes for QA testing, benchmarking, development environment creation and production rollouts.

The CloudFormation templates shipped with **SoftNAS Cloud®** are provided as working samples, to be modified and customized for each customers' particular needs and use cases. At a minimum, modify the templates to reference the **SoftNAS Cloud® AMIs** to be used; also modify the storage size, pool and volume names and other aspects to fit deployment needs.

Note: These templates are fairly basic. More advanced users may want to involve making additional CLI calls to perform certain tasks; e.g., configuring replication between two **SoftNAS Cloud®** instances.

There are two templates to choose from:

1. **SoftNAS-AWSCloudTemplate-Basic.json** - for paravirtual machine (PVM) systems, use **this template** as a starting point.

2. **Softnas-AWSCloudTemplateHVM.json** - for hardware virtual machine (HVM) systems, start with **this template**.

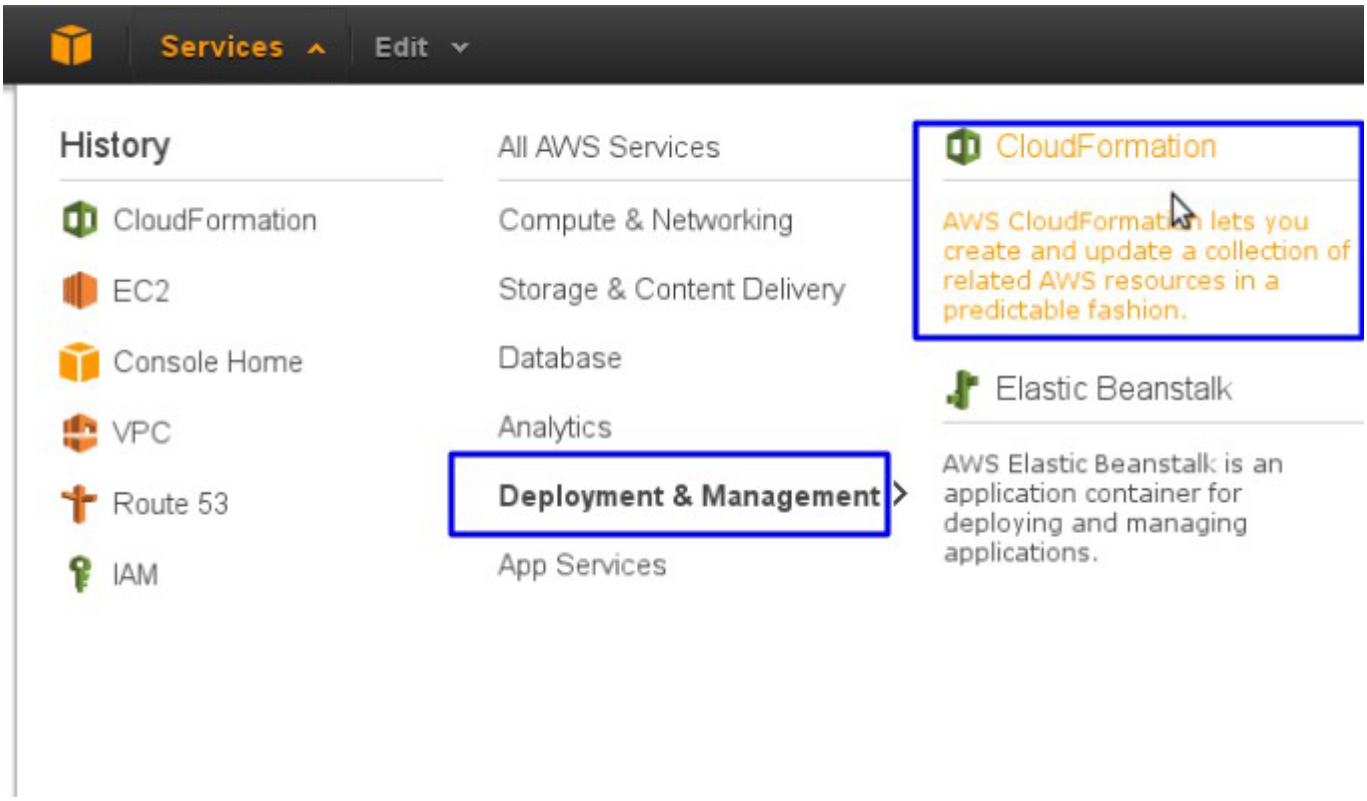
Using the SoftNAS CloudFormation Templates

Using AWS cloud formation templates, launch a **SoftNAS Cloud®** instance with the following features:

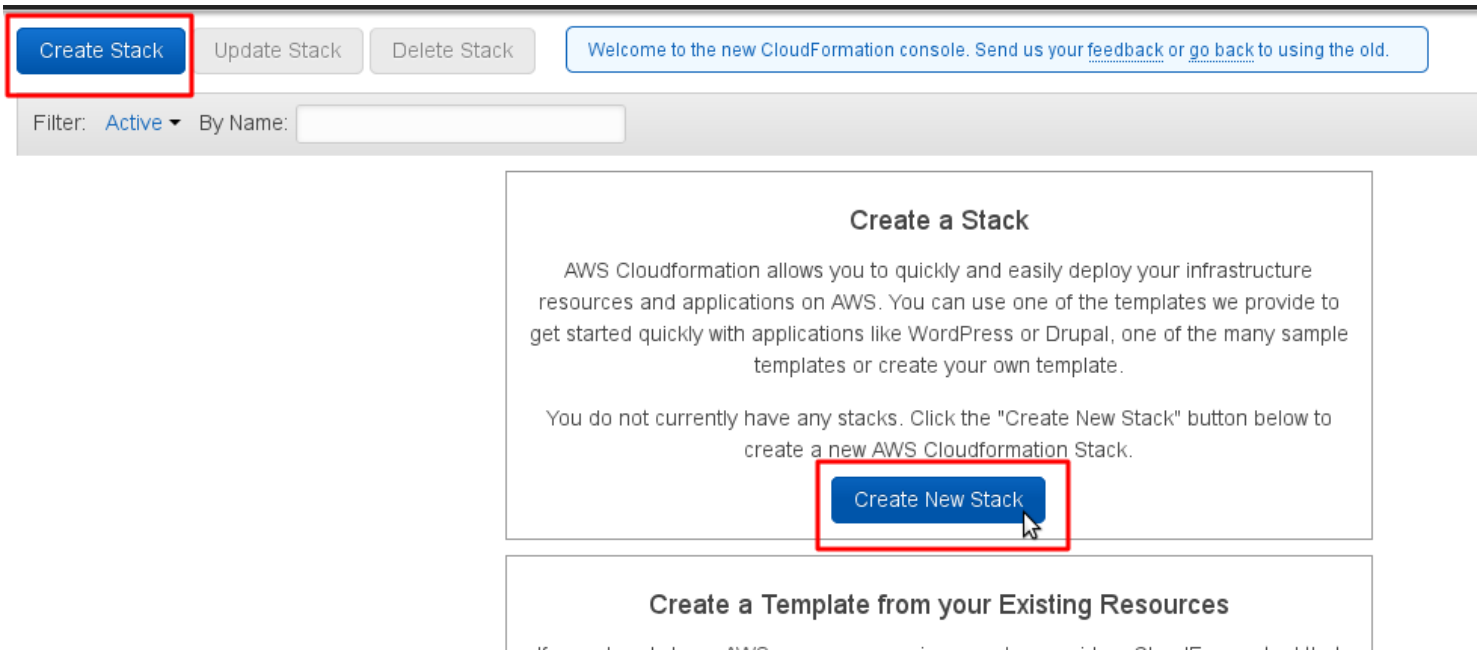
1. Two EBS volumes with 50 GB each
2. Allocate and assign EIP to the instance
3. Mount the EBS volume with storage pool in RAID0 (striping) configuration
4. Enable CIFS and NFS shares

To begin, download the template using one of the links above. Modify the template to include the SoftNAS Cloud® AMI ID along with any other preferred customizations.

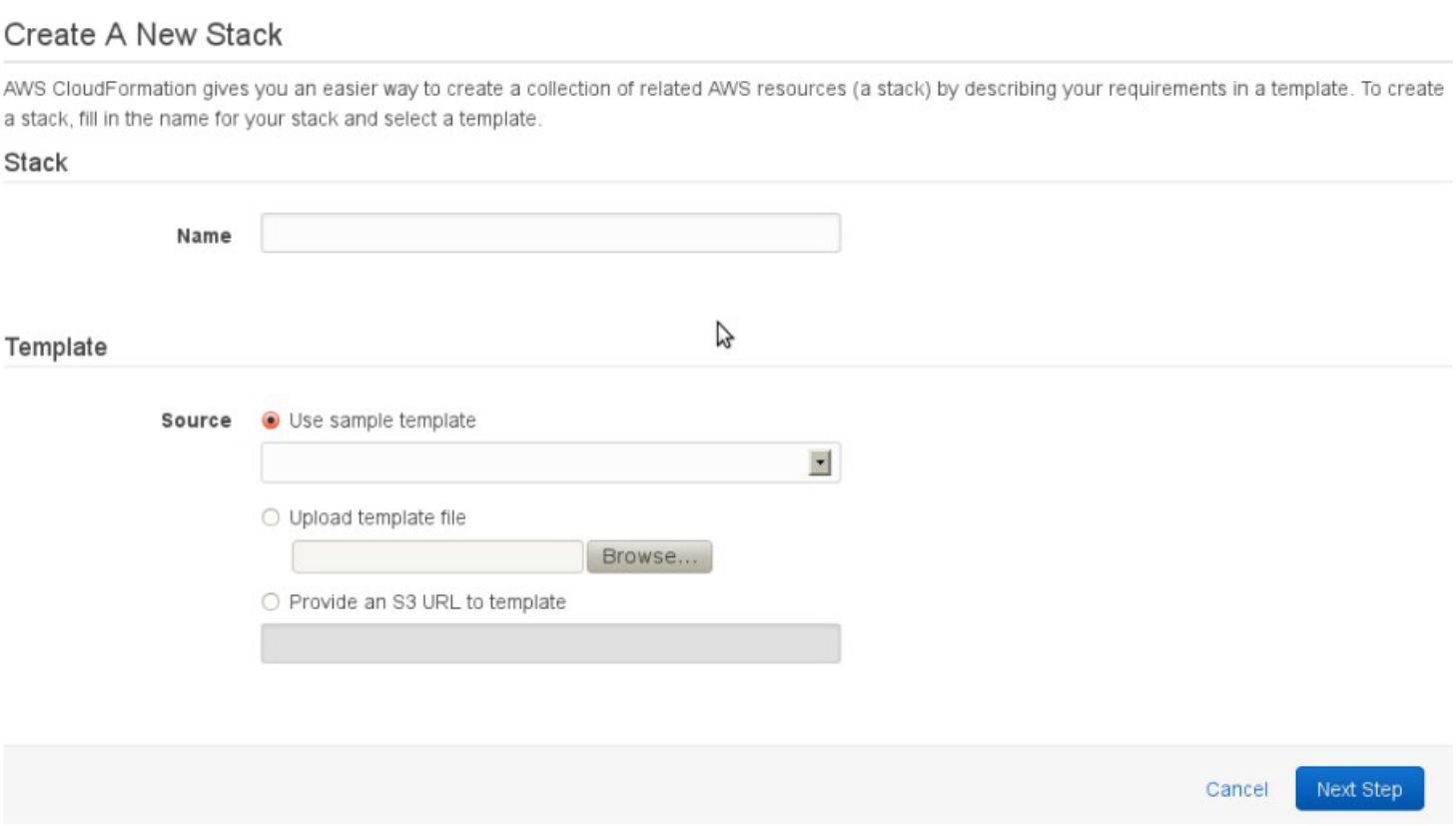
To deploy the template, in the AWS console choose **Services**, then **Deployment & Management** as shown below. Then choose **CloudFormation**.



Select **Create Stack** and then **Create New Stack**.

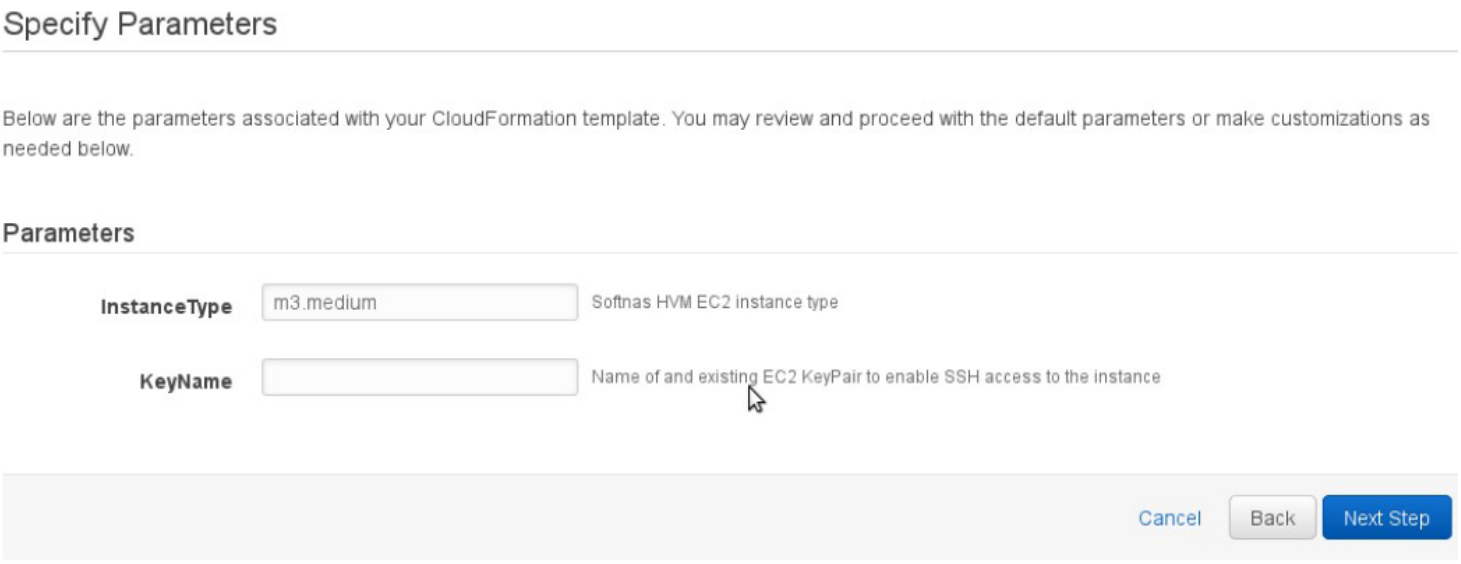


Enter the **Stack** name and **Template File**. Then select **upload** or provide an **s3 path** to the template file.

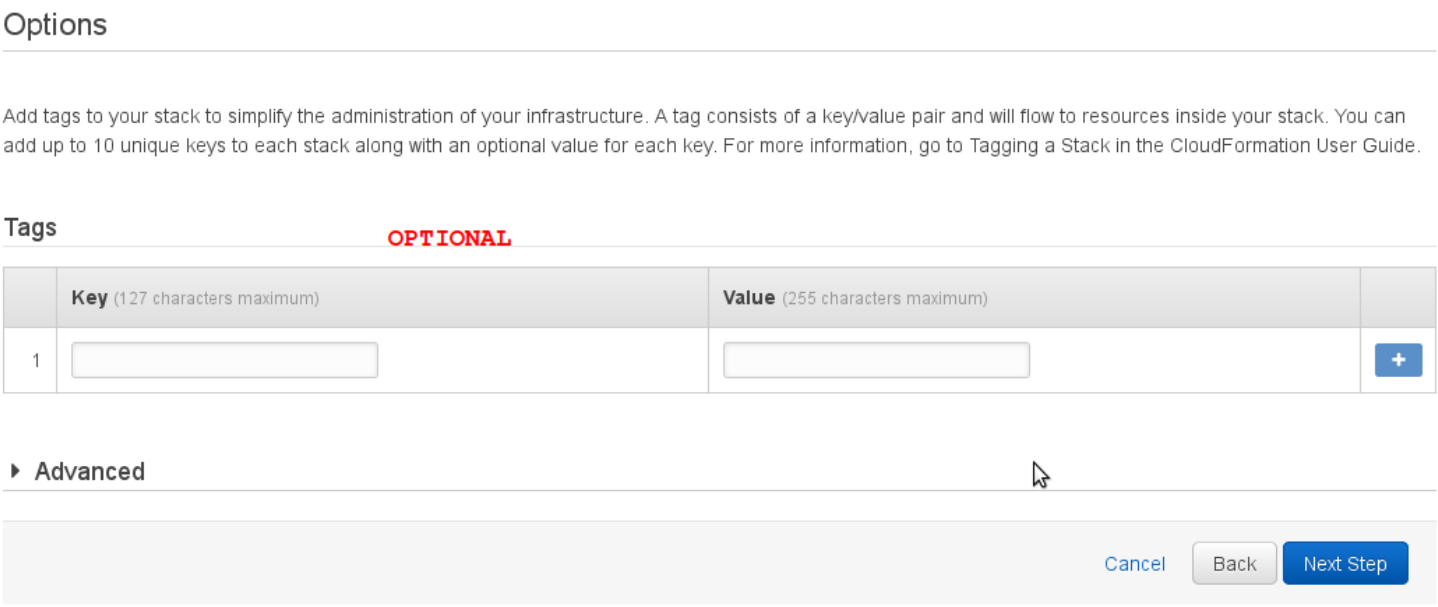


In the **Parameters** screen, enter the type of the instance (**t1.micro**, **m3.medium**, etc.) and the available **SSH key** name.

Note: Instance launch will fail if an SSH key is not assigned. If a key is not available, create one and then proceed from here.



Optionally, tag the instance name and skip the advanced settings. Instance is already tagged in the template with the name **Softnas-CF**. Click **Next Step** to proceed to the final screen.



Review the details and click **Create** to launch the instance. It may take up to 5 minutes to fully deploy and start the **SoftNAS Cloud®** instance using the CloudFormation template.

For more information on **CloudFormation** templates and customization, please refer to the **Command Line Interface (CLI) Reference** and AWS **CloudFormation Documentation**.