



NetApp

SolidFire CLI Tools User Guide

Version 1.4

May 2017 | 215-12208_A0
doccomments@netapp.com

Contents

CLI Tools overview	4
Installing CLI Tools	4
Using CLI Tools: Examples	5
Using CLI Tools with nonstandard options	7
Options	7
Output formats overview	8
Enabling autocomplete	11
Accessing embedded help	11
Managing connections	11
Enabling debug messages	12
List of commands	15
API command	16
Account commands	16
Backup target commands	18
Cluster commands	19
Drive commands	29
Hardware information commands	32
Hardware sensor commands	33
Initiator commands	33
Lightweight Directory Access Protocol (LDAP) commands	36
Logging session commands	39
Network commands	40
Node commands	41
Pairing commands	44
Restart commands	46
Schedule commands	48
Service command	52
Snapshot commands	52
Storage container commands	56
Test commands	57
Virtual network commands	59
Virtual volume commands	63
Volume commands	72
Volume access group commands	85
Where to find additional information	87
Contacting NetApp SolidFire Active Support	88
Copyright information	89
Trademark information	90
How to send comments about documentation and receive update notifications	91

CLI Tools overview

This document introduces CLI Tools and provides information about the prerequisites for using it, software components that it supports, and how to install and use it. It also lists the functions available in CLI Tools, and provides instructions for accessing inline help, managing connections, executing commands with standard and nonstandard parameters, and executing unsupported API methods.

CLI Tools is for users who want to control their SolidFire cluster from a command line. It also provides programmatic control of the system via scripts. From the command-line interface, you can execute commands that are based on the API methods supported by Element OS version 9.0 and later. You can choose to get the responses to the commands in any of the following formats:

- Tree
- JSON
- Pickle

CLI Tools also includes an inline help system that provides more information about the commands.

Support matrix

CLI Tools supports the following software and operating systems:

Component	Version
SolidFire Element OS	9.0 and later
Python	<ul style="list-style-type: none"> • Python 2: 2.7 and later • Python 3: 3.5 or later
Operating system	<ul style="list-style-type: none"> • Windows 7, 8, and 10 • macOS • Linux

Related tasks

[Accessing embedded help](#) on page 11
[Installing CLI Tools](#) on page 4

Related references

[Output formats overview](#) on page 8
[Options](#) on page 7

Installing CLI Tools

You can install CLI Tools by using either pip or by unpacking the `.tar.gz` file.

Before you begin

You must have Python and pip installed, if that is your preferred installation method. Starting with Python version 2.7.9, pip is installed by default with the Python installation.

Note: If you are a macOS user, you have to set up a virtual environment before you install CLI Tools. To set up the virtual environment, run the following command:

```
pip install virtualenv
virtualenv pythoncli
source pythoncli/bin/activate
```

This activates the virtual environment. After you do this, you can use one of the following methods to install CLI Tools.

Step

1. Use either of the following methods to install CLI Tools:

- Enter the following command from pip:
`pip install solidfire-cli`
- Navigate to `solidfirecli1.tar.gz`, and run `easy_install solidfirecli1.tar.gz` from the command line in the file.

Using CLI Tools: Examples

This section includes examples of using CLI Tools to perform common cluster operations, such as creating volumes and creating accounts. For detailed descriptions of the command options, see the embedded help.

Creating a new account

To create a new account in the system, use the command as shown in the following example:

```
sfcli Account add --username abc123
```

This command is based on the `AddAccount` API method, and creates a new account in the system. The `username` option is mandatory for this command. The value should be unique, between 1 and 64 characters in length. The value specified above (“abc123”) is an example.

The following options are optional for this command:

- `initiatorsecret`: The secret for CHAP authentication for the initiator. The value must be between 12 and 16 characters in length, and should be impenetrable.
- `targetsecret`: The secret for CHAP authentication for the target. The value must be between 12 and 16 characters in length, and should be impenetrable.
- `attributes`: List of name-value pairs in JSON object format.

Creating a new volume

You can create new volumes under the new account you created. To create a new volume, use the command as shown in the following example:

```
sfcli Volume create --name xyz456 --accountid 289 -- totalsize
1000000000 --enable512e true
```

This command is based on the `CreateVolume` API method, and creates a new volume on the cluster. The following options are mandatory for this command:

- **name:** Name of the volume; between 1 to 64 characters in length. It is best if the name is unique. In the example given above, name is specified as “xyz456.”
- **accountid:** Account ID for the owner of the volume. In the example given above, accountid is specified as “289.”
- **totalsize:** Total size of the volume, in bytes. In the example given above, totalsize is specified as “1000000000.”
- **enable512e:** Default value is false, which specifies that 512e emulation is not enabled. If you set the value to true, the volume provides 512 byte sector emulation. In the example given above, enable512e is set to “true.”

The following options are optional with this command:

- **attributes:** List of name-value pairs in JSON object format.
- **qos:** Initial quality of service settings for this volume. Default values are used, if you do not specify any values. Valid settings are minIOPS, maxIOPS, and burstIOPS.

Getting details about an account

To get details about an account, use the command as shown in the following example:

```
sfcli -c 0 Account GetByID --accountid 3065
```

Note: The account ID provided in the command above is an example.

This command is based on the `GetAccountByID` API method, and returns details about a specific account (3065, in the above example). You should specify the `accountid` option for this command.

The output for the above command is as follows:

```
account:
  attributes:
    target_secret:
      secret: q7788;0a:Cd0xCE4
    initiator_secret:
      secret: r0Jul}t:02"5p<L^
  status: active
  account_id: 3065
  username: example
  volumes:
```

Adding drives to a cluster

To add one or more available drives to a cluster, use the following command in the CLI:

```
sfcli Drive add <options>
```

This command is based on the `AddDrives` API method. When you add a node to the cluster or install new drives in an existing node, the new drives are marked as “available” and must be added before they can be utilized. When you add multiple drives, it is more efficient to add them in a single `AddDrives` method call. This reduces the amount of data balancing that must occur to stabilize the storage load on the cluster. When you add a drive, the system automatically determines the type of drive it should be. The method is asynchronous and returns immediately. However, it can take some time for the data in the cluster to be rebalanced using the newly added drives.

Using CLI Tools with nonstandard options

In some cases, you input options in JSON format directly to the command-line interface. There are two such “nonstandard” options. They are called `parameters` and `attributes`. In the inline help for these two parameters, a JSON object will be requested.

The following example shows the command syntax for making a call directly to the API:

```
$account = sfcli -c 0 SFApi Invoke --method GetAccountByID --parameters
{"accountID\":\"94\"}
```

Note: In the above example, `parameters` is a “nonstandard” option, for which the value has been specified as a JSON string.

Options

This section lists the options that you can use in CLI Tools.

Option	Description	Type
<code>-m, --mvip</code>	The management virtual IP address of the cluster.	Text
<code>-l, --login</code>	The login ID for the cluster.	Text
<code>-p, --password</code>	The password for the cluster.	Text
<code>-n, --name</code>	The name of the connection you want to use in <code>connections.csv</code> . You can use this option, if you have a stored connection.	Text
<code>-c</code> <code>--connectionindex</code>	The index of the connection you want to use in <code>connections.csv</code> . You can use this option, if you have a stored connection.	Integer
<code>-j, --json</code>	Displays the output in JSON format.	NA
<code>-k, --pickle</code>	Displays the output in pickled JSON format.	NA
<code>-d, --depth</code>	Displays the output in tree format with the depth that you specify.	Integer
<code>-f, --filter_tree</code>	Filters the fields that will be displayed in tree format. Specify the fields to be displayed in a comma-separated list of keypaths. For example, to filter the accounts list by username and status, specify “accounts.username,accounts.status”.	Text
<code>--debug [0 1 2 3]</code>	The debug level that you want to set.	NA
<code>--help</code>	Displays the inline help.	NA

Related references

[Output formats overview](#) on page 8

Output formats overview

When you run commands in CLI Tools, you can choose to obtain output in tree format, JSON format, or pickle format.

Tree format

This is the default output format. It provides output that does not require formatting (for example, removing extra characters). See the following sample:

```
accounts:

  attributes:
    storage_container_id:
      int:  <to see more details, increase depth>
    account_id:  2404
    status:  active
    initiator_secret:
      secret:  <to see more details, increase depth>
    target_secret:
      secret:  <to see more details, increase depth>
    volumes:
      <to see more details, increase depth>
    username:  XYZ

  attributes:
    storage_container_id:
      int:  <to see more details, increase depth>
    account_id:  2405
    status:  active
    initiator_secret:
      secret:  <to see more details, increase depth>
    target_secret:
      secret:  <to see more details, increase depth>
    volumes:
      <to see more details, increase depth>
    username:  haxecliFV8QdeT6fn5DxtvFuYzjsFwWtclYzXfT5-
NQE5pHiQAQBelNqVskTsJY8

  attributes:
    storage_container_id:
      int:  <to see more details, increase depth>
    account_id:  2406
    status:  active
    initiator_secret:
      secret:  <to see more details, increase depth>
    target_secret:
      secret:  <to see more details, increase depth>
    volumes:
      <to see more details, increase depth>
    username:
haxecliJ0yA7YVfCDiq9jZXdkdiKfkSyTK2flKk9Gi9NFq0677FcG44QIDc9inqF
```

JSON format

The output in this format resembles the output of the API. This format is useful if you want to save the data for later use with Postman or if you want to import it using JSON libraries. To get your output in this format, use the `-j` option after `sfcli` in your command. See the following sample:

```
{
  "accounts": [
    {
      "attributes": {},

```



```

        "initiator_secret": {
            "secret": "3,gG[sP02V'@911}"
        },
        "volumes": [
            4588
        ],
        "target_secret": {
            "secret": "aAe6Bb&q]63zU0Ei"
        },
        "status": "active",
        "account_id": 2404,
        "username": "XYZ",
        "storage_container_id": {
            "hex": "00000000000000000000000000000000"
        }
    },
    {
        "attributes": {},
        "initiator_secret": {
            "secret": "haxecliSgmB"
        },
        "volumes": [],
        "target_secret": {
            "secret": "haxecliLUuvO9s"
        },
        "status": "active",
        "account_id": 2405,
        "username": "haxecliFV8QdeT6fn5DxtvFuYzjsFwWtclYzXfT5-
NQE5pHiQAQBelNqVskTsJY8",
        "storage_container_id": {
            "hex": "00000000000000000000000000000000"
        }
    },
    {
        "attributes": {},
        "initiator_secret": {
            "secret": "haxecli7tOAtk"
        },
        "volumes": [],
        "target_secret": {
            "secret": "haxecliXFty4F2"
        },
        "status": "active",
        "account_id": 2406,
        "username":
        "haxecliJ0yA7YVfCDiq9jZXdkdikfkSytk2flKk9Gi9NFq0677Fcg44QIDc9inqF",
        "storage_container_id": {
            "hex": "00000000000000000000000000000000"
        }
    }
]
}

```

Pickle format

The output in this format is similar to the JSON format, except that there is an extra object type field for every object. This format is useful if you want to save data for use later with a Python SDK. To get your output in this format, use the `-k` option after `sfcli` in your command. See the following sample:

```

{
    "py/object": "solidfire.models.ListAccountsResult",
    "accounts": [
        {
            "account_id": 2404,
            "attributes": {},
            "status": "active",
            "py/object": "solidfire.models.Account",

```

```

        "target_secret": {
            "secret": "aAe6Bb&q]63zU0Ei",
            "py/object": "solidfire.custom.models.CHAPSecret"
        },
        "initiator_secret": {
            "secret": "3,gG[sP02V'@911}",
            "py/object": "solidfire.custom.models.CHAPSecret"
        },
        "username": "XYZ",
        "volumes": [
            4588
        ],
        "storage_container_id": {
            "hex": "00000000000000000000000000000000",
            "py/object": "uuid.UUID"
        }
    },
    {
        "account_id": 2405,
        "attributes": {},
        "status": "active",
        "py/object": "solidfire.models.Account",
        "target_secret": {
            "secret": "haxecliLUuv09s",
            "py/object": "solidfire.custom.models.CHAPSecret"
        },
        "initiator_secret": {
            "secret": "haxecliISgmB",
            "py/object": "solidfire.custom.models.CHAPSecret"
        },
        "username": "haxecliFV8QdeT6fn5DxtvFuYzjsFwWtc1YzXft5-
NQE5pHiQAQBelnqVskTsJY8",
        "volumes": [],
        "storage_container_id": {
            "hex": "00000000000000000000000000000000",
            "py/object": "uuid.UUID"
        }
    },
    {
        "account_id": 2406,
        "attributes": {},
        "status": "active",
        "py/object": "solidfire.models.Account",
        "target_secret": {
            "secret": "haxecliXFty4F2",
            "py/object": "solidfire.custom.models.CHAPSecret"
        },
        "initiator_secret": {
            "secret": "haxecli7tOatk",
            "py/object": "solidfire.custom.models.CHAPSecret"
        },
        "username":
        "haxecliJ0yA7YVfCDiq9jZXdkdiKfkSytk2flKk9Gi9NFq0677Fcg44QIDc9inqF",
        "volumes": [],
        "storage_container_id": {
            "hex": "00000000000000000000000000000000",
            "py/object": "uuid.UUID"
        }
    }
]
}

```

Enabling autocomplete

You can set up CLI Tools to complete the command syntax automatically for you.

Step

1. Do either of the following to use the autocomplete feature:

- Copy and paste the following syntax in your `.bashrc` file:

```
eval "$(_SFCLI_COMPLETE=source sfcli)"
```

Entering the syntax in your `.bashrc` file enables you to use the feature every time you open a new terminal window.

- Enter the following syntax in your command prompt when you want to use the feature:

```
eval "$(_SFCLI_COMPLETE=source sfcli)"
```

Accessing embedded help

CLI Tools includes an inline help that you can access through the command line. Help content includes details about how to use the commands.

Step

1. To access inline help, enter the following command:

```
sfcli --help
```

See the following examples for different levels at which you can access inline help:

- To access inline help for an object (for example, account), enter the following command:

```
sfcli account --help
```

- To access inline help for a specific command (for example, getbyid), enter the following command:

```
sfcli account getbyid --help
```

Managing connections

You can use commands in CLI Tools to manage connections to a SolidFire cluster. This section lists the commands that you can use to manage connections and provides some examples.

- To execute a command on a connection without storing it, use the `mvip`, `login`, and `password` options, as in the following example:

```
sfcli --mvip 10.117.60.15 --login admin --password admin Account List
```

If you do not store the connection, you need to enter your credentials every time you execute a command.

- To store a connection, use `Connection Push` and the `name` option, as in the following example:

```
sfcli --mvip 10.117.60.15 --login admin --password admin --name
"Example" Connection Push
```

When you store a connection, your credentials are saved locally in an encrypted form. This ensures that you do not need to enter your credentials every time you run a command.

- To use a stored connection, do the following:
 - Use `-n` or `--name` to find the connection by name, as in the following example:

```
sfcli -n Example Account List # by name
```

- Use `-c` or `--connectionIndex` to find the connection by index, as in the following example:

```
sfcli -c 0 Account List # by index
```

- Use connection 0 to use the default, as in the following example:

```
sfcli Account List # use connection 0
```

- To remove a connection, use the `Connection Remove` command as follows:
 - Use `-n` or `--name` to remove a connection by name, as in the following example:

```
sfcli Connection Remove -n Example
```

- Use `-i` to remove a connection by index, as in the following examples:

```
sfcli Connection Remove -i -1 # Removes the newly pushed connection.
```

```
sfcli Connection Remove -i 0 # Removes the oldest pushed connection.
```

```
sfcli Connection Remove -i 1 # Removes the second oldest connection.
```

- To list the stored connections, use the `Connection List` command as follows:

```
sfcli Connection List
```

- To prune a broken connection from the `connection.csv` file, use the `Connection Prune` command as follows:

```
sfcli Connection Prune
```

Enabling debug messages

You can use the `debug` option in CLI Tools to enable debug messages. Debug messages provide information about the request that you sent to the system. This information will help you understand command errors. You also need this information if you have to contact NetApp SolidFire Active Support.

You can specify the debug level by choosing from one of the following values:

- 0: Critical logging. Displays only errors.

- 1: Warning logging. Displays errors and warnings.
- 2: Info logging. Displays errors, warnings, and information.
- 3: Debug logging. Displays errors, warnings, information, and debug information.

To set the debug level, specify the debug option in your command as shown in the following example:

```
sfcli --debug 2 account list
```

In the above example, the debug level specified is 2, which returns the following response:

```
INFO in cmd_account.py@50: startaccountid = None;limit = None;
accounts:

    target_secret:
        secret: <to see more details, increase depth>
    initiator_secret:
        secret: <to see more details, increase depth>
    storage_container_id:
        int: <to see more details, increase depth>
    username: haxepyFK5C2hKpfbVitWjvZ8cxo3-
nf0lybhhqj7feAS7FcOBxKR2DjtMCHevhdN1
    account_id: 2385
    attributes:
    volumes:
        <to see more details, increase depth>
        <to see more details, increase depth>
        <to see more details, increase depth>
    status: active

    target_secret:
        secret: <to see more details, increase depth>
    initiator_secret:
        secret: <to see more details, increase depth>
    storage_container_id:
        int: <to see more details, increase depth>
    username: haxecli55I8QWrxEB2oAuOzqYqk8zEu4dWjhFDjPHrh-
TWcsxzd0sR4adpN1Jq
    account_id: 2559
    attributes:
    volumes:
        <to see more details, increase depth>
    status: active
```

You might want to save the above output. To save the information from the server, use the following command:

```
sfcli --debug 2 account list 2> info.txt
```

info.txt will include only the following text labeled “INFO” in the above sample response:

```
INFO in cmd_account.py@50: startaccountid = None;limit = None;
```

To save the response from the server, use the following command:

```
sfcli --debug 2 account list > data.txt
```

data.txt will include the following text labeled “accounts” in the above sample response:

```
accounts:

  target_secret:
    secret: <to see more details, increase depth>
  initiator_secret:
    secret: <to see more details, increase depth>
  storage_container_id:
    int: <to see more details, increase depth>
  username: haxepyFK5C2hKpfbVitWjvZ8cxo3-
nf01ybhqj7feAS7FcOBxKR2DjtMCHeVhdNl
  account_id: 2385
  attributes:
  volumes:
    <to see more details, increase depth>
    <to see more details, increase depth>
    <to see more details, increase depth>
  status: active

  target_secret:
    secret: <to see more details, increase depth>
  initiator_secret:
    secret: <to see more details, increase depth>
  storage_container_id:
    int: <to see more details, increase depth>
  username: haxecli55I8QWrxEB2oAuOzqYqk8zEu4dWjhFDjPHrh-
TWcsxzdZ0sjR4adpNlJq
  account_id: 2559
  attributes:
  volumes:
    <to see more details, increase depth>
  status: active
```

List of commands

This section lists all the commands that you can use in the tool and their descriptions.

- [*API command*](#) on page 16
- [*Account commands*](#) on page 16
- [*Backup target commands*](#) on page 18
- [*Cluster commands*](#) on page 19
- [*Drive commands*](#) on page 29
- [*Hardware information commands*](#) on page 32
- [*Hardware sensor commands*](#) on page 33
- [*Initiator commands*](#) on page 33
- [*Lightweight Directory Access Protocol \(LDAP\) commands*](#) on page 36
- [*Logging session commands*](#) on page 39
- [*Network commands*](#) on page 40
- [*Node commands*](#) on page 41
- [*Pairing commands*](#) on page 44
- [*Restart commands*](#) on page 46
- [*Schedule commands*](#) on page 48
- [*Service command*](#) on page 52
- [*Snapshot commands*](#) on page 52
- [*Storage container commands*](#) on page 56
- [*Test commands*](#) on page 57
- [*Virtual network commands*](#) on page 59
- [*Virtual volume commands*](#) on page 63
- [*Volume commands*](#) on page 72
- [*Volume access group commands*](#) on page 85

API command

You can use the command in this section to invoke any API method that is supported by the SolidFire API for the version and port that the connection is using.

sfcli SFApi invoke <options>

Description	Options
You can use this command to invoke any API method supported by the SolidFire API for the version and port the connection is using. The command returns a nested hashtable of key-value pairs that contain the result of the invoked method.	<p>method: The name of the method to invoke. This is case sensitive.</p> <p>parameters: An object, normally a dictionary or hashtable of the key-value pairs, to be passed as the parameters for the method being invoked.</p>

Account commands

This section describes the account-related commands that you can use in CLI Tools.

sfcli Account list <options>

Description	Options
This command returns the entire list of accounts, with optional paging support.	<p>startaccountid: Starting AccountID to return. If no account exists with this AccountID, the next account by AccountID order is used as the start of the list. To page through the list, pass the AccountID of the last account in the previous response + 1.</p> <p>limit: Maximum number of AccountInfo objects to return.</p> <p>includestoragecontainers: Includes storage containers in the response by default. To exclude storage containers, set to false.</p>

sfcli Account getefficiency <options>

Description	Options
This command enables you to retrieve efficiency statistics about a volume account. It returns efficiency information only for the account you specify.	<p>accountid: Specifies the volume account for which efficiency statistics are returned.</p>

sfcli Account modify <options>

Description	Options
This command enables you to modify an existing account. When you lock an account, any existing connections from that account are immediately terminated. When you change an account's CHAP settings, any existing connections remain active, and the new CHAP settings are used on subsequent connections or reconnections. To clear an account's attributes, specify {} for the attributes option.	<p>accountid: Specifies the AccountID for the account to be modified.</p> <p>username: Specifies the username associated with the account. (Might be 1 to 64 characters in length).</p> <p>status: Sets the status for the account. Possible values are:</p> <ul style="list-style-type: none"> active: The account is active and connections are allowed. locked: The account is locked and connections are refused. <p>initiatorsecret: Specifies the CHAP secret to use for the initiator. This secret must be 12-16 characters in length and should be impenetrable. The initiator CHAP secret must be unique and cannot be the same as the target CHAP secret.</p> <p>targetsecret: Specifies the CHAP secret to use for the target (mutual CHAP authentication). This secret must be 12-16 characters in length and should be impenetrable. The target CHAP secret must be unique and cannot be the same as the initiator CHAP secret.</p> <p>attributes: List of name-value pairs in JSON object format.</p>

sfcli Account remove <options>

Description	Options
This command enables you to remove an existing account. You must delete and purge all volumes associated with the account before you can remove the account. If volumes on the account are still pending deletion, you cannot use this command to remove the account.	<p>accountid: Specifies the AccountID for the account to be removed.</p>

sfcli Account getbyname <options>

Description	Options
This command enables you to retrieve details about a specific account, given its username.	<p>username: Username for the account.</p>

sfcli Account add <options>

Description	Options
This command enables you to add a new account to the system. You can create new volumes under the new account. The CHAP settings you specify for the account apply to all volumes owned by the account.	<p>username: Specifies the username for this account. (Might be 1 to 64 characters in length).</p> <p>initiatorsecret: The CHAP secret to use for the initiator. This secret must be 12-16 characters in length and should be impenetrable. The initiator CHAP secret must be unique and cannot be the same as the target CHAP secret. If unspecified, a random secret is created.</p> <p>targetsecret: The CHAP secret to use for the target (mutual CHAP authentication). This secret must be 12-16 characters in length and should be impenetrable. The target CHAP secret must be unique and cannot be the same as the initiator CHAP secret. If unspecified, a random secret is created.</p> <p>attributes: List of name-value pairs in JSON object format.</p>

sfcli Account getbyid <options>

Description	Options
This command enables you to return details about a specific account, given its accountid.	accountid: Specifies the account for which details are gathered.

Backup target commands

You can use the commands in this section to perform various operations related to your backup targets.

sfcli BackupTarget modify <options>

Description	Options
This command enables you to change attributes of a backup target.	<p>backuptargetid: The unique target ID for the target to modify.</p> <p>name: The new name for the backup target.</p> <p>attributes: List of name-value pairs in JSON object format.</p>

sfcli BackupTarget create <options>

Description	Options
This command enables you to create and store backup target information so that you do not need to reenter it each time a backup is created.	<p>name: The name of the backup target.</p> <p>attributes: List of name-value pairs in JSON object format.</p>

sfcli BackupTarget list <options>

Description	Options
This command enables you to retrieve information about all backup targets that have been created.	None

sfcli BackupTarget remove <options>

Description	Options
This command enables you to delete backup targets.	backuptargetid: The unique target ID of the target to remove.

sfcli BackupTarget get <options>

Description	Options
This command enables you to get information about a specific backup target that you created.	backuptargetid: The unique identifier assigned to the backup target.

Cluster commands

You can use the commands in this section to perform various operations on the cluster, such as getting the configuration information about the cluster, retrieving information about any faults on your cluster, and so on.

sfcli Cluster getinfo <options>

Description	Options
This command enables you to return configuration information about the cluster.	None

sfcli Cluster getcompletestats <options>

Description	Options
NetApp engineering uses this command to troubleshoot new features. The data returned from is not documented, changes frequently, and is not guaranteed to be accurate. NetApp does not recommend using this command for collecting performance data or any other management integration with a SolidFire cluster.	None

sfcli Cluster getrawstats <options>

Description	Options
NetApp engineering uses this command to troubleshoot new features. The data returned is not documented, changes frequently, and is not guaranteed to be accurate. NetApp does not recommend using this command for collecting performance data or any other management integration with a SolidFire cluster.	None

sfcli Cluster getapi <options>

Description	Options
This command enables you to return a list of all the API methods and supported API endpoints that can be used in the system.	None

sfcli Cluster disablesnmp <options>

Description	Options
This command enables you to disable SNMP on the cluster nodes.	None

sfcli Cluster getsnmpstate <options>

Description	Options
This command enables you to get the current state of the SNMP feature.	None

sfcli Cluster getsnmpinfo <options>

Description	Options
<p>This command enables you to retrieve the current simple network management protocol (SNMP) configuration information.</p> <p>Note: This command is deprecated for versions later than Element OS version 8.0. NetApp recommends that you use the <code>getsnmpstate</code> and <code>setacl</code> commands.</p>	None

sfcli Cluster getconfig <options>

Description	Options
This command enables you to return information about the cluster configuration this node uses to communicate with the cluster that it is a part of.	None

sfcli Cluster deleteallsupportbundles <options>

Description	Options
This command enables you to delete all support bundles generated with the <code>createsupportbundle</code> command.	None

sfcli Cluster getsystemstatus <options>

Description	Options
This command enables you to return whether a reboot is required or not.	None

sfcli Cluster setsnmpttrapinfo <options>

Description	Options
<p>You can use this command to enable and disable the generation of cluster SNMP notifications (traps) and to specify the set of network host computers that receive the notifications. The values you pass with each <code>setsnmpttrapinfo</code> command replace all values set in any previous runs of the <code>setsnmpttrapinfo</code> command.</p>	<p><code>traprecipients</code>: List of hosts that are to receive the traps generated by the Cluster Master. At least one object is required if any one of the trap types is enabled.</p> <p><code>clusterfaultttrapsenabled</code>: If the value is set to true, a corresponding <code>solidFireClusterFaultNotification</code> is sent to the configured list of trap recipients when a cluster fault is logged. The default value is false.</p> <p><code>clusterfaultresolvedtrapsenabled</code>: If the value is set to true, a corresponding <code>solidFireClusterFaultResolvedNotification</code> is sent to the configured list of trap recipients when a cluster fault is resolved. The default value is false.</p> <p><code>clustereventttrapsenabled</code>: If the value is set to true, a corresponding <code>solidFireClusterEventNotification</code> is sent to the configured list of trap recipients when a cluster event is logged. The default value is false.</p>

sfcli Cluster listfaults <options>

Description	Options
<p>This command enables you to retrieve information about any faults detected on the cluster. With this method, you can retrieve both current faults as well as faults that have been resolved. The system caches faults every 30 seconds.</p>	<p><code>bestpractices</code>: Specifies whether to include faults triggered by suboptimal system configuration. Possible values are: true or false.</p> <p><code>faulttypes</code>: Determines the types of faults returned. Possible values are: current (List active, unresolved faults), resolved (list faults that were previously detected and resolved), and all (default; list both current and resolved faults). You can see the fault status in the resolved field of the Cluster Fault object.</p>

sfcli Cluster listadmins <options>

Description	Options
This command returns the list of all cluster administrators for the cluster. There can be several cluster administrator accounts with different levels of permissions. There can be only one primary cluster administrator in the system. The primary Cluster Admin is the administrator that was created when the cluster was created. You can also create LDAP administrators when setting up an LDAP system on the cluster.	None

sfcli Cluster create <options>

Description	Options
<p>This command enables you to initialize the node in a cluster that has ownership of the storage virtual IP address (SVIP) and management virtual IP address (MVIP). Each new cluster is initialized using the management IP (MIP) of the first node in the cluster. This command also automatically adds all the nodes being configured into the cluster. You only need to use this command once each time a new cluster is initialized.</p> <p>Note: You need to log in to the node that is used as the master node for the cluster. After you log in, run <code>getbootstrapconfig</code> on the node to get the IP addresses for the rest of the nodes that you want to include in the cluster. Then, run the <code>create</code> command.</p>	<p><code>accepteula</code>: Required to indicate your acceptance of the End User License Agreement when creating this cluster. Set this option to true.</p> <p><code>mvip</code>: Floating (virtual) IP address for the cluster on the management network.</p> <p><code>svip</code>: Floating (virtual) IP address for the cluster on the storage (iSCSI) network.</p> <p><code>repcount</code>: Number of replicas of each piece of data to store in the cluster. Valid value is 2.</p> <p><code>username</code>: Username for the cluster admin.</p> <p><code>password</code>: Initial password for the cluster admin account.</p> <p><code>nodes</code>: CIP/SIP addresses of the initial set of nodes making up the cluster. This node's IP must be in the list.</p> <p><code>attributes</code>: List of name-value pairs in JSON object format.</p>

sfcli Cluster enableencryptionatrest <options>

Description	Options
<p>This command enables the Advanced Encryption Standard (AES) 256-bit encryption at rest on the cluster, so that the cluster can manage the encryption key used for the drives on each node. This feature is not enabled by default. When you enable this feature, the cluster automatically manages encryption keys internally for the drives on each node in the cluster.</p> <p>Note: If you have a node type with a model number ending in “-NE”, the command fails with a response of “Encryption not allowed. Cluster detected non-encryptable node”. You should enable or disable encryption only when the cluster is running and in a healthy state. You can enable or disable encryption at your discretion and as often as you need.</p>	None

sfcli Cluster disableencryptionatrest <options>

Description	Options
<p>This command enables you to remove the encryption that was previously applied to the cluster using the <code>enableencryptionatrest</code>. This command is asynchronous and returns a response before encryption is disabled.</p>	None

sfcli Cluster addadmin <options>

Description	Options
<p>This command enables you to add a new Cluster Admin account. Cluster Admins can manage the cluster using the API and management tools. Cluster Admins are completely separate and unrelated to standard tenant accounts. NetApp recommends using multiple Cluster Admin accounts for different users and applications. You should give each Cluster Admin the minimal permissions necessary; this reduces the potential impact of credential compromise.</p>	<p><code>username</code>: Unique username for this Cluster Admin. Must be between 1 and 1024 characters in length.</p> <p><code>access</code>: Controls which methods this Cluster Admin can use. For more details on the levels of access, see “Access Control” in the <i>Element API Reference Guide</i>.</p> <p><code>password</code>: Password used to authenticate this Cluster Admin.</p> <p><code>accepteula</code>: Required to indicate your acceptance of the End User License Agreement when creating this cluster. Set this option to true.</p> <p><code>attributes</code>: List of name-value pairs in JSON object format.</p>

sfcli Cluster setntpinfo <options>

Description	Options
<p>This command enables you to configure NTP on cluster nodes. The values you set with this interface apply to all nodes in the cluster. If an NTP broadcast server periodically broadcasts time information on your network, you can optionally configure nodes as broadcast clients.</p> <p>Note: NetApp recommends using NTP servers that are internal to your network, rather than the installation defaults.</p>	<p>servers: List of NTP servers to add to each nodes NTP configuration.</p> <p>broadcastclient: Enables every node in the cluster as a broadcast client.</p>

sfcli Cluster setconfig <options>

Description	Options
<p>This command enables you to set the configuration the node uses to communicate with the cluster it is associated with.</p>	<p>cluster: Objects that are changed for the cluster interface settings.</p>

sfcli Cluster modifyadmin <options>

Description	Options
<p>This command enables you to change the settings for a Cluster Admin or LDAP Cluster Admin. You cannot change access for the administrator Cluster Admin account.</p>	<p>clusteradminid: ClusterAdminID for the Cluster Admin or LDAP Cluster Admin to modify.</p> <p>password: Password used to authenticate this Cluster Admin.</p> <p>access: Controls which methods this Cluster Admin can use. For more details, see “Access Control” in the <i>Element API Reference Guide</i>.</p> <p>attributes: List of name-value pairs in JSON object format.</p>

sfcli Cluster getsnmptrapinfo <options>

Description	Options
<p>You can use this command to return current SNMP trap configuration information.</p>	<p>None</p>

sfcli Cluster listevents <options>

Description	Options
<p>This command returns events detected on the cluster, sorted from oldest to newest.</p>	<p>maxevents: Specifies the maximum number of events to return.</p> <p>starteventid: Identifies the beginning of a range of events to return.</p> <p>endeventid: Identifies the end of a range of events to return.</p>

sfcli Cluster snmpsendtesttraps <options>

Description	Options
This command enables you to test SNMP functionality for a cluster. It instructs the cluster to send test SNMP traps to the currently configured SNMP manager.	None

sfcli Cluster removeadmin <options>

Description	Options
This command enables you to remove a Cluster Admin. You cannot remove the administrator Cluster Admin account.	clusteradminid: ClusterAdminID for the Cluster Admin to remove.

sfcli Cluster modifyfullthreshold <options>

Description	Options
You can use this command to change the level at which the system generates an event when the storage cluster approaches a certain capacity utilization.	<p>stage2awarethreshold: The number of nodes of capacity remaining in the cluster before the system triggers a capacity notification.</p> <p>stage3blockthresholdpercent: The percentage of block storage utilization below the Error threshold that causes the system to trigger a cluster Warning alert.</p> <p>maxmetadataoverprovisionfactor : A value representative of the number of times metadata space can be overprovisioned relative to the amount of space available.</p>

sfcli Cluster getlimits <options>

Description	Options
<p>This command enables you to retrieve the limit values set by the API. These values might change between releases of Element OS, but do not change without an update to the system.</p> <p>Note: This command returns the limits for the current software version regardless of the API endpoint version used to pass the method.</p>	None

sfcli Cluster getcurrentadmin <options>

Description	Options
This command returns information for the current primary cluster administrator. The primary cluster administrator was created when the cluster was created.	None

sfcli Cluster createsupportbundle <options>

Description	Options
This command enables you to create a support bundle file under the node's directory. The bundle is stored on the node as a <code>tar.gz</code> file.	<p><code>bundlename</code>: The unique name for the support bundle. If no name is provided, <code>supportbundle</code> and the node name are used as the filename.</p> <p><code>extraargs</code>: Passed to the <code>sf_make_support_bundle</code> script. You should use this option only at the request of NetApp SolidFire Support.</p> <p><code>timeoutsec</code>: The number of seconds to allow the support bundle script to run before stopping. The default value is 1500 seconds.</p>

sfcli Cluster getcapacity <options>

Description	Options
This command enables you to return the high-level capacity measurements for an entire cluster. You can use the fields returned from this command to calculate the efficiency rates that are displayed in the Element OS Web UI.	None

sfcli Cluster getntpinfo <options>

Description	Options
This command enables you to return the current network time protocol (NTP) configuration information.	None

sfcli Cluster getversioninfo <options>

Description	Options
This command enables you to retrieve information about the Element software version running on each node in the cluster. This command also returns information about nodes that are currently in the process of upgrading software.	None

sfcli Cluster setsnmpacl <options>

Description	Options
<p>This command enables you to configure SNMP access permissions on the cluster nodes. The values you set with this interface apply to all nodes in the cluster, and the values you specify replace, in whole, all values set in any previous runs of the <code>setsnmpacl</code> command.</p> <p>Note: The values set with this interface replace all network or <code>usmUsers</code> values set with the <code>setsnmpinfo</code> command.</p>	<p><code>networks</code>: List of networks and type of access they have to the SNMP servers running on the cluster nodes. See <code>SNMP Network Object</code> for possible values. This option is required if SNMP v3 is disabled.</p> <p><code>usmusers</code>: List of users and the type of access they have to the SNMP servers running on the cluster nodes.</p>

sfcli Cluster clearfaults <options>

Description	Options
<p>This command enables you to remove information about both current and previously detected faults. You can remove both resolved and unresolved faults.</p>	<p><code>faulttypes</code>: Determines the types of faults cleared. Possible values are:</p> <ul style="list-style-type: none"> • <code>current</code>: Faults that are currently detected and have not been resolved. • <code>resolved</code>: (Default) Faults that were previously detected and resolved. • <code>all</code>: Both current and resolved faults. The fault status can be determined by the <code>resolved</code> field of the fault object.

sfcli Cluster getsnmpacl <options>

Description	Options
<p>This command enables you to return the current SNMP access permissions on the cluster nodes.</p>	<p>None</p>

sfcli Cluster getstate <options>

Description	Options
<p>This command enables you to indicate if a node is part of a cluster or not. The three states are:</p> <ul style="list-style-type: none"> • <code>Available</code>: Node has not been configured with a cluster name. • <code>Pending</code>: Node is pending for a specific named cluster and can be added. • <code>Active</code>: Node is an active member of a cluster and may not be added to another cluster. 	<p><code>force</code>: To run this command, set this option to true.</p>

sfcli Cluster enablesnmp <options>

Description	Options
This command enables you to enable SNMP on cluster nodes. When you enable SNMP, the action applies to all nodes in the cluster, and the values you specify replace, in whole, all values set in any previous runs of the <code>enablesnmp</code> command.	<code>snmpv3enabled</code> : If set to true, SNMP v3 is enabled on each node in the cluster. If set to false, SNMP v2 is enabled.

sfcli Cluster getstats <options>

Description	Options
This command enables you to retrieve high-level activity measurements for the cluster. Values returned are cumulative from the creation of the cluster.	None

sfcli Cluster getmasternodeid <options>

Description	Options
This command enables you to retrieve the ID of the node that can perform cluster-wide administration tasks and holds the SVIP and MVIP.	None

sfcli Cluster setsnmpinfo <options>

Description	Options
<p>This command enables you to configure SNMP version 2 and version 3 on cluster nodes. The values you set with this interface apply to all nodes in the cluster, and the values you specify replace, in whole, all values set in any previous runs of the <code>setsnmpinfo</code> command.</p> <p>Note: The <code>setsnmpinfo</code> command is deprecated. Use the <code>enablesnmp</code> and <code>setsnmppacl</code> commands instead.</p>	<p><code>networks</code>: List of networks and type of access they have to the SNMP servers running on the cluster nodes. See the SNMP Network Object for possible values. This option is required only for SNMP v2.</p> <p><code>enabled</code>: If set to true, SNMP is enabled on each node in the cluster.</p> <p><code>snmpv3enabled</code>: If set to true, SNMP v3 is enabled on each node in the cluster.</p> <p><code>usmusers</code>: If SNMP v3 is enabled, this value must be passed in place of the <code>networks</code> option. This option is required only for SNMP v3.</p>

sfcli Cluster getfullthreshold <options>

Description	Options
<p>This command enables you to view the stages set for cluster fullness levels. It returns all fullness metrics for the cluster.</p> <p>Note: When a cluster reaches the Error stage of block cluster fullness, the maximum IOPS on all volumes are reduced linearly to the volume's minimum IOPS as the cluster approaches the Critical stage. This helps prevent the cluster from reaching the Critical stage of block cluster fullness.</p>	None

sfcli Cluster listsyncjobs <options>

Description	Options
<p>This enables you to return information about synchronization jobs that are running on a SolidFire cluster. The type of synchronization jobs that are returned with this command are slice, clone, and remote.</p>	None

Drive commands

You can use the commands in this section to perform operations, such as adding drives to a cluster, removing drives from a cluster, getting hardware information about the drives in a cluster.

sfcli Drive reset <options>

Description	Options
<p>This command enables you to proactively initialize drives and remove all data currently residing on a drive. The drive can then be reused in an existing node or used in an upgraded node. You must specify the <code>force</code> option for this command.</p>	<p><code>drives</code>: List of device names (not driveIDs) to reset.</p> <p><code>force</code>: Required option to successfully reset a drive.</p>

sfcli Drive secureerase <options>

Description	Options
<p>This command enables you to remove any residual data from drives that have a status of “available”. You might want to use this when replacing a drive nearing the end of its service life that contained sensitive data.</p>	<p><code>drives</code>: List of driveIDs to be secure erased.</p>

sfcli Drive listdrivestats <options>

Description	Options
This command enables you to retrieve high-level activity measurements for multiple drives in the cluster. By default, this command returns statistics for all drives in the cluster, and these measurements are cumulative from the addition of the drive to the cluster. Some values this command returns are specific to block drives, and some are specific to metadata drives.	drives: Optional list of DriveIDs for which to return drive statistics. If you omit this option, measurements for all drives are returned.

sfcli Drive list <options>

Description	Options
This command enables you to retrieve the list of the drives that exist in the cluster's active nodes. It returns drives that have been added as volume metadata or block drives as well as drives that have not been added and are available.	None

sfcli Drive remove <options>

Description	Options
This command enables you to proactively remove drives that are part of the cluster. You might want to use this when reducing cluster capacity or preparing to replace drives nearing the end of their service life. Any data on the drives is removed and migrated to other drives in the cluster before the drive is removed from the cluster. This is an asynchronous method. Depending on the total capacity of the drives being removed, it might take several minutes to migrate all of the data.	drives: List of driveIDs to remove from the cluster. forceduringupgrade: If you want to remove a drive during upgrade, this must be set to true.

sfcli Drive gethardwareinfo <options>

Description	Options
This command returns all the hardware information for the given drive. This generally includes details about manufacturers, vendors, versions, and other associated hardware identification information.	driveid: DriveID for the drive information requested.

sfcli Drive add <options>

Description	Options
This command enables you to add one or more available drives to the cluster, enabling the drives to host a portion of the cluster's data.	<p><code>drives</code>: Returns information about each drive to be added to the cluster. Possible values are:</p> <ul style="list-style-type: none"> <code>driveID</code>: Optional; ID of the drive to add. <code>type</code>: Type of drive to add; valid values are “slice” or “block.”

sfcli Drive getstats <options>

Description	Options
This command returns high-level activity measurements for a single drive. Values are cumulative from the addition of the drive to the cluster. Some values are specific to block drives. You might not obtain statistical data for both block and metadata drives when you run this method.	<code>driveid</code> : Specifies the drive for which statistics are gathered.

sfcli Drive getconfig <options>

Description	Options
This command enables you to display drive information for expected slice and block drive counts as well as the number of slices and block drives that are currently connected to the node.	None

sfcli Drive test <options>

Description	Options
<p>This command enables you to run a hardware validation on all drives on the node. It detects hardware failures on the drives (if present) and reports them in the results of the validation tests. You can use this command only on nodes that are not “active” in a cluster.</p> <p>Note: This test takes approximately 10 minutes.</p>	<code>minutes</code> : Specifies the number of minutes to run the test.

sfcli Drive listhardware <options>

Description	Options
This command returns all the drives connected to a node. Use this command on individual nodes to return drive hardware information or use it on the cluster master node MVIP to see information for all the drives on all nodes.	<code>force</code> : Set this option to true.

Hardware information commands

You can use the commands in this section to get hardware information about nodes and drives in a cluster.

sfcli Hardware getnvraminfo <options>

Description	Options
This command enables you to retrieve information from each node about the NVRAM card.	<code>force</code> : Required option for the method to successfully run on all nodes in the cluster.

sfcli Hardware gethardwareinfo <options>

Description	Options
The command allows you to return hardware information and status for a single node. This generally includes details about manufacturers, vendors, versions, drives, and other associated hardware identification information.	None

sfcli Hardware getnodeinfo <options>

Description	Options
This command enables you to return all the hardware information and status for the node specified. This generally includes details about manufacturers, vendors, versions, and other associated hardware identification information.	<code>nodeid</code> : The ID of the node for which you need the hardware information. Information about a Fibre Channel node is returned if a Fibre Channel node is specified.

sfcli Hardware getclusterinfo <options>

Description	Options
This command retrieves the hardware status and information for all Fibre Channel nodes, iSCSI nodes and drives in the cluster. This generally includes details about manufacturers, vendors, versions, and other associated hardware identification information.	<code>type</code> : Includes only a certain type of hardware information in the response. Possible values are: drives (list only drive information in the response), nodes (list only node information in the response), and all (include both drive and node information in the response). If this option is omitted, a type of “all” is assumed.

sfcli Hardware getconfig <options>

Description	Options
This command enables you to display the hardware configuration information for a node.	None

Hardware sensor commands

You can use the commands in this section to get hardware information from the sensors in your node. You can get detailed reporting of sensors for node fans, power supplies, and so on.

sfcli Sensors getipmiinfo <options>

Description	Options
This command enables you to display a detailed reporting of sensors (objects) for node fans, intake and exhaust temperatures, and power supplies that are monitored by the system.	None

sfcli Sensors getipmiconfig <options>

Description	Options
This command enables you to retrieve hardware sensor information from sensors that are in your node.	<p>chassistype: Displays information for each node chassis type. Valid values are:</p> <ul style="list-style-type: none"> all: (returns sensor information for each chassis type. {chassis type}: Returns sensor information for a specified chassis type.

Initiator commands

Initiators enable external clients access to volumes in a cluster, serving as the entry point for communication between clients and volumes. You can create and delete initiators, and give them

friendly aliases to simplify administration and volume access. When you add an initiator to a volume access group, that initiator enables access to all volumes in the group.

sfcli Initiators modify <options>

Description	Options
This command enables you to change the attributes of one or more existing initiators. You cannot change the name of an existing initiator. If you need to change the name of an initiator, delete it first and create a new one. If this command fails to change one of the initiators you specify, you get an error, and the command does not modify any initiators (no partial completion is possible).	<p>initiators: A list of objects containing characteristics of each initiator to modify. Values are:</p> <ul style="list-style-type: none">• initiatorID: (Required) The ID of the initiator to modify. (Integer)• alias: (Optional) A new friendly name to assign to the initiator. (String)• attributes: (Optional) A new set of JSON attributes to assign to the initiator. (JSON Object)• volumeAccessGroupID: (Optional) The ID of the volume access group to which the initiator should be added. If the initiator was previously in a different volume access group, it is removed from the old volume access group. If this key is present but null, the initiator is removed from its current volume access group, but not placed in any new volume access group. (Integer)

sfcli Initiators create <options>

Description	Options
This command enables you to create multiple new initiator IQNs or World Wide Port Names (WWPNs) and optionally assign them aliases and attributes. When you use this command to create new initiators, you can also add them to volume access groups. If this command fails to create one of the initiators you specify, you get an error, and the command does not create any initiators (no partial completion is possible).	<p>initiators: A list of objects containing characteristics of each new initiator. Values are: alias: (Optional) The friendly name to assign to this initiator. (String) attributes: (Optional) A set of JSON attributes to assign to this initiator. (JSON Object) volumeAccessGroupID: (Optional) The ID of the volume access group into to which this newly created initiator will be added. (Integer)</p> <ul style="list-style-type: none"> • name: (Required) The name of the initiator (IQN or WWPN) to create. (String) • alias: (Optional) The friendly name to assign to this initiator. (String) • attributes: (Optional) A set of JSON attributes to assign to the initiator. (JSON Object) • volumeAccessGroupID: (Optional) The ID of the volume access group to which this newly created initiator will be added. (Integer)

sfcli Initiators list <options>

Description	Options
This command enables you to list initiator IQNs or World Wide Port Names (WWPNs).	<p>startinitiatorid: The initiator ID at which to begin the listing. You can supply this option or the initiators option, but not both.</p> <p>limit: The maximum number of initiator objects to return.</p> <p>initiators: A list of initiator IDs to retrieve. You can provide a value for this parameter or the startinitiatorid option, but not both.</p>

sfcli Initiators delete <options>

Description	Options
This command enables you to delete one or more initiators from the system (and from any associated volumes or volume access groups). If this command fails to delete one of the initiators you specify, the system returns an error and does not delete any initiators (no partial completion is possible).	<p>initiators: An array of IDs of initiators to delete.</p>

sfcli Initiators removefromvolumeaccessgroup <options>

Description	Options
This command enables you to remove initiators from a specified volume access group.	<p>volumeaccessgroupid: The ID of the volume access group from which the initiators are removed.</p> <p>initiators: The list of initiators to remove from the volume access group.</p> <p>deleteorphaninitiators: Specifies whether to delete initiator objects after they are removed from a volume access group or not. Possible values are:</p> <ul style="list-style-type: none"> • true: Delete initiator objects after they are removed from a volume access group. • false: Do not delete initiator objects after they are removed from a volume access group.

sfcli Initiators addtovolumeaccessgroup <options>

Description	Options
This command enables you to add initiators to a specified volume access group.	<p>initiators: The list of initiators to add to the volume access group.</p> <p>volumeaccessgroupid: The ID of the volume access group to modify.</p>

Lightweight Directory Access Protocol (LDAP) commands

You can set up LDAP to enable secure directory-based login functionality to SolidFire storage. You can use the commands in this section to perform operations, such as enabling and disabling LDAP authentication.

sfcli LDAP addclusteradmin <options>

Description	Options
This command enables you to add a new LDAP cluster administrator user.	<p>username: The distinguished username for the new LDAP cluster admin.</p> <p>access: Controls which methods this cluster admin can use. For more details on the levels of access, see the “Access Control” appendix in the <i>SolidFire API Reference</i>.</p> <p>accepteula: Accept the End User License Agreement. Set to true to add a cluster administrator account to the system. If omitted or set to false, the method call fails.</p> <p>attributes: List of name-value pairs in JSON object format.</p>

sfcli LDAP getconfiguration <options>

Description	Options
This command enables you to get the currently active LDAP configuration on the cluster.	None

sfcli LDAP testauthentication <options>

Description	Options
This command enables you to validate the currently enabled LDAP authentication settings. If the configuration is correct, you get the group membership of the tested user.	<p>username: The username to be tested.</p> <p>password</p> <p>password: The password for the username to be tested.</p> <p>ldapconfiguration: An ldapConfiguration object to be tested. If specified, the API call tests the provided configuration even if LDAP authentication is disabled.</p>

sfcli LDAP disableauthentication <options>

Description	Options
This command enables you to disable LDAP authentication and remove all LDAP configuration settings. It does not remove any configured cluster admin accounts (user or group). However, those cluster admin accounts will no longer be able to log in.	None

sfcli LDAP enableauthentication <options>

Description	Options
<p>This command enables you to configure an LDAP directory connection to use for LDAP authentication to a cluster. Users that are members of the LDAP directory can then log in to the storage system using their LDAP credentials.</p>	<p>authtype: Identifies which user authentication method to use. Must be either DirectBind or SearchAndBind.</p> <p>groupsearchbasedn: The base DN of the tree to start the group search (will do a subtree search from here).</p> <p>groupsearchcustomfilter: For use with the CustomFilter search type, an LDAP filter to use to return the DNs of a users groups. The string can have placeholder text of %USERNAME% and %USERDN% to be replaced with their username and full userDN as needed.</p> <p>groupsearchtype: Controls the default group search filter used, and must be one of the following: NoGroups (no group support), ActiveDirectory (nested membership of all of a user's AD groups), MemberDN (memberDN style groups (single level)).</p> <p>searchbinddn: A fully qualified DN to log in with to perform an LDAP search for the user (needs read access to the LDAP directory).</p> <p>searchbindpassword: The password for the searchBindDN account used for searching.</p> <p>serveruris: A comma-separated list of LDAP server URIs (for example, "ldap://1.2.3.4" and ldaps://1.2.3.4:123").</p> <p>userdntemplate: A string that is used to form a fully qualified user DN. The string should have the placeholder text %USERNAME%, which is replaced with the username of the authenticating user.</p> <p>usersearchbasedn: The base DN of the tree to start the search (will do a subtree search from here).</p> <p>usersearchfilter: The LDAP filter to use. The string should have the placeholder text %USERNAME%, which is replaced with the username of the authenticating user. Example: (&(objectClass=person)(sAMAccountName=%USERNAME%)) will use the sAMAccountName field in Active Directory to match the username entered at cluster login.</p>

Logging session commands

You can use the commands in this section to perform operations, such as setting the duration for which the login authentication is valid for a session. After the specified duration elapses, you need to reenter login credentials to continue to access the cluster.

sfcli LoggingSession getremotelogginghosts <options>

Description	Options
This command enables you to retrieve the current list of log servers.	None

sfcli LoggingSession setremotelogginghosts <options>

Description	Options
This command enables you to configure remote logging from the nodes in the storage cluster to a centralized log server or servers. Remote logging is performed over TCP using the default port 514. This command does not add to the existing logging hosts. Rather, it replaces what currently exists with new values specified.	remotehosts: List of hosts to send log messages to.

sfcli LoggingSession setloginsessioninfo <options>

Description	Options
This command enables you to set the period of time that a session's login authentication is valid. After the log in period elapses without activity on the system, the authentication expires. New login credentials are required for continued access to the cluster after the timeout period has elapsed.	timeout: Cluster authentication expiration period. Formatted in HH:mm:ss. For example, 01:30:00, 00:90:00, and 00:00:5400 can be used to equal a 90 minute timeout period. The default value is 30 minutes.

sfcli LoggingSession getloginsessioninfo <options>

Description	Options
This command enables you to return the period of time a log in authentication session is valid for both login shells and the TUI.	None

Network commands

You can use the commands in this section to perform operations, such as getting information about the Fibre Channel and iSCSI sessions on your cluster.

sfcli Network listnodefibrechannelportinfo <options>

Description	Options
This command enables you to retrieve information about the Fibre Channel ports on a node. It is intended for use on individual nodes; userid and password authentication is required for access to individual Fibre Channel nodes.	None

sfcli Network listfibrechannelsessions <options>

Description	Options
This command enables you to retrieve information about the active Fibre Channel sessions on a cluster.	None

sfcli Network listfibrechannelportinfo <options>

Description	Options
This command enables you to retrieve information about the Fibre Channel ports on a node. It is intended for use on individual nodes; userid and password authentication is required for access to individual Fibre Channel nodes.	None

sfcli Network listiscsisessions <options>

Description	Options
This command enables you to return iSCSI information for volumes in the cluster.	None

sfcli Network listinterfaces <options>

Description	Options
This command enables you to retrieve information about each network interface on a node. It is intended for use on individual nodes; userid and password authentication is required for access to individual nodes.	None

Node commands

You can use the commands in this section to perform operations, such as adding nodes to a cluster, removing nodes from a cluster, setting the network configuration for a node, and so on.

sfcli Node add <options>

Description	Options
<p>This command enables you to add one or more new nodes to a cluster. When a node that is not configured starts up for the first time, you are prompted to configure the node. After you configure the node, it is registered as a “pending node” with the cluster.</p> <p>Note: It might take several seconds after adding a new node for it to start up and register its drives as available.</p>	<p><code>pendingnodes</code>: List of pending NodeIDs for the nodes to be added. You can obtain the list of pending nodes using the <code>sfcli Node listpending <options></code> command.</p>

sfcli Node remove <options>

Description	Options
<p>This command enables you to remove one or more nodes that should no longer participate in the cluster. Before removing a node, you must remove all drives the node contains. You cannot remove a node until the process to remove drives has completed and all data has been migrated away from the node.</p>	<p><code>pendingnodes</code>: List of NodeIDs for the nodes to be removed.</p>

After you remove a node, it registers itself as a pending node. You can add the node again or shut it down (shutting the node down removes it from the *Pending Node* list).

sfcli Node setnetworkconfig <options>

Description	Options
<p>This command enables you to set the network configuration for a node.</p>	<p><code>network</code>: An object containing node network settings to modify.</p>

Caution: Changing the “bond-mode” on a node can cause a temporary loss of network connectivity. Exercise caution when using this command.

sfcli Node setconfig <options>

Description	Options
<p>This command enables you to set all the configuration information for the node.</p>	<p><code>config</code>: Objects that you want changed for the cluster interface settings.</p>

Caution: Changing the “bond-mode” on a node can cause a temporary loss of network connectivity. Exercise caution when using this command.

sfcli Node listpending <options>

Description	Options
This command returns a list of the currently pending nodes in the system.	None

sfcli Node listpendingactive <options>

Description	Options
This command returns the list of nodes in the cluster that are currently in the PendingActive state, between the pending and active states. These are nodes that are currently being returned to the factory image.	None

sfcli Node listall <options>

Description	Options
This command enables you to retrieve a list of active and pending nodes in the cluster.	None

sfcli Node liststats <options>

Description	Options
This command enables you to view the high-level activity measurements for all nodes in a cluster.	None

sfcli Node listactive <options>

Description	Options
This command returns the list of currently active nodes that are in the cluster.	None

sfcli Node getorigin <options>

Description	Options
This command enables you to retrieve the origination certificate for where the node was built. It might return null if there is no origination certification.	None

sfcli Node getpendingoperation <options>

Description	Options
You can use this command to detect an operation on a node that is currently in progress. You can also use this to report back when an operation has completed.	None

sfcli Node getnetworkconfig <options>

Description	Options
This command enables you to display the network configuration information for a node.	None

sfcli Node getstats <options>

Description	Options
This command enables you to retrieve the high-level activity measurements for a single node.	nodeid: Specifies the node for which statistics are gathered.

sfcli Node getconfig <options>

Description	Options
This command enables you to retrieve all configuration information for a node.	None

sfcli Node getbootstrapconfig <options>

Description	Options
This command returns cluster and node information from the bootstrap configuration file. Use this command on an individual node before it has been joined with a cluster. You can use the information you get in the cluster configuration interface when you create a cluster.	None

Pairing commands

You can use the commands in the section to perform operations, such as complete the pairing between two volumes, remove the pairing between two volumes, and so on.

sfcli Pairing startcluster <options>

Description	Options
This command enables you to create an encoded key from a cluster that is used to pair with another cluster. You can use this key with the key you get when you run <code>sfcli Pairing completecluster <options></code> to establish a cluster pairing. You can pair a cluster with a maximum of four other clusters.	None

sfcli Pairing completecluster <options>

Description	Options
You can use this command with the <code>sfcli Pairing startcluster <options></code> command to complete the cluster pairing process.	<code>clusterpairingkey</code> : A string of characters that is returned from the <code>sfcli Pairing startcluster <options></code> command.

sfcli Pairing listclusterpairs <options>

Description	Options
This command enables you to list all the clusters that a cluster is paired with. It returns information about active and pending cluster pairings, such as statistics about the current pairing as well as the connectivity and latency (in milliseconds) of the cluster pairing.	None

sfcli Pairing removeclusterpair <options>

Description	Options
This command enables you to close the open connections between two paired clusters. Note: Before you remove a cluster pair, you must first remove all volume pairing to the clusters.	<code>clusterpairingid</code> : Unique identifier used to pair two clusters.

sfcli Pairing startvolume <options>

Description	Options
This command enables you to create an encoded key from a volume that is used to pair with another volume.	<p>volumeid: The ID of the volume on which to start the pairing process.</p> <p>mode: The mode of the volume on which to start the pairing process. The mode can only be set if the volume is the source volume. Possible values are: volumeid (ID of the volume on which to start the pairing process), mode (mode of the volume on which to start the pairing process). The mode can only be set if the volume is the source volume.</p> <p>Possible values for mode are:</p> <ul style="list-style-type: none"> • Async: (default if no mode parameter specified) Writes are acknowledged when they complete locally. The cluster does not wait for writes to be replicated to the target cluster. • Sync: Source acknowledges write when the data is stored locally and on the remote cluster. • SnapshotsOnly: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated.

sfcli Pairing completevolume <options>

Description	Options
This command enables you to complete the pairing of two volumes.	<p>volumeid: The ID of the volume on which to complete the pairing process.</p> <p>volumepairingkey: The key returned from the <code>sfcli Pairing startvolume <options></code> command.</p>

sfcli Pairing removevolumepair <options>

Description	Options
This command enables you to remove the remote pairing between two volumes. Use this command on both the source and target volumes that are paired together. When you remove the volume pairing information, data is no longer replicated to or from the volume.	<p>volumeid: The ID of the volume on which to stop the replication process.</p>

sfcli Pairing listactivepairedvolumes <options>

Description	Options
This command enables you to list all the active volumes paired with a volume. You get information about volumes with active and pending pairings.	None

sfcli Pairing modifyvolumepair <options>

Description	Options
This command enables you to pause or restart replication between a pair of volumes.	<p>volumeid: The ID of the volume to be modified.</p> <p>pausedmanual: Specifies whether to pause or restart volume replication process. Valid values are: true (pauses volume replication) or false (restarts volume replication). If you do not specify a value, no change in replication is performed.</p> <p>mode: Specifies the volume replication mode. Possible values are:</p> <ul style="list-style-type: none"> • Async: Writes are acknowledged when they complete locally. The cluster does not wait for writes to be replicated to the target cluster. • Sync: The source acknowledges the write when the data is stored locally and on the remote cluster. • SnapshotsOnly: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated.

Restart commands

You can use the commands in this section to perform operations, such as restarting a node, restarting the networking services on a node, and so on.

sfcli Restart services <options>

Description	Options
<p>This command enables you to restart the service on a node.</p> <p>Caution: Exercise caution when using this command, because it causes temporary node service interruption.</p>	<p>force: Required parameter to successfully restart service on a node.</p> <p>service: Service name to be restarted.</p> <p>action: Action to perform on the service (start, stop, restart).</p>

sfcli Restart networking <options>

Description	Options
<p>This command enables you to restart the networking services on a node.</p> <p>Caution: This command restarts all networking services on a node, causing temporary loss of networking connectivity. Exercise caution when using this command.</p>	<p>force: Required parameter to successfully restart the node.</p>

sfcli Restart resetnode <options>

Description	Options
<p>This command enables you to reset a node to the factory settings. All data, packages (software upgrades, and so on), configurations, and log files are deleted from the node when you use this command. However, network settings for the node are preserved during this operation. Nodes that are participating in a cluster cannot be reset to the factory settings. This command can only be used on nodes that are in an “Available” state. It cannot be used on nodes that are “Active” in a cluster, or in a “Pending” state.</p> <p>Caution: Exercise caution when using this command, because it clears any data that is on the node.</p> <p>Note: This method is available only through the per-node API endpoint 5.0 or later.</p>	<p>build: Specifies the URL to a remote Element software image to which the node will be reset.</p> <p>force: Required parameter to successfully reset the node.</p> <p>options: Specifications for running the reset operation.</p> <p>reboot: Specifies whether to reboot the node or not. Set to true, if you want to reboot the node.</p>

sfcli Restart shutdown <options>

Description	Options
<p>This command enables you to restart or shutdown a node that has not yet been added to a cluster. To use this command, log in to the MIP for the pending node, and run this command with either the <code>restart</code> or <code>halt</code> options.</p>	<p>nodes: List of NodeIDs for the nodes to be shut down.</p> <p>options: Specifies the action to take for the node shutdown. Possible values are:</p> <ul style="list-style-type: none"> <code>restart</code>: Restarts the node. <code>halt</code>: Shuts down the node.

Schedule commands

You can use the commands in this section to perform operations, such as create a point-in-time copy (snapshot) of a volume, specify a schedule for when to create a snapshot of the volume, and so on.

sfcli Schedule list <options>

Description	Options
This command enables you to retrieve information about all scheduled snapshots that have been created.	None

sfcli Schedule create <options>

Description	Options
<p>This command enables you to schedule an automatic snapshot of a volume at a defined interval. You can use the created snapshot later as a backup or rollback to ensure the data on a volume or group of volumes is consistent for the point in time in which the snapshot was created. If you schedule a snapshot to run at a time period that is not divisible by 5 minutes, the snapshot runs at the next time period that is divisible by 5 minutes. For example, if you schedule a snapshot to run at 12:42:00 UTC, it runs at 12:45:00 UTC.</p> <p>Note: You can create snapshots if cluster fullness is at stage 1, 2, or 3. You cannot create snapshots after cluster fullness reaches stage 4 or 5.</p>	<p>hours: If provided with minutes and days, indicates how much time is in between each snapshot. If provided with weekdays or monthdays, indicates the time at which a snapshot will occur.</p> <p>minutes: If provided with hours and days, indicates how much time is in between each snapshot. If provided with weekdays or monthdays, indicates the time at which a snapshot will occur. If unspecified, defaults to 0.</p> <p>days: Indicates the number of days between each snapshot.</p> <p>weekdays: Indicates the weekday on which the snapshot will occur.</p> <p>monthdays: Indicates the days of the month on which snapshots occur.</p> <p>haserror: Indicates if the schedule has errors or not.</p> <p>scheduleid: Unique ID of the schedule.</p> <p>paused: Indicates if the schedule is paused or not.</p> <p>recurring: Indicates if the schedule is recurring or not.</p> <p>tobedeleted: Indicates if the schedule is marked for deletion.</p> <p>runnextinterval: Indicates whether or not the schedule will run the next time the scheduler is active. When set to “true”, the schedule will run the next time the schedule is active, and then reset back to “false.”</p> <p>lastruntimestarted: Indicates the last time the schedule started in ISO 8601 date string. Valid values are “success” or “failed.”</p> <p>lastrunstatus: Indicates the status of the last scheduled snapshot. Valid values are “success” or “failed.”</p> <p>name: Specifies the unique name assigned to the schedule. This option is required.</p> <p>startingdate: Indicates the date on which your schedule runs for the first time. It is formatted in UTC time.</p>

sfcli Schedule get <options>

Description	Options
This command enables you to retrieve information about a scheduled snapshot. You can see information about a specific schedule if there are many snapshot schedules in the system. You also retrieve information about more than one schedule with this method by specifying the <code>scheduleid</code> option.	<code>scheduleid</code> : Specifies the unique ID of the schedule or multiple schedules to display.

sfcli Schedule modify <options>

Description	Options
<p>This command enables you to change the intervals at which a scheduled snapshot occurs. This allows for adjustment to the snapshot frequency and retention.</p>	<p>hours: If provided with minutes and days, indicates how much time is in between each snapshot. If provided with weekdays or monthdays, indicates the time at which a snapshot will occur.</p> <p>minutes: If provided with hours and days, indicates how much time is in between each snapshot. If provided with weekdays or monthdays, indicates the time at which a snapshot will occur. If unspecified, defaults to 0.</p> <p>days: Indicates the number of days between each snapshot.</p> <p>weekdays: Indicates the weekday on which the snapshot will occur.</p> <p>monthdays: Indicates the days of the month on which snapshots occur.</p> <p>haserror: Indicates if the schedule has errors or not.</p> <p>paused: Indicates if the schedule is paused or not.</p> <p>recurring: Indicates if the schedule is recurring or not.</p> <p>runnextinterval: Indicates whether or not the schedule will run the next time the scheduler is active. When set to “true”, the schedule will run the next time the schedule is active, and then reset back to “false.”</p> <p>scheduleid: Specifies the unique ID of the schedule.</p> <p>lastrunstatus: Indicates the status of the last scheduled snapshot. Valid values are: success or failed.</p> <p>lastruntimestarted: Indicates the last time the schedule started as an ISO 8601 date string. Valid values are: success or failed.</p> <p>name: Specifies the unique name assigned to the schedule. This option is required.</p> <p>startingdate: Indicates the date on which your schedule runs for the first time. It is formatted in UTC time.</p> <p>tobedeleted: Indicates if the schedule is marked for deletion.</p>

Service command

You can use the command in this section to get service information for the nodes and drives on your cluster.

sfcli Service list <options>

Description	Options
This command enables you to return the services information for nodes, drives, current software, and other services that are running on the cluster.	None

Snapshot commands

You can use the commands in this section to perform operations, such as create a point-in-time copy (snapshot) of a volume, change the attributes assigned to a snapshot, and so on.

sfcli Snapshot listgroup <options>

Description	Options
This command enables you to get information about all group snapshots that have been created.	<p>volumeid: An array of unique volume IDs to query. If you do not specify this, all group snapshots on the cluster are included.</p> <p>groupsnapshotid: Retrieves information for a specific group snapshot ID.</p>

sfcli Snapshot modifygroup <options>

Description	Options
This command enables you to change the attributes of a group of snapshots. You can also use it to enable snapshots created on the Read/Write (source) volume to be remotely replicated to a target SolidFire storage system.	<p>groupsnapshotid: Specifies the ID of the group of snapshots.</p> <p>expirationtime: Sets the time when the snapshot should be removed. If unspecified, the current time is used.</p> <p>enableremotereplication: Replicates the snapshot created to a remote cluster. Possible values are:</p> <ul style="list-style-type: none"> true: The snapshot is replicated to remote storage. false: Default; the snapshot is not replicated.

sfcli Snapshot modify <options>

Description	Options
This command enables you to change the attributes currently assigned to a snapshot. You can use it to enable snapshots created on the Read/Write (source) volume to be remotely replicated to a target SolidFire storage system.	<p>snapshotid: Specifies the ID of the snapshot.</p> <p>expirationtime: Sets the time when the snapshot should be removed.</p> <p>enableremotereplication: Replicates the snapshot created to a remote cluster. Possible values are:</p> <ul style="list-style-type: none"> • true: The snapshot is replicated to remote storage. • false: Default; the snapshot is not replicated.

sfcli Snapshot create <options>

Description	Options
<p>This command enables you to create a point-in-time copy of a volume. You can create a snapshot from any volume or from an existing snapshot. If you do not provide a SnapshotID, a snapshot is created from the volume's active branch. If the volume from which the snapshot is created is being replicated to a remote cluster, the snapshot can also be replicated to the same target. Use the enableRemoteReplication option to enable snapshot replication.</p> <p>Note: Creating a snapshot is allowed if cluster fullness is at stage 2 or 3. Snapshots are not created when cluster fullness is at stage 4 or 5.</p>	<p>volumeid: Specifies the unique ID of the volume image from which to copy.</p> <p>snapshotid: Specifies the unique ID of a snapshot from which the new snapshot is made. The snapshotID passed must be a snapshot on the given volume.</p> <p>enableremotereplication: Replicates the snapshot created to a remote cluster. Possible values are:</p> <ul style="list-style-type: none"> • true: The snapshot is replicated to remote storage. • false: Default; the snapshot is not replicated. <p>name: Specifies a name for the snapshot. If unspecified, the date and time the snapshot was taken is used.</p> <p>retention: Specifies the amount of time for which the snapshot is retained. The format is HH:mm:ss.</p> <p>attributes: List of name-value pairs in JSON object format.</p>

sfcli Snapshot list <options>

Description	Options
This command enables you to return the attributes of each snapshot taken on the volume.	<p>volumeid: Retrieves snapshots for a volume. If volumeID is not provided, all snapshots for all volumes are returned.</p> <p>snapshotid: Retrieves information for a specific snapshot ID.</p>

sfcli Snapshot deletegroup <options>

Description	Options
This command enables you to delete a group snapshot. You can use the <code>saveMembers</code> option to preserve all the snapshots that were made for the volumes in the group, but the group association is removed.	<p><code>groupsnapshotid</code>: Specifies the unique ID of the group snapshot.</p> <p><code>savemembers</code>: Specifies whether to preserve snapshots or delete them. Valid values are:</p> <ul style="list-style-type: none"> <code>true</code>: Snapshots are preserved, but group association is removed. <code>false</code>: The group and snapshots are deleted.

sfcli Snapshot rollbacktogroup <options>

Description	Options
<p>This command enables you to roll back all individual volumes in a snapshot group to each volume's individual snapshot.</p> <p>Note: Rolling back to a group snapshot creates a temporary snapshot of each volume within the group snapshot. Snapshots are allowed if cluster fullness is at stage 2 or 3. Snapshots are not created when cluster fullness is at stage 4 or 5.</p>	<p><code>groupsnapshotid</code>: Specifies the unique ID of the group snapshot.</p> <p><code>savecurrentstate</code>: Specifies whether to save an active volume image or delete it. Valid values are: <code>true</code> (the previous active volume image is kept) or <code>false</code> (default; the previous active volume image is deleted).</p> <p><code>name</code>: Name for the group snapshot of the volume's current state that is created if <code>saveCurrentState</code> is set to <code>true</code>. If you do not give a name, the name of the snapshots (group and individual volume) are set to a timestamp of the time that the rollback occurred.</p> <p><code>attributes</code>: List of name-value pairs in JSON object format.</p>

sfcli Snapshot rollbackto <options>

Description	Options
<p>This command enables you to make an existing snapshot of the “active” volume image. This method creates a new snapshot from an existing snapshot. The new snapshot becomes “active” and the existing snapshot is preserved until you delete it. The previously “active” snapshot is deleted unless you set the <code>saveCurrentState</code> option to <code>true</code>.</p> <p>Note: Creating a snapshot is allowed if cluster fullness is at stage 2 or 3. Snapshots are not created when cluster fullness is at stage 4 or 5.</p>	<p><code>volumeid</code>: VolumeID for the volume.</p> <p><code>snapshotid</code>: ID of a previously created snapshot on the given volume.</p> <p><code>savecurrentstate</code>: Specifies whether to save an active volume image or delete it. Valid values are:</p> <ul style="list-style-type: none"> <code>true</code>: The previous active volume image is kept. <code>false</code>: Default; the previous active volume image is deleted. <p><code>name</code>: Name for the snapshot. If no name is given, then the name of the snapshot being rolled back to is used with “-copy” appended to the end of the name.</p> <p><code>attributes</code>: List of name-value pairs in JSON object format.</p>

sfcli Snapshot creategroup <options>

Description	Options
<p>This command enables you to create a point-in-time copy of a group of volumes. You can use this snapshot later as a backup or rollback to ensure the data on the group of volumes is consistent for the point in time that you created the snapshot.</p> <p>Note: Creating a group snapshot is allowed if cluster fullness is at stage 2 or 3. Snapshots are not created when cluster fullness is at stage 4 or 5.</p>	<p><code>volumes</code>: Unique ID of the volume image from which to copy.</p> <p><code>name</code>: Name for the group snapshot. If no name is given, then the date and time the group snapshot was taken is used.</p> <p><code>retention</code>: Specifies the amount of time for which the snapshots are retained. The format is HH:mm:ss.</p> <p><code>attributes</code>: List of name-value pairs in JSON object format.</p> <p><code>enableremotereplication</code>: Replicates the snapshot created to remote storage. Possible values are:</p> <ul style="list-style-type: none"> <code>true</code>: The snapshot is replicated to remote storage. <code>false</code>: Default; the snapshot is not replicated.

sfcli Snapshot delete <options>

Description	Options
This command enables you to delete a snapshot. A snapshot that is currently the “active” snapshot cannot be deleted. You must rollback and make another snapshot “active” before the current snapshot can be deleted.	snapshotid: The ID of the snapshot to be deleted.

Storage container commands

Storage containers are logical constructs that map to SolidFire accounts. You can use storage containers for reporting and resource allocation. This section includes the commands that you can use in CLI Tools to perform operations on virtual volume storage containers.

sfcli StorageContainers modifystoragecontainer <options>

Description	Options
This command enables you to make changes to an existing virtual volume storage container.	storagecontainerid: The unique ID of the virtual volume storage container to modify. initiatorsecret: The new secret for CHAP authentication for the initiator. targetsecret: The new secret for CHAP authentication for the target.

sfcli StorageContainers list <options>

Description	Options
This command enables you to retrieve information about all virtual volume storage containers known to the system.	storagecontainerids: A list of storage container IDs for which to retrieve information. If you omit this option, the method returns information about all storage containers in the system.

sfcli StorageContainers getstoragecontainerefficiency <options>

Description	Options
This command enables you to retrieve efficiency information about a virtual volume storage container.	storagecontainerid: The ID of the storage container for which to retrieve efficiency information.

sfcli StorageContainers createstoragecontainer <options>

Description	Options
This command enables you to create a Virtual Volume (VVol) storage container. Storage containers can only be associated with virtual volumes. You need at least one storage container to use the Virtual Volumes feature.	<p>name: The name of the storage container. Follows SolidFire account naming restrictions.</p> <p>initiatorsecret: The secret for CHAP authentication for the initiator.</p> <p>accountid: Non-storage container account that will become a storage container.</p> <p>targetsecret: The secret for CHAP authentication for the target.</p>

sfcli StorageContainers delete <options>

Description	Options
This command enables you to remove up to 2000 Virtual Volume (VVol) storage containers from the system at one time. The storage containers you remove must not contain any VVols.	storagecontainerids: A list of IDs of the storage containers to delete. You can specify up to 2000 IDs in the list.

Test commands

This section describes the test commands that you can use in CLI Tools.

sfcli Test list <options>

Description	Options
You can use this command to return the tests that are available to run on a node.	None

sfcli Test ping <options>

Description	Options
You can use this command to validate the connection to all the nodes in a cluster on both 1G and 10G interfaces by using ICMP packets. The test uses the appropriate MTU sizes for each packet based on the MTU settings in the network configuration.	<p>attempts: Specifies the number of times the system should repeat the test ping. The default value is 5.</p> <p>hosts: Specifies a comma-separated list of addresses or hostnames of devices to ping.</p> <p>totaltimeoutsec: Specifies the length of time the ping should wait for a system response before issuing the next ping attempt or ending the process.</p> <p>packetssize: Specifies the number of bytes to send in the ICMP packet that is sent to each IP. The number must be less than the maximum MTU specified in the network configuration.</p> <p>pingtimeoutmsec: Specifies the number of milliseconds to wait for each individual ping response. The default value is 500 ms.</p> <p>prohibitfragmentation: Specifies that the Do not Fragment (DF) flag is enabled for the ICMP packets.</p>

sfcli Test connectmvip <options>

Description	Options
This command enables you to test the management connection to the cluster. The test pings the MVIP and executes a simple API method to verify connectivity.	<p>mvip: If specified, tests the management connection of a different MVIP. You do not need to use this value when testing the connection to the target cluster. This parameter is optional.</p>

sfcli Test listutilities <options>

Description	Options
You can use this command to return the operations that are available to run on a node.	None

sfcli Test connectensemble <options>

Description	Options
The command enables you to verify connectivity with a specified database ensemble. By default, it uses the ensemble for the cluster that the node is associated with. Alternatively, you can provide a different ensemble to test connectivity with.	<p>ensemble: Uses a comma-separated list of ensemble node cluster IP addresses to test connectivity. This parameter is optional.</p>

sfcli Test connectsvip <options>

Description	Options
The command enables you to test the storage connection to the cluster. The test pings the SVIP using ICMP packets, and when successful, connects as an iSCSI initiator.	<i>svip</i> : If specified, tests the storage connection of a different SVIP. You do not need to use this value when testing the connection to the target cluster. This parameter is optional.

Virtual network commands

Virtual networking in SolidFire storage enables traffic between multiple clients that are on separate logical networks to be connected to one cluster. You can use the commands in this section to add a

new virtual network to a cluster, changes the attributes of an existing virtual network, remove a virtual network you added, and so on.

sfcli VirtualNetwork add <options>

Description	Options
<p>This command enables you to add a new virtual network to a cluster configuration. When you add a virtual network, an interface for each node is created and each interface will require a virtual network IP address. The number of IP addresses you specify must be equal to or greater than the number of nodes in the cluster. The system bulk provisions virtual network addresses and assigns them to individual nodes automatically. You do not need to assign virtual network addresses to nodes manually.</p> <p>Note: To make changes to an existing virtual network, use the <code>sfcli VirtualNetwork modify options</code> command.</p>	<p><code>virtualnetworktag</code>: A unique virtual network (VLAN) tag. Supported values are 1 through 4094. The number zero (0) is not supported.</p> <p><code>name</code>: Name for the new virtual network.</p> <p><code>addressblocks</code>: Unique range of IP addresses to include in the virtual network. Attributes for this option are:</p> <ul style="list-style-type: none"> <code>start</code>: The start of the IP address range. (String) <code>size</code>: The number of IP addresses to include in the block. (Integer) <p><code>attributes</code>: List of name-value pairs in JSON object format.</p> <p><code>netmask</code>: Unique network mask for the virtual network being created.</p> <p><code>svip</code>: Unique storage IP address for the virtual network being created.</p> <p><code>start</code>: Start of the IP address range.</p> <p><code>size</code>: Number of IP addresses to include in the block.</p> <p><code>available</code>: Number of available blocks.</p> <p><code>gateway</code>: The IP address of a gateway of the virtual network. This option is only valid if the <code>namespace</code> option is set to true.</p> <p><code>namespace</code>: When set to true, enables the Routable Storage VLANs functionality by creating and configuring a namespace and the virtual network contained by it.</p>

sfcli VirtualNetwork list <options>

Description	Options
This command enables you to list all configured virtual networks for the cluster. You can use this to verify the virtual network settings in the cluster. There are no required options. However, to filter the results, you can pass one or more <code>virtualnetworkid</code> or <code>virtualnetworktag</code> values.	<code>virtualnetworkid</code> : Network ID to filter the list for a single virtual network. <code>virtualnetworktag</code> : Network tag to filter the list for a single virtual network. <code>virtualnetworkids</code> : Network IDs to include in the list. <code>virtualnetworktags</code> : Network tag to include in the list.

sfcli VirtualNetwork remove <options>

Description	Options
<p>This command enables you to remove a previously added virtual network.</p> <p>Note: You must specify either the <code>virtualnetworkid</code> or the <code>virtualnetworktag</code> option, but not both.</p>	<code>virtualnetworkid</code> : Network ID that identifies the virtual network to remove. <code>virtualnetworktag</code> : Network tag that identifies the virtual network to remove.

sfcli VirtualNetwork modify <options>

Description	Options
<p>This command enables you to change the attributes of an existing virtual network. It enables you to add or remove address blocks, change the netmask, or modify the name or description of the virtual network. You can also use it to enable or disable namespaces, as well as add or remove a gateway if namespaces are enabled on the virtual network.</p> <p>Note: You must specify either the <code>virtualnetworkid</code> or the <code>virtualnetworktag</code> option, but not both.</p> <p>Caution: Enabling or disabling the Routable Storage VLANs functionality for an existing virtual network by changing the <code>namespace</code> option disrupts any traffic handled by the virtual network. NetApp strongly recommends changing the <code>namespace</code> option only during a scheduled maintenance window.</p>	<p><code>virtualnetworkid</code>: The unique identifier of the virtual network to modify. This is the virtual network ID assigned by the cluster.</p> <p><code>virtualnetworktag</code>: The network tag that identifies the virtual network to modify.</p> <p><code>name</code>: Name for the new virtual network.</p> <p><code>addressblocks</code>: The new addressBlock to set for this virtual network. This might contain new address blocks to add to the existing object or omit unused address blocks that need to be removed. Alternatively, you can extend or reduce the size of existing address blocks. You can only increase the size of the starting addressBlocks for a virtual network object; you can never decrease it. Attributes for this option are:</p> <ul style="list-style-type: none"> <code>start</code>: The start of the IP address range. (String) <code>size</code>: The number of IP addresses to include in the block. (Integer) <p><code>attributes</code>: List of name-value pairs in JSON object format.</p> <p><code>netmask</code>: New network mask for the virtual network.</p> <p><code>svip</code>: The storage virtual IP address for this virtual network. The <code>svip</code> for a virtual network cannot be changed. You must create a new virtual network to use a different <code>svip</code> address.</p> <p><code>gateway</code>: The IP address of a gateway of the virtual network. This option is only valid if the <code>namespace</code> option is set to true.</p> <p><code>namespace</code>: When set to true, enables Routable Storage VLANs functionality by recreating the virtual network and configuring a namespace to contain it. When set to false, disables the VRF functionality for the virtual network. Changing this value disrupts traffic running through this virtual network.</p>

Virtual volume commands

This section lists the commands that enable you to perform operations on the virtual volumes, such as creating virtual volumes, delete virtual volumes, and so on.

sfcli VirtualVolume modifyhost <options>

Description	Options
This command enables you to change an existing ESX host.	virtualvolumehostid: The GUID of the ESX host. clusterid: The GUID of the ESX cluster. visibleprotocolendpointids: A list of PEs the host is aware of. initiatornames: List of iSCSI initiator IQNs for the host. hostaddress: IP or DNS name for the host.

sfcli VirtualVolume gettaskupdate <options>

Description	Options
This command checks the status of a VVol Async Task.	virtualvolumetaskid: The UUID of the VVol Task.

sfcli VirtualVolume unbindallfromhost <options>

Description	Options
This command removes all VVol host binding.	None

sfcli VirtualVolume modifymetadata <options>

Description	Options
This command enables you to selectively modify the VVol metadata.	virtualvolumeid: VvolVolumeID for the volume to be modified.

sfcli VirtualVolume modifyvasaproviderinfo <options>

Description	Options
This command enables you to update the VASA provider information.	keystore: Signed SSL certificate for the VASA provider. vasaproviderid: UUID identifying the VASA provider.

sfcli VirtualVolume querymetadata <options>

Description	Options
This command returns a list of VVols matching a metadata query.	None

sfcli VirtualVolume listtasks <options>

Description	Options
This command returns a list of virtual volume tasks in the system.	<code>virtualvolumetaskids</code> : A list of virtual volume task IDs for which to retrieve information. If unspecified, the method returns information about all virtual volume tasks.

sfcli VirtualVolume listprotocolendpoints <options>

Description	Options
This command enables you to retrieve information about all protocol endpoints in the cluster.	<code>protocolendpointids</code> : A list of protocol endpoint IDs for which to retrieve information. If unspecified, the command returns information about all protocol endpoints.

sfcli VirtualVolume listvolumestatsby <options>

Description	Options
This command enables you to list volume statistics for any volumes in the system that are associated with virtual volumes. Statistics are cumulative from the creation of the volume.	<code>virtualvolumeids</code> : A list of one or more virtual volume IDs for which to retrieve information. If you specify this option, you get information about only these virtual volumes.

sfcli VirtualVolume create <options>

Description	Options
<p>This command creates a new (empty) virtual volume on the cluster. When the volume is created successfully, it is available for connection via PE.</p>	<p>name: Name of the virtual volume; between 1 to 64 characters in length. Not required to be unique, but it is recommended.</p> <p>totalsize: Total size of the volume, in bytes. Size is rounded up to the nearest 1MB size.</p> <p>storagecontainerid: UUID for the storage container of this volume.</p> <p>virtualvolumetype: VMW_TYPE value for this volume.</p> <p>totalsize: Total size of the volume, in bytes. Size is rounded up to the nearest 1MB size.</p> <p>qosminiops: Desired minimum 4KB IOPS to guarantee. The allowed IOPS will drop below this level only if all volumes have been capped at their minimum IOPS value and there is still insufficient performance capacity.</p> <p>qosmaxiops: Desired maximum 4KB IOPS allowed over an extended period of time.</p> <p>qosburstiops: Maximum “peak” 4KB IOPS allowed for short periods of time. Allows for bursts of I/O activity over the normal max IOPS value.</p> <p>qosbursttime: The length of time burst IOPS is allowed. The value returned is represented in time units of seconds.</p> <p>Note: This value is calculated by the system based on IOPS set for QoS.</p> <p>metadata: List of name-value pairs to save in the volume's metadata.</p>

sfcli VirtualVolume fastclone <options>

Description	Options
This command enables you to execute a VMware virtual volume fast clone.	<p>virtualvolumeid: The ID of the virtual volume to clone.</p> <p>name: The name for the newly created volume.</p> <p>qosminiops: Desired minimum 4KB IOPS to guarantee. The allowed IOPS will drop below this level only if all volumes have been capped at their minimum IOPS value and there is still insufficient performance capacity.</p> <p>qosmaxiops: Desired maximum 4KB IOPS allowed over an extended period of time.</p> <p>qosburstiops: Maximum “peak” 4KB IOPS allowed for short periods of time. Allows for bursts of I/O activity over the normal max IOPS value.</p> <p>qosbursttime: The length of time burst IOPS is allowed. The value returned is represented in time units of seconds.</p> <p>Note: This value is calculated by the system based on IOPS set for QoS.</p>

sfcli VirtualVolume canceltask <options>

Description	Options
This command attempts to cancel the VVol Async Task.	virtualvolumetaskid: The UUID of the VVol Task to cancel.

sfcli VirtualVolume getallocatedbitmap <options>

Description	Options
This command returns a b64-encoded block of data representing a bitmap, where non-zero bits indicate the allocation of a segment (LBA range) of the volume.	<p>virtualvolumeid: The ID of the virtual volume.</p> <p>segmentstart: Byte offset.</p> <p>segmentlength: Byte length adjusted to end on a chunk boundary.</p> <p>chunksize: Number of bytes represented by one bit in the bitmap.</p>

sfcli VirtualVolume getunsharedbitmap <options>

Description	Options
This command returns a b64-encoded block of data representing a bitmap, where non-zero bits indicate that data is not the same between two volumes for a common segment (LBA range) of the volumes.	<p>virtualvolumeid: The ID of the virtual volume.</p> <p>basevirtualvolumeid: The ID of the virtual volume to compare with.</p> <p>segmentstart: Byte offset.</p> <p>segmentlength: Byte length adjusted to end on a chunk boundary.</p> <p>chunksize: Number of bytes represented by one bit in the bitmap.</p>

sfcli VirtualVolume listhosts <options>

Description	Options
This command returns a list of all virtual volume hosts known to the cluster. A virtual volume host is a VMware ESX host that has initiated a session with the VASA API provider.	virtualvolumehostids: A list of virtual volume host IDs for which to retrieve information. If unspecified, you get information about all virtual volume hosts.

sfcli VirtualVolume rollback <options>

Description	Options
This command enables you to restore a VMware virtual volume snapshot.	<p>srcvirtualvolumeid: The ID of the virtual volume snapshot.</p> <p>dstvirtualvolumeid: The ID of the virtual volume to restore to.</p>

sfcli VirtualVolume getunsharedchunks <options>

Description	Options
This command scans a VVol segment and returns the number of chunks not shared between two volumes. It returns results in less than 30 seconds. If the specified VVol and the base VVol are not related, you get an error. If the offset or length combination you specify is invalid or out of range, you get an error.	<p>virtualvolumeid: The ID of the virtual volume.</p> <p>basevirtualvolumeid: The ID of the virtual volume to compare with.</p> <p>segmentstart: Start byte offset.</p> <p>segmentlength: Length of the scan segment in bytes.</p> <p>chunksize: Number of bytes represented by one bit in the bitmap.</p>

sfcli VirtualVolume getallocatedbitmap <options>

Description	Options
This command returns a b64-encoded block of data representing a bitmap, where non-zero bits indicate the allocation of a segment (LBA range) of the volume.	<p>virtualvolumeid: The ID of the virtual volume.</p> <p>basevirtualvolumeid: The ID of the virtual volume to compare with.</p> <p>segmentstart: Start byte offset.</p> <p>segmentlength: Length of the scan segment in bytes.</p> <p>chunksize: Number of bytes represented by one bit in the bitmap.</p>

sfcli VirtualVolume clone <options>

Description	Options
This command enables you to create a VMware virtual volume clone.	<p>virtualvolumeid: The ID of the virtual volume to clone.</p> <p>name: The name for the newly created volume.</p> <p>qosminiops: Desired minimum 4KB IOPS to guarantee. The allowed IOPS will drop below this level only if all volumes have been capped at their minimum IOPS value and there is still insufficient performance capacity.</p> <p>qosmaxiops: Desired maximum 4KB IOPS allowed over an extended period of time.</p> <p>qosburstiops: Maximum “peak” 4KB IOPS allowed for short periods of time. Allows for bursts of I/O activity over the normal max IOPS value.</p> <p>qosbursttime: The length of time burst IOPS is allowed. The value returned is represented in time units of seconds.</p> <p>Note: This value is calculated by the system based on IOPS set for QoS.</p>

sfcli VirtualVolume modify <options>

Description	Options
This command enables you to modify settings on an existing virtual volume.	<p>virtualvolumeid: VvolVolumeID for the volume to be modified.</p> <p>totalsize: New size of the volume in bytes. Size is rounded up to the nearest 1MiB size. Specify this option if you want to increase the size of a volume.</p> <p>qosminiops: Desired minimum 4KB IOPS to guarantee. The allowed IOPS will drop below this level only if all volumes have been capped at their minimum IOPS value and there is still insufficient performance capacity.</p> <p>qosmaxiops: Desired maximum 4KB IOPS allowed over an extended period of time.</p> <p>qosburstiops: Maximum “peak” 4KB IOPS allowed for short periods of time. Allows for bursts of I/O activity over the normal max IOPS value.</p> <p>qosbursttime: The length of time burst IOPS is allowed. The value returned is represented in time units of seconds.</p> <p>Note: This value is calculated by the system based on IOPS set for QoS.</p>

sfcli VirtualVolume preparevirtualsnapshot <options>

Description	Options
This command enables you to set up VMware virtual volume snapshot.	<p>virtualvolumeid: The ID of the virtual volume to clone.</p> <p>name: The name for the newly created volume.</p> <p>writablesnapshot</p>

sfcli VirtualVolume getfeaturestatus <options>

Description	Options
This command enables you to retrieve the status of a cluster feature.	<p>feature: Specifies the feature for which the status is returned. Valid value is: vvols (retrieve status for the NetApp SolidFire VVols cluster feature).</p>

sfcli VirtualVolume unbind <options>

Description	Options
This command removes the VVol Host binding.	unbindcontext

sfcli VirtualVolume createhost <options>

Description	Options
This command enables you to create a new ESX host.	<code>virtualvolumehostid</code> : The GUID of the ESX host. <code>clusterid</code> : The GUID of the ESX cluster. <code>visibleprotocolendpointids</code> : A list of PEs the host is aware of. <code>hostaddress</code> : IP or DNS name for the host.

sfcli VirtualVolume bind <options>

Description	Options
This command binds a virtual volume with a host.	<code>virtualvolumeids</code> : The UUID of the virtual volume to bind. <code>virtualvolumehostid</code> : The UUID of the ESX host. <code>bindcontext</code>

sfcli VirtualVolume list <options>

Description	Options
This command enables you to list the virtual volumes currently in the system. You can use this to list all virtual volumes or a subset of the virtual volumes in the system.	<p>details: Specifies the level of detail about each virtual volume that is returned. Possible values are:</p> <ul style="list-style-type: none"> • true: Include more details about each virtual volume in the response. • false: Include the standard level of detail about each virtual volume in the response. <p>limit: The maximum number of virtual volumes to list.</p> <p>recursive: Specifies whether to include information about the children of each virtual volume in the response. Possible values are:</p> <ul style="list-style-type: none"> • true: Include information about the children of each virtual volume in the response. • false: Do not include information about the children of each virtual volume in the response. <p>startvirtualvolumeid: The ID of the virtual volume at which to begin the list.</p> <p>virtualvolumeids: A list of virtual volume IDs for which to retrieve information. If you specify this, the method returns information about only these virtual volumes.</p>

sfcli VirtualVolume getvasaproviderinfo <options>

Description	Options
This command enables you to get the VASA provider information.	None

sfcli VirtualVolume snapshot <options>

Description	Options
This command enables you to take a VMware virtual volume snapshot.	<p>virtualvolumeid: The ID of the virtual volume to clone.</p> <p>timeout: Number of seconds to complete or fail.</p>

sfcli VirtualVolume listbindings <options>

Description	Options
This command returns a list of all virtual volumes in the cluster that are bound to protocol endpoints.	virtualvolumebindingids: A list of virtual volume binding IDs for which to retrieve information. If unspecified, you get information about all virtual volume bindings.

sfcli VirtualVolume getcount <options>

Description	Options
This command enables you to retrieve the number of virtual volumes currently in the system.	None

sfcli VirtualVolume enablefeature <options>

Description	Options
This command allows you to enable cluster features that are disabled by default.	feature: Indicates which feature to enable. Valid value is: vvols, which enables the NetApp SolidFire VVols cluster feature.

sfcli VirtualVolume delete <options>

Description	Options
This command marks an active volume for deletion. It is purged (permanently deleted) after the cleanup interval elapses. After making a request to delete a volume, any active iSCSI connections to the volume are immediately terminated and no further connections are allowed while the volume is in this state. It is not returned in target discovery requests. If a volume is marked for deletion, and it has a bulk volume read or bulk volume write operation in progress, the bulk volume operation is stopped. If the volume you delete is paired with a volume, replication between the paired volumes is suspended and no data is transferred to it or from it while in a deleted state.	virtualvolumes: The UUID of the volume to delete.

Volume commands

This section describes the volume commands that you can use in CLI Tools.

sfcli Volume getefficiency <options>

Description	Options
This command enables you to retrieve information about a volume. Only the volume you specify is used to compute the capacity.	volumeid: Specifies the volume for which capacity is computed.

sfcli Volume liststats <options>

Description	Options
This command returns high-level activity measurements for a volume, list of volumes, or all volumes (if you omit the <code>volumeids</code> option). Measurement values are cumulative from the creation of the volume.	<code>volumeids</code> : A list of volumes from which to retrieve activity information.

sfcli Volume removefromaccessgroup <options>

Description	Options
This command enables you to remove volumes from a volume access group.	<code>volumeaccessgroupid</code> : The ID of the volume access group to remove volumes from. <code>volumes</code> : Volumes to remove from the volume access group.

sfcli Volume addtoaccessgroup <options>

Description	Options
This command enables you to add volumes to a specified volume access group.	<code>volumeaccessgroupid</code> : The ID of the volume access group to which volumes are added. <code>volumes</code> : The list of volumes to add to the volume access group.

sfcli Volume liststatsbyaccount <options>

Description	Options
This command returns high-level activity measurements for every account. Values are summed from all the volumes owned by the account.	<code>accounts</code> : One or more account IDs by which to filter the result. <code>includevirtualvolumes</code> : Includes virtual volumes in the response by default. To exclude virtual volumes, set to false.

sfcli Volume startbulkwrite <options>

Description	Options
This command enables you to initialize a bulk volume write session on a specified volume. Only two bulk volume processes can run simultaneously on a volume. When you initialize the write session, data is written to a SolidFire storage volume from an external backup source. The external data is accessed by a web server running on an SF-series node. Communications and server interaction information for external data access is passed by a script running on the storage system.	<p>volumeid: The ID of the volume to be written to.</p> <p>format: The format of the volume data. It can be either of the following:</p> <ul style="list-style-type: none"> uncompressed: Every byte of the volume is returned without any compression. native: Opaque data is returned that is smaller and more efficiently stored and written on a subsequent bulk volume write. <p>script: The executable name of a script. If unspecified, the key and URL are necessary to access SF-series nodes. The script runs on the primary node and the key and URL are returned to the script, so the local web server can be contacted.</p> <p>scriptparameters: JSON parameters to pass to the script</p> <p>attributes: JSON attributes for the bulk volume job.</p>

sfcli Volume updatebulkstatus <options>

Description	Options
This command enables you to update the status of a bulk volume job that you started with <code>sfcli Volume startbulkread <options></code> or <code>sfcli Volume startbulkwrite <options></code> .	<p>key: The key assigned during initialization of a <code>StartBulkVolumeRead</code> or <code>StartBulkVolumeWrite</code> session.</p> <p>status: The status of the given bulk volume job. The system sets the status. Possible values are:</p> <ul style="list-style-type: none"> running: Jobs that are still active. complete: Jobs that are done. failed: Jobs that failed. <p>percentcomplete: The completed progress of the bulk volume job as a percentage value.</p> <p>message: The message returned indicating the status of the bulk volume job after the job is complete.</p> <p>attributes: JSON attributes; updates what is on the bulk volume job.</p>

sfcli Volume startbulkread <options>

Description	Options
<p>This command enables you to initialize a bulk volume read session on a specified volume. Only two bulk volume processes can run simultaneously on a volume.</p> <p>Note: This process creates a new snapshot if the ID of an existing snapshot is not provided. Snapshots can be created if cluster fullness is at stage 2 or 3. Snapshots are not created when cluster fullness is at stage 4 or 5.</p>	<p>volumeid: The ID of the volume to be read.</p> <p>format: The format of the volume data. It can be either of the following:</p> <ul style="list-style-type: none"> uncompressed: Every byte of the volume is returned without any compression. native: Opaque data is returned that is smaller and more efficiently stored and written on a subsequent bulk volume write. <p>snapshotid: The ID of a previously created snapshot used for bulk volume reads. If no ID is entered, a snapshot of the current active volume image is made.</p> <p>script: The executable name of a script. If unspecified, the key and URL are necessary to access SF-series nodes. The script is run on the primary node, and the key and URL are returned to the script, so the local web server can be contacted.</p> <p>scriptparameters: JSON parameters to pass to the script</p> <p>attributes: JSON attributes for the bulk volume job.</p>

sfcli Volume listdeleted <options>

Description	Options
<p>This command enables you to retrieve the list of volumes that have been marked for deletion and purged from the system.</p>	<p>includevirtualvolumes: Specifies that virtual volumes are included in the response by default. To exclude virtual volumes, set to false.</p>

sfcli Volume purgeddeleted <options>

Description	Options
<p>This command immediately and permanently purges a volume that has been deleted. You must delete a volume before it can be purged. Volumes are purged automatically after a period of time, so you might not need to use this method.</p>	<p>volumeid: The ID of the volume to be purged.</p>

sfcli Volume liststatsby <options>

Description	Options
This command returns high-level activity measurements for every volume, by volume. Values are cumulative from the creation of the volume.	<code>includevirtualvolumes</code> : Specifies that virtual volumes are included in the response by default. To exclude virtual volumes, set to false.

sfcli Volume create <options>

Description	Options
This command enables you to create a new (empty) volume on the cluster. As soon as the volume is created, the volume is available for connection via iSCSI.	<p><code>name</code>: The name of the volume access group (might be user specified). Not required to be unique, but recommended. Might be 1 to 64 characters in length.</p> <p><code>accountid</code>: AccountID for the owner of this volume.</p> <p><code>totalsize</code>: Total size of the volume, in bytes. Size is rounded up to the nearest 1MB size.</p> <p><code>enable512e</code>: Specifies whether 512e emulation is enabled or not. Possible values are:</p> <ul style="list-style-type: none"> <code>true</code>: The volume provides 512 byte sector emulation. <code>false</code>: 512e emulation is not enabled. <p><code>qosminiops</code>: Desired minimum 4KB IOPS to guarantee. The allowed IOPS will drop below this level only if all volumes have been capped at their minimum IOPS value and there is still insufficient performance capacity.</p> <p><code>qosmaxiops</code>: Desired maximum 4KB IOPS allowed over an extended period of time.</p> <p><code>qosburstiops</code>: Maximum “peak” 4KB IOPS allowed for short periods of time. Allows for bursts of I/O activity over the normal max IOPS value.</p> <p><code>qosbursttime</code>: The length of time burst IOPS is allowed. The value returned is represented in time units of seconds.</p> <p>Note: This value is calculated by the system based on IOPS set for QoS.</p> <p><code>attributes</code>: The list of name pairs in JSON object format. Total attribute size must be less than 1000B, or 1KB, including JSON formatting characters.</p>

sfcli Volume cancelclone <options>

Description	Options
This command enables you to stop an ongoing volume clone or volume copy process. When you cancel a group clone operation, the system completes and removes the operation's associated asyncHandle.	cloneid: The cloneID for the ongoing clone process.

sfcli Volume getdefaultqos <options>

Description	Options
This command enables you to retrieve the default QoS values for a newly created volume.	None

sfcli Volume getasyncreresult <options>

Description	Options
This command enables you to retrieve the result of asynchronous method calls. Some method calls require some time to run, and might not be finished when the system sends the initial response.	asynchandle: A value that was returned from the original asynchronous method call. keepresult: If true, GetAsyncResult does not remove the asynchronous result upon returning it, enabling future queries to that asyncHandle.

sfcli Volume listasyncreresults <options>

Description	Options
This command lists the results of all currently running and completed asynchronous methods on the system.	asyncreresulttypes: An optional list of types of results. You can use this list to restrict the results to only these types of operations. Possible values are: <ul style="list-style-type: none"> • BulkVolume: Copy operations between volumes, such as backups or restores. • Clone: Volume cloning operations. • DriveRemoval: Operations involving the system copying data from a drive in preparation to remove it from the cluster. • RtfiPendingNode: Operations involving the system installing compatible software on a node before adding it to the cluster.

sfcli Volume liststatsbyaccessgroup <options>

Description	Options
This command enables you to get total activity measurements for all of the volumes that are members of the specified volume access group(s).	<p>volumeaccessgroups: An array of VolumeAccessGroupIDs for which volume activity is returned. If omitted, statistics for all volume access groups are returned.</p> <p>includevirtualvolumes: Specifies that virtual volumes are included in the response by default. To exclude virtual volumes, set to false.</p>

sfcli Volume listbulkjobs <options>

Description	Options
This command enables you to retrieve information about each bulk volume read or write operation that is occurring in the system.	None

sfcli Volume clone <options>

Description	Options
<p>This command enables you to create a copy of a volume. This method is asynchronous and might take a variable amount of time to complete.</p> <p>Note: The initial attributes and QoS settings for the volume are inherited from the volume being cloned. You can change these settings with <code>sfcli Volume modify <options></code>.</p> <p>Note: Cloned volumes do not inherit volume access group memberships from the source volume.</p>	<p><code>volumeid</code>: VolumeID for the volume to be cloned.</p> <p><code>name</code>: The name of the new cloned volume. Might be 1 to 64 characters in length.</p> <p><code>newaccountid</code>: AccountID for the owner of the new volume. If unspecified, the accountID of the owner of the volume being cloned is used.</p> <p><code>newsize</code>: New size of the volume, in bytes. Might be greater or less than the size of the volume being cloned. If unspecified, the volume size is not changed. Size is rounded to the nearest 1MB.</p> <p><code>access</code>: Specifies the level of access allowed for the new volume. Possible values are:</p> <ul style="list-style-type: none"> • <code>readOnly</code>: Only read operations are allowed. • <code>readWrite</code>: Reads and writes are allowed. • <code>locked</code>: No reads or writes are allowed. If unspecified, the level of access of the volume being cloned is used. • <code>replicationTarget</code>: Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is locked. If a value is not specified, the access value does not change. <p><code>snapshotid</code>: ID of the snapshot that is used as the source of the clone. If no ID is provided, the current active volume is used.</p> <p><code>attributes</code>: List of name pairs in JSON object format.</p> <p><code>enable512e</code>: Specifies whether 512e emulation is enabled or not. Possible values are:</p> <ul style="list-style-type: none"> • <code>true</code>: The volume provides 512 byte sector emulation. • <code>false</code>: 512e emulation is not enabled.

sfcli Volume modify <options>

Description	Options
<p>This command enables you to modify settings on an existing volume. You can make modifications to one volume at a time and changes take place immediately. If you do not specify QoS values when you modify a volume, they remain the same as before the modification.</p> <p>Note: If you change the value of the <code>access</code> option to “locked” or “replicationTarget”, all existing iSCSI connections are terminated.</p>	<p><code>volumeid</code>: VolumeID for the volume to be modified.</p> <p><code>accountid</code>: AccountID to which the volume is reassigned. If unspecified, the previous account name is used.</p> <p><code>access</code>: Specifies the access allowed for the volume. Possible values are:</p> <ul style="list-style-type: none"> • <code>readOnly</code>: Only read operations are allowed. • <code>readWrite</code>: Reads and writes are allowed. • <code>locked</code>: No reads or writes are allowed. If not specified, the access value does not change. • <code>replicationTarget</code>: Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is locked. If a value is not specified, the access value does not change. <p><code>qosminiops</code>: Desired minimum 4KB IOPS to guarantee. The allowed IOPS will drop below this level only if all volumes have been capped at their minimum IOPS value and there is still insufficient performance capacity.</p> <p><code>qosmaxiops</code>: Desired maximum 4KB IOPS allowed over an extended period of time.</p> <p><code>qosburstiops</code>: Maximum “peak” 4KB IOPS allowed for short periods of time. Allows for bursts of I/O activity over the normal max IOPS value.</p> <p><code>qosbursttime</code>: The length of time burst IOPS is allowed. The value returned is represented in time units of seconds.</p> <p>Note: This value is calculated by the system based on IOPS set for QoS.</p> <p><code>totalsize</code>: New size of the volume in bytes. 1000000000 is equal to 1GB. Size is rounded up to the nearest 1MB. You can use this option only to increase the size of a volume.</p> <p><code>attributes</code>: List of name-value pairs in JSON object format.</p>

sfcli Volume restoredeleted <options>

Description	Options
This command marks a deleted volume as active again. This action makes the volume immediately available for iSCSI connection.	volumeid: VolumeID of the deleted volume to be restored.

sfcli Volume copy <options>

Description	Options
This command enables you to overwrite the data contents of an existing volume with the data contents of another volume (or snapshot). Attributes of the destination volume such as IQN, QoS settings, size, account, and volume access group membership are not changed. The destination volume must already exist and must be the same size as the source volume. NetApp recommends that clients unmount the destination volume before the volume copy operation begins. If the destination volume is modified during the copy operation, the changes will be lost. This method is asynchronous and might take a variable amount of time to complete.	volumeid: VolumeID of the volume to be read from. dstvolumeid: VolumeID of the volume to be overwritten. snapshotid: ID of the snapshot that is used as the source of the clone. If no ID is provided, the current active volume is used.

sfcli Volume listactive <options>

Description	Options
This command enables you to return the list of active volumes currently in the system. The list of volumes is returned sorted in VolumeID order and can be returned in multiple parts (pages).	startvolumeid: Starting VolumeID to return. If no volume exists with this VolumeID, the next volume by VolumeID order is used as the start of the list. limit: Maximum number of Volume Info objects to return. A value of 0 (zero) returns all volumes (unlimit). includevirtualvolumes: Specifies that virtual volumes are included in the response by default. To exclude virtual volumes, set to false.

sfcli Volume list <options>

Description	Options
<p>This command enables you to retrieve a list of volumes that are in a cluster. You can specify the volumes you want to return in the list by using the available parameters.</p>	<p>startvolumeid: Only volumes with an ID greater than or equal to this value are returned. Mutually exclusive with the volumeids option.</p> <p>limit: Specifies the maximum number of volume results that are returned. Mutually exclusive with the volumeids option.</p> <p>volumestatus: Only volumes with a status equal to the status value are returned. Possible values are:</p> <ul style="list-style-type: none"> • creating • snapshotting • active • deleted <p>accounts: Returns only the volumes owned by the accounts you specify here. Mutually exclusive with the volumeids option.</p> <p>ispaired: Returns volumes that are paired or not paired. Possible values are:</p> <ul style="list-style-type: none"> • true: Returns all paired volumes. • false: Returns all volumes that are not paired. <p>volumeids: A list of volume IDs. If you specify this option, other options operate only on this set of volumes. Mutually exclusive with the accounts, startvolumeid, and limit options.</p> <p>volumename: Only volume object information matching the volume name is returned.</p> <p>includevirtualvolumes: Specifies that virtual volumes are included in the response by default. To exclude virtual volumes, set to false.</p>

sfcli Volume clonemultiple <options>

Description	Options
<p>This command enables you to create a clone of a group of specified volumes. You can assign a consistent set of characteristics to a group of multiple volumes when they are cloned together.</p> <p>Note: Cloning multiple volumes is allowed if cluster fullness is at stage 2 or 3. Clones are not created when cluster fullness is at stage 4 or 5.</p>	<p>volumes: Unique ID for each volume to include in the clone. The volumes option includes additional options that you can specify. You must specify the volumeid option. The following options are not required:</p> <ul style="list-style-type: none"> access: Can be one of <code>readOnly</code>, <code>readWrite</code>, <code>locked</code>, or <code>replicationTarget</code>. attributes: List of name-value pairs in JSON object format. name: New name for the clone. newAccountID Account ID for the new volumes. newSize: New size Total size of the volume, in bytes. Size is rounded up to the nearest 1MB. <p>If you do not specify options, the values are inherited from the source volumes.</p> <p>access: New default access method for the new volumes if not overridden by information passed in the volume's array.</p> <p>groupsnapshotid: ID of the group snapshot to use as a basis for the clone.</p> <p>newaccountid: New account ID for the volumes if not overridden by information passed in the volumes array.</p>

sfcli Volume setdefaultqos <options>

Description	Options
<p>This command enables you to configure the default Quality of Service (QoS) values (measured in inputs and outputs per second, or IOPS) for a volume. For more information about QoS in a SolidFire cluster, see the <i>Element OS User Guide</i>.</p>	<p>miniops: The minimum number of sustained IOPS provided by the cluster to a volume.</p> <p>maxiops: The maximum number of sustained IOPS provided by the cluster to a volume.</p> <p>burstiops: The maximum number of IOPS allowed in a short burst scenario.</p>

sfcli Volume getstats <options>

Description	Options
<p>This command enables you to retrieve high-level activity measurements for a single volume. Values are cumulative from the creation of the volume.</p>	<p>volumeid: Specifies the volume for which statistics are gathered.</p>

sfcli Volume listforaccount <options>

Description	Options
This command returns the list of active and (pending) deleted volumes for an account.	<p>accountid: Returns all volumes owned by this AccountID.</p> <p>startvolumeid: The ID of the first volume to list. This can be useful for paging results. By default, this starts at the lowest VolumeID.</p> <p>limit: The maximum number of volumes to return from the API.</p> <p>includevirtualvolumes: Specifies that virtual volumes are included in the response by default. To exclude virtual volumes, set to false.</p>

sfcli Volume getcount <options>

Description	Options
This command enables you to retrieve the number of volumes currently in the system.	None

sfcli Volume cancelgroupclone <options>

Description	Options
This command enables you to stop an ongoing process for cloning multiple volumes. When you cancel a group clone operation, the system completes and removes the operation's associated asyncHandle.	groupcloneid: The cloneID for the ongoing clone process.

sfcli Volume delete <options>

Description	Options
This command marks an active volume for deletion. When marked, the volume is purged (permanently deleted) after the cleanup interval elapses. After making a request to delete a volume, any active iSCSI connections to the volume are immediately terminated and no further connections are allowed while the volume is in this state. A marked volume is not returned in target discovery requests. Any snapshots of a volume that has been marked for deletion are not affected. Snapshots are kept until the volume is purged from the system. If a volume is marked for deletion and has a bulk volume read or bulk volume write operation in progress, the bulk volume read or write operation is stopped. If the volume you delete is paired with a volume, replication between the paired volumes is suspended and no data is transferred to it or from it while in a deleted state.	volumeid: The ID of the volume to be deleted.

Volume access group commands

This section describes commands that you can use to create volume access groups, retrieve information about the volume access groups in the system, and so on.

sfcli VolumeAccessGroup create <options>

Description	Options
This command enables you to create a new volume access group. When you create the volume access group, you need to give it a name, and you can optionally enter initiators and volumes. After you create the group, you can add volumes and initiator IQNs. Any initiator IQN that you add to the volume access group is able to access any volume in the group without CHAP authentication.	<p>name: The name for this volume access group. Not required to be unique, but recommended.</p> <p>attributes: List of name-value pairs in JSON object format.</p> <p>initiators: List of initiators to include in the volume access group. If unspecified, the access group's configured initiators are not modified.</p> <p>virtualnetworkid: The ID of the SolidFire virtual network to associate the volume access group with.</p> <p>virtualnetworktags: The ID of the VLAN virtual network tag to associate the volume access group with.</p> <p>volumes: List of volumes to initially include in the volume access group. If unspecified, the access group's volumes are not modified.</p>

sfcli VolumeAccessGroup modify <options>

Description	Options
This command enables you to update initiators and add or remove volumes from a volume access group. If a specified initiator or volume is a duplicate of what currently exists, the volume access group is not modified. If you do not specify a value for volumes or initiators, the current list of initiators and volumes is not changed.	<p>volumeaccessgroupid: The ID of the volume access group to modify.</p> <p>name: The new name for this volume access group. Not required to be unique, but recommended.</p> <p>attributes: List of name-value pairs in JSON object format.</p> <p>initiators: List of initiators to include in the volume access group. If unspecified, the access group's configured initiators are not modified.</p> <p>virtualnetworkid: The ID of the SolidFire virtual network to associate the volume access group with.</p> <p>virtualnetworktags: The ID of the VLAN virtual network tag to associate the volume access group with.</p> <p>volumes: List of volumes to initially include in the volume access group. If unspecified, the access group's volumes are not modified.</p>

sfcli VolumeAccessGroup modifylunassignments <options>

Description	Options
This command enables you to define custom LUN assignments for specific volumes. It changes only LUN values specified in the <code>lunassignments</code> option in the volume access group. All other LUN assignments remain unchanged.	<p><code>volumeaccessgroupid</code>: Unique volume access group ID for which the LUN assignments will be modified.</p> <p><code>lunassignments</code>: The volume IDs with new assigned LUN values.</p>

LUN assignment values must be unique for volumes in a volume access group. You cannot define duplicate LUN values within a volume access group. However, you can use the same LUN values again in different volume access groups.

Note: Correct LUN values are 0 through 16383. The system generates an exception if you pass a LUN value outside of this range. None of the specified LUN assignments are modified if there is an exception.

Caution: If you change a LUN assignment for a volume with active I/O, the I/O can be disrupted. You might need to change the server configuration before changing volume LUN assignments.

sfcli VolumeAccessGroup list <options>

Description	Options
This command enables you to return information about the volume access groups that are currently in the system.	<p><code>startvolumeaccessgroupid</code>: The volume access group ID at which to begin the listing. If unspecified, there is no lower limit (implicitly 0).</p> <p><code>limit</code>: The maximum number of results to return. This can be useful for paging.</p>

sfcli VolumeAccessGroup getlunassignments <options>

Description	Options
This command enables you to retrieve details on LUN mappings of a specified volume access group.	<code>volumeaccessgroupid</code> : The unique volume access group ID used to return information.

sfcli VolumeAccessGroup getefficiency <options>

Description	Options
This command enables you to retrieve efficiency information about a volume access group. Only the volume access group you provide as the option for this command is used to compute the capacity.	<code>volumeaccessgroupid</code> : The volume access group for which capacity is computed.

sfcli VolumeAccessGroup delete <options>

Description	Options
This command enables you to delete a volume access group.	<code>volumeaccessgroupid</code> : The ID of the volume access group to be deleted.

Where to find additional information

You can use the resources in this section for additional information about NetApp SolidFire storage systems.

- For more information about deploying, configuring, and using your cluster, see the *NetApp SolidFire Element OS User Guide*.
- For information about the NetApp SolidFire API, see the *NetApp SolidFire Element OS API Reference Guide*.

Contacting NetApp SolidFire Active Support

You can contact NetApp SolidFire Active Support if you have any questions or comments about SolidFire documents or products in general.

Visit [NetApp SolidFire support](#) or email ng-SF-Support@netapp.com for help with NetApp SolidFire systems.

Copyright information

Copyright © 1994–2017 NetApp, Inc. All rights reserved. Printed in the U.S.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

Active IQ, AltaVault, Arch Design, ASUP, AutoSupport, Campaign Express, Clustered Data ONTAP, Customer Fitness, Data ONTAP, DataMotion, Element, Fitness, Flash Accel, Flash Cache, Flash Pool, FlexArray, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexVol, FPolicy, Fueled by SolidFire, GetSuccessful, Helix Design, LockVault, Manage ONTAP, MetroCluster, MultiStore, NetApp, NetApp Insight, OnCommand, ONTAP, ONTAPI, RAID DP, RAID-TEC, SANscreen, SANshare, SANtricity, SecureShare, Simplicity, Simulate ONTAP, Snap Creator, SnapCenter, SnapCopy, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, SolidFire, SolidFire Helix, StorageGRID, SyncMirror, Tech OnTap, Unbound Cloud, and WAFL and other names are trademarks or registered trademarks of NetApp, Inc., in the United States, and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. A current list of NetApp trademarks is available on the web.

<http://www.netapp.com/us/legal/netapptmlist.aspx>

How to send comments about documentation and receive update notifications

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email.

doccomments@netapp.com

To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277