

---

# **SoftLayer API Python Client Documentation**

*Release latest*

**SoftLayer Technologies, Inc.**

**Jan 29, 2020**



<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	What's Included . . . . .	3
1.2	Using Pip . . . . .	3
1.3	Debian/Ubuntu . . . . .	3
1.4	From Source . . . . .	3
<b>2</b>	<b>Configuration File</b>	<b>5</b>
<b>3</b>	<b>API Documentation</b>	<b>7</b>
3.1	Getting Started . . . . .	7
3.2	Managers . . . . .	8
3.2.1	SoftLayer.account . . . . .	8
3.2.2	SoftLayer.autoscale . . . . .	9
3.2.3	SoftLayer.block . . . . .	10
3.2.4	SoftLayer.cdn . . . . .	16
3.2.5	SoftLayer.dedicatedhost . . . . .	18
3.2.6	SoftLayer.dns . . . . .	20
3.2.7	SoftLayer.event_log . . . . .	23
3.2.8	SoftLayer.file . . . . .	24
3.2.9	SoftLayer.firewall . . . . .	29
3.2.10	SoftLayer.hardware . . . . .	30
3.2.11	SoftLayer.image . . . . .	35
3.2.12	SoftLayer.ipsec . . . . .	37
3.2.13	SoftLayer.load_balancer . . . . .	40
3.2.14	SoftLayer.metadata . . . . .	43
3.2.15	SoftLayer.network . . . . .	44
3.2.16	SoftLayer.ordering . . . . .	48
3.2.17	SoftLayer.sshkey . . . . .	54
3.2.18	SoftLayer.ssl . . . . .	55
3.2.19	SoftLayer.ticket . . . . .	56
3.2.20	SoftLayer.vs . . . . .	58
3.2.21	SoftLayer.vs_capacity . . . . .	66
3.2.22	SoftLayer.vs_placement . . . . .	68
3.3	Making API Calls . . . . .	69
3.4	Debugging . . . . .	71
3.5	API Reference . . . . .	71
3.5.1	SoftLayer Python API Client . . . . .	71

<b>4</b>	<b>Command-line Interface</b>	<b>75</b>
4.1	Configuration Setup . . . . .	75
4.2	Usage Examples . . . . .	76
<b>5</b>	<b>Debugging</b>	<b>79</b>
<b>6</b>	<b>Command Directory</b>	<b>81</b>
6.1	Account Commands . . . . .	81
6.1.1	account summary . . . . .	81
6.1.2	account events . . . . .	81
6.1.3	account event-detail . . . . .	81
6.1.4	account invoices . . . . .	82
6.1.5	account invoice-detail . . . . .	82
6.2	Autoscale Commands . . . . .	82
6.2.1	autoscale list . . . . .	83
6.2.2	autoscale detail . . . . .	83
6.2.3	autoscale scale . . . . .	83
6.2.4	autoscale logs . . . . .	84
6.2.5	autoscale tag . . . . .	84
6.2.6	autoscale edit . . . . .	84
6.3	Block Commands . . . . .	85
6.3.1	block access-authorize . . . . .	85
6.3.2	block access-list . . . . .	85
6.3.3	block access-revoke . . . . .	86
6.3.4	block access-password . . . . .	86
6.3.5	block replica-failback . . . . .	87
6.3.6	block replica-failover . . . . .	87
6.3.7	block replica-order . . . . .	87
6.3.8	block replica-partners . . . . .	88
6.3.9	block replica-locations . . . . .	88
6.3.10	block snapshot-cancel . . . . .	89
6.3.11	block snapshot-create . . . . .	89
6.3.12	block snapshot-delete . . . . .	90
6.3.13	block snapshot-disable . . . . .	90
6.3.14	block snapshot-enable . . . . .	90
6.3.15	block snapshot-schedule-list . . . . .	91
6.3.16	block snapshot-list . . . . .	91
6.3.17	block snapshot-order . . . . .	91
6.3.18	block snapshot-restore . . . . .	92
6.3.19	block volume-cancel . . . . .	92
6.3.20	block volume-count . . . . .	93
6.3.21	block volume-detail . . . . .	93
6.3.22	block volume-duplicate . . . . .	93
6.3.23	block volume-list . . . . .	94
6.3.24	block volume-modify . . . . .	94
6.3.25	block volume-order . . . . .	95
6.3.26	block volume-set-lun-id . . . . .	96
6.3.27	block volume-limits . . . . .	96
6.3.28	block subnets-list . . . . .	97
6.3.29	block subnets-assign . . . . .	97
6.3.30	block subnets-remove . . . . .	97
6.4	Interacting with CDN . . . . .	98
6.4.1	cdn detail . . . . .	98
6.4.2	cdn list . . . . .	98

6.4.3	cdn origin-add	98
6.4.4	cdn origin-list	99
6.4.5	cdn origin-remove	100
6.4.6	cdn purge	100
6.5	Call API	100
6.5.1	call-api	100
6.6	Shell	101
6.7	MetaData	101
6.7.1	metadata	101
6.8	Config	102
6.8.1	config setup	102
6.8.2	config show	102
6.8.3	setup	102
6.9	Dedicated Host Commands	103
6.9.1	dedicatedhost list	103
6.9.2	dedicatedhost create	103
6.9.3	dedicatedhost create-options	104
6.9.4	dedicatedhost detail	104
6.9.5	dedicatedhost cancel	105
6.9.6	dedicatedhost cancel-guests	105
6.9.7	dedicatedhost list-guests	105
6.10	DNS Management	106
6.10.1	dns import	106
6.10.2	dns record-add	106
6.10.3	dns record-edit	107
6.10.4	dns record-list	108
6.10.5	dns record-remove	108
6.10.6	dns zone-create	109
6.10.7	dns zone-delete	109
6.10.8	dns zone-list	109
6.10.9	dns zone-print	109
6.11	Event-Log Commands	109
6.11.1	event-log get	109
6.11.2	event-log types	110
6.12	File Commands	111
6.12.1	file access-authorize	111
6.12.2	file access-list	111
6.12.3	file access-revoke	112
6.12.4	file replica-failback	112
6.12.5	file replica-failover	112
6.12.6	file replica-order	113
6.12.7	file replica-partners	113
6.12.8	file replica-locations	114
6.12.9	file snapshot-cancel	114
6.12.10	file snapshot-create	114
6.12.11	file snapshot-delete	115
6.12.12	file snapshot-disable	115
6.12.13	file snapshot-enable	115
6.12.14	file snapshot-list	116
6.12.15	file snapshot-order	116
6.12.16	file snapshot-restore	117
6.12.17	file volume-cancel	117
6.12.18	file volume-count	117
6.12.19	file volume-detail	118

6.12.20	file volume-duplicate	118
6.12.21	file volume-list	119
6.12.22	file volume-modify	119
6.12.23	file volume-order	120
6.12.24	file volume-limits	121
6.12.25	file snapshot-schedule-list	121
6.13	Firewall Management	121
6.13.1	firewall add	121
6.13.2	firewall cancel	122
6.13.3	firewall detail	122
6.13.4	firewall edit	122
6.13.5	firewall list	122
6.14	Global IP Addresses	122
6.14.1	globalip assign	122
6.14.2	globalip cancel	123
6.14.3	globalip create	123
6.14.4	globalip list	123
6.14.5	globalip unassign	124
6.15	Interacting with Hardware	124
6.15.1	hw bandwidth	124
6.15.2	hw cancel-reasons	125
6.15.3	hw cancel	125
6.15.4	hw create-options	125
6.15.5	hw create	125
6.15.6	hw credentials	126
6.15.7	hw detail	126
6.15.8	hw edit	127
6.15.9	hw list	128
6.15.10	hw power-cycle	128
6.15.11	hw power-off	128
6.15.12	hw power-on	129
6.15.13	hw reboot	129
6.15.14	hw reload	129
6.15.15	hw rescue	130
6.15.16	hw reflash-firmware	130
6.15.17	hw update-firmware	130
6.15.18	hw toggle-ipmi	130
6.15.19	hw ready	131
6.15.20	hw dns-sync	131
6.16	Disk Image Commands	132
6.16.1	image delete	132
6.16.2	image detail	132
6.16.3	image edit	132
6.16.4	image list	133
6.16.5	image import	133
6.16.6	image export	134
6.17	Interacting with IPSEC Tunnels	134
6.17.1	ipsec list	134
6.17.2	ipsec detail	135
6.17.3	ipsec update	136
6.17.4	ipsec configure	137
6.17.5	ipsec subnet-add	137
6.17.6	ipsec subnet-remove	137
6.17.7	ipsec translation-add	138

6.17.8	ipsec translation-remove	138
6.17.9	ipsec translation-update	138
6.18	LoadBalancers	138
6.18.1	LBaaS Commands	139
6.18.2	NetScaler Commands	145
6.19	Object Storage Commands	145
6.19.1	object-storage accounts	145
6.19.2	object-storage endpoints	145
6.19.3	object-storage credential list	145
6.19.4	object-storage credential limit	146
6.19.5	object-storage credential delete	146
6.19.6	object-storage credential create	146
6.20	Ordering	146
6.20.1	order package-list	147
6.20.2	order package-locations	147
6.20.3	order category-list	148
6.20.4	order item-list	148
6.20.5	order preset-list	149
6.20.6	order place	150
6.20.7	order place <preset>	151
6.20.8	order place <Virtual Server>	151
6.21	Quotes	151
6.21.1	order quote	151
6.21.2	order quote-list	152
6.21.3	order quote-detail	152
6.21.4	order place-quote	153
6.22	Reports	154
6.22.1	summary	154
6.22.2	report bandwidth	154
6.23	Reverse Whois Commands	154
6.23.1	rwhois edit	154
6.23.2	rwhois show	155
6.24	Security Groups	155
6.24.1	securitygroup list	155
6.24.2	securitygroup detail	156
6.24.3	securitygroup create	156
6.24.4	securitygroup edit	156
6.24.5	securitygroup delete	157
6.24.6	securitygroup rule-list	157
6.24.7	securitygroup rule-add	157
6.24.8	securitygroup rule-edit	158
6.24.9	securitygroup rule-remove	159
6.24.10	securitygroup interface-list	159
6.24.11	securitygroup interface-add	160
6.24.12	securitygroup interface-remove	160
6.24.13	securitygroup event-log	160
6.25	SSH Keys	161
6.25.1	sshkey add	161
6.25.2	sshkey remove	161
6.25.3	sshkey edit	161
6.25.4	sshkey list	162
6.25.5	sshkey print	162
6.26	SSL Certificates	162
6.26.1	ssl add	162

6.26.2	ssl download	163
6.26.3	ssl edit	163
6.26.4	ssl list	164
6.26.5	ssl remove	164
6.27	Subnets	164
6.27.1	subnet cancel	164
6.27.2	subnet create	165
6.27.3	subnet detail	165
6.27.4	subnet list	166
6.27.5	subnet lookup	166
6.28	Support Tickets	167
6.28.1	ticket create	167
6.28.2	ticket detail	167
6.28.3	ticket list	167
6.28.4	ticket update	168
6.28.5	ticket upload	168
6.28.6	ticket subjects	168
6.28.7	ticket summary	169
6.28.8	ticket attach	169
6.28.9	ticket detach	169
6.29	Users	170
6.29.1	user list	170
6.29.2	user detail	170
6.29.3	user permissions	171
6.29.4	user edit-permissions	171
6.29.5	user edit-details	171
6.29.6	user create	172
6.29.7	user delete	172
6.30	VLANs	173
6.30.1	vlan detail	173
6.30.2	vlan list	173
6.31	Working with Virtual Servers	174
6.31.1	vs bandwidth	177
6.31.2	vs cancel	178
6.31.3	vs capture	178
6.31.4	vs create	178
6.31.5	vs create-options	180
6.31.6	vs dns-sync	180
6.31.7	vs edit	181
6.31.8	vs list	182
6.31.9	vs pause	182
6.31.10	vs power-on	183
6.31.11	vs power-off	183
6.31.12	vs resume	183
6.31.13	vs rescue	184
6.31.14	vs reboot	184
6.31.15	vs ready	184
6.31.16	vs upgrade	185
6.31.17	vs usage	185
6.31.18	Reserved Capacity	186
6.31.19	Placement Groups	187
<b>7</b>	<b>Contributing</b>	<b>191</b>
7.1	Contribution Guide	191



7.1.1	Code Organization . . . . .	191
7.1.2	Setting Up A Dev Environment . . . . .	191
7.1.3	Testing . . . . .	192
7.1.4	Documentation . . . . .	193
7.1.5	Style . . . . .	193
7.1.6	Contributing . . . . .	193
7.1.7	Developer Resources . . . . .	193
7.2	Command-Line Interface Developer Guide . . . . .	193
7.2.1	First Example . . . . .	193
7.2.2	Arguments . . . . .	194
7.2.3	Accessing the API . . . . .	195
7.2.4	Aborting execution . . . . .	196
7.2.5	Documenting Commands . . . . .	196
7.2.6	Architecture . . . . .	196
<b>8</b>	<b>External Links</b>	<b>197</b>
	<b>Python Module Index</b>	<b>199</b>
	<b>Index</b>	<b>201</b>



[API Docs](#) | [GitHub](#) | [Issues](#) | [Pull Requests](#) | [PyPI](#) |

This is the documentation to SoftLayer's Python API Bindings. These bindings use SoftLayer's [XML-RPC](#) interface in order to manage SoftLayer services.



### 1.1 What's Included

When you install `softlayer-python` you will get the following:

- a python package called 'SoftLayer' (casing is important) available in your python path.
- a command-line client placed in your system path named 'slcli'.

### 1.2 Using Pip

Install via pip:

```
$ pip install softlayer
```

### 1.3 Debian/Ubuntu

For Debian "jessie" (currently testing) and Ubuntu 14.04, official system packages are available. **These are typically a couple versions behind so it is recommended to install from pypi if problems are encountered.**

```
$ sudo apt-get install python-softlayer
```

### 1.4 From Source

The project is developed on GitHub, at <https://github.com/softlayer/softlayer-python>.

Install from source via pip (requires git):

```
$ pip install git+git://github.com/softlayer/softlayer-python.git
```

You can clone the public repository:

```
$ git clone git@github.com:softlayer/softlayer-python.git
```

Or, Download the [tarball](#):

```
$ curl -OL https://github.com/softlayer/softlayer-python/tarball/master
```

Or, download the [zipball](#):

```
$ curl -OL https://github.com/softlayer/softlayer-python/zipball/master
```

Once you have a copy of the source you can install it with one of the following commands:

```
$ python setup.py install
```

Or:

```
$ pip install .
```

For more information about working with the source, or contributing to the project, please see the [Contribution Guide](#).

---

## Configuration File

---

The SoftLayer API bindings load your settings from a number of different locations.

- Input directly into `SoftLayer.create_client_from_env(...)`
- Environment variables (`SL_USERNAME`, `SL_API_KEY`)
- Config file locations (`~/.softlayer`, `/etc/softlayer.conf`)
- Or argument (`-C/path/to/config` or `-config=/path/to/config`)

The configuration file is INI-based and requires the `softlayer` section to be present. The only required fields are `username` and `api_key`. You can optionally supply the `endpoint_url` as well. This file is created automatically by the `slcli setup` command detailed here: [Configuration Setup](#).

### *Config Example*

```
[softlayer]
username = username
api_key = oyVmeipYQCNrjVS4rF9bHWV7D75S6pa1fghF1384v7mwRCbHTfuJ8qRORIqoVnha
endpoint_url = https://api.softlayer.com/xmlrpc/v3/
timeout = 40
```

### *Cloud.ibm.com Config Example*

```
[softlayer]
username = apikey
api_key = 123cNyhzg45Ab6789ADyzwR_2LAagNVbySgY73tAQOz1
endpoint_url = https://api.softlayer.com/rest/v3.1/
timeout = 40
```





---

## API Documentation

---

This is the primary API client to make API calls. It deals with constructing and executing XML-RPC calls against the SoftLayer API. Below are some links that will help to use the SoftLayer API.

- [SoftLayer API Documentation](#)
- [Source on GitHub](#)

```
>>> import SoftLayer
>>> client = SoftLayer.create_client_from_env(username="username", api_key="api_key")
>>> resp = client.call('Account', 'getObject')
>>> resp['companyName']
'Your Company'
```

### 3.1 Getting Started

You can pass in your username and api\_key when creating a SoftLayer client instance. However, you can also set these in the environmental variables 'SL\_USERNAME' and 'SL\_API\_KEY'.

Creating a client instance by passing in the username/api\_key:

```
import SoftLayer
client = SoftLayer.create_client_from_env(username='YOUR_USERNAME', api_key='YOUR_API_
↪KEY')
```

Creating a client instance with environmental variables set:

```
$ export SL_USERNAME=YOUR_USERNAME
$ export SL_API_KEY=YOUR_API_KEY
$ python
>>> import SoftLayer
>>> client = SoftLayer.create_client_from_env()
```

Below is an example of creating a client instance with more options. This will create a client with the private API endpoint (only accessible from the SoftLayer private network) and a timeout of 4 minutes.

```
client = SoftLayer.create_client_from_env(username='YOUR_USERNAME',
                                         api_key='YOUR_API_KEY',
                                         endpoint_url=SoftLayer.API_PRIVATE_ENDPOINT,
                                         timeout=240)
```

## 3.2 Managers

For day-to-day operation, most users will find the managers to be the most convenient means for interacting with the API. Managers abstract a lot of the complexities of using the API into classes that provide a simpler interface to various services. These are higher-level interfaces to the SoftLayer API.

```
from SoftLayer import VSManger, Client
client = Client(...)
vs = VSManger(client)
vs.list_instances()
[...]
```

**Available managers:**

### 3.2.1 SoftLayer.account

Account manager

**license** MIT, see License for more details.

**class** `SoftLayer.managers.account.AccountManager` (*client*)

Common functions for getting information from the Account service

**Parameters** *client* (`SoftLayer.API.BaseClient`) – the client instance

**ack\_event** (*event\_id*)

Acknowledge an event. This mostly prevents it from appearing as a notification in the control portal.

**Parameters** *event\_id* (*int*) – Notification\_Occurrence\_Event ID you want to ack

**Returns** True on success, Exception otherwise.

**get\_billing\_items** (*identifier*)

Gets all topLevelBillingItems from a specific invoice

**Parameters** *identifier* (*int*) – Invoice Id

**Returns** Billing\_Invoice\_Item

**get\_event** (*event\_id*)

Gets details about a maintenance event

**Parameters** *event\_id* (*int*) – Notification\_Occurrence\_Event ID

**Returns** Notification\_Occurrence\_Event

**get\_invoices** (*limit=50, closed=False, get\_all=False*)

Gets an accounts invoices.

**Parameters**

- **limit** (*int*) – Number of invoices to get back in a single call.

- **closed** (*bool*) – If True, will also get CLOSED invoices
- **get\_all** (*bool*) – If True, will paginate through invoices until all have been retrieved.

**Returns** Billing\_Invoice

**get\_summary** ()

Gets some basic account information

**Returns** Account object

**get\_upcoming\_events** ()

Retrieves a list of Notification\_Occurrence\_Events that have not ended yet

**Returns** SoftLayer\_Notification\_Occurrence\_Event

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly ‘identifier’ can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

### 3.2.2 SoftLayer.autoscale

Autoscale manager

**license** MIT, see LICENSE for more details.

**class** SoftLayer.managers.autoscale.**AutoScaleManager** (*client*)

Manager for interacting with Autoscale instances.

**details** (*identifier, mask=None*)

Calls SoftLayer\_Scale\_Group::getObject()

**Parameters**

- **identifier** – SoftLayer\_Scale\_Group id
- **mask** – optional SoftLayer\_Scale\_Group objectMask

**edit** (*identifier, template*)

Calls SoftLayer\_Scale\_Group::editObject()

**Parameters**

- **identifier** – SoftLayer\_Scale\_Group id
- **template** – SoftLayer\_Scale\_Group

**get\_logs** (*identifier, mask=None, object\_filter=None*)

Calls SoftLayer\_Scale\_Group::getLogs()

**Parameters**

- **identifier** – SoftLayer\_Scale\_Group Id
- **mask** – optional SoftLayer\_Scale\_Group\_Log objectMask
- **object\_filter** – optional SoftLayer\_Scale\_Group\_Log objectFilter

**get\_policy** (*identifier, mask=None*)

Calls SoftLayer\_Scale\_Policy::getObject()

**Parameters**

- **identifier** – SoftLayer\_Scale\_Policy id
- **mask** – optional SoftLayer\_Scale\_Policy objectMask

**get\_virtual\_guests** (*identifier, mask=None*)  
Calls `SoftLayer_Scale_Group::getVirtualGuestMembers()`

**Parameters**

- **identifier** – SoftLayer\_Scale\_Group Id
- **mask** – optional SoftLayer\_Scale\_Member objectMask

**list** (*mask=None*)  
Calls `SoftLayer_Account::getScaleGroups()`

**Parameters** **mask** – optional SoftLayer\_Scale\_Group objectMask

**scale** (*identifier, amount*)  
Calls `SoftLayer_Scale_Group::scale()`

**Parameters**

- **identifier** – SoftLayer\_Scale\_Group Id
- **amount** – positive or negative number to scale the group by

**scale\_to** (*identifier, amount*)  
Calls `SoftLayer_Scale_Group::scaleTo()`

**Parameters**

- **identifier** – SoftLayer\_Scale\_Group Id
- **amount** – number to scale the group to.

### 3.2.3 SoftLayer.block

Block Storage Manager

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.block.BlockStorageManager` (*client*)  
Manages SoftLayer Block Storage volumes.

See product information here: <http://www.softlayer.com/block-storage>

**Parameters** **client** (`SoftLayer.API.BaseClient`) – the client instance

**assign\_subnets\_to\_acl** (*access\_id, subnet\_ids*)  
Assigns subnet records to ACL for the access host.

*access\_id* is the *host\_id* obtained by: `slcli block access-list <volume_id>`

**Parameters**

- **access\_id** (*integer*) – id of the access host
- **subnet\_ids** (*list*) – The ids of the subnets to be assigned

**Returns** Returns int array of assigned subnet ids

**authorize\_host\_to\_volume** (*volume\_id, hardware\_ids=None, virtual\_guest\_ids=None, ip\_address\_ids=None, \*\*kwargs*)  
Authorizes hosts to Block Storage Volumes

**Parameters**

- **volume\_id** – The Block volume to authorize hosts to
- **hardware\_ids** – A List of SoftLayer\_Hardware ids
- **virtual\_guest\_ids** – A List of SoftLayer\_Virtual\_Guest ids
- **ip\_address\_ids** – A List of SoftLayer\_Network\_Subnet\_IpAddress ids

**Returns** Returns an array of SoftLayer\_Network\_Storage\_Allowed\_Host objects which now have access to the given Block volume

**cancel\_block\_volume** (*volume\_id*, *reason*='No longer needed', *immediate*=False)

Cancels the given block storage volume.

**Parameters**

- **volume\_id** (*integer*) – The volume ID
- **reason** (*string*) – The reason for cancellation
- **immediate\_flag** (*boolean*) – Cancel immediately or on anniversary date

**cancel\_snapshot\_space** (*volume\_id*, *reason*='No longer needed', *immediate*=False)

Cancels snapshot space for a given volume.

**Parameters**

- **volume\_id** (*integer*) – The volume ID
- **reason** (*string*) – The reason for cancellation
- **immediate\_flag** (*boolean*) – Cancel immediately or on anniversary date

**create\_or\_update\_lun\_id** (*volume\_id*, *lun\_id*)

Set the LUN ID on a volume.

**Parameters**

- **volume\_id** (*integer*) – The id of the volume
- **lun\_id** (*integer*) – LUN ID to set on the volume

**Returns** a SoftLayer\_Network\_Storage\_Property object

**create\_snapshot** (*volume\_id*, *notes*=", \*\*kwargs)

Creates a snapshot on the given block volume.

**Parameters**

- **volume\_id** (*integer*) – The id of the volume
- **notes** (*string*) – The notes or “name” to assign the snapshot

**Returns** Returns the id of the new snapshot

**deauthorize\_host\_to\_volume** (*volume\_id*, *hardware\_ids*=None, *virtual\_guest\_ids*=None, *ip\_address\_ids*=None, \*\*kwargs)

Revokes authorization of hosts to Block Storage Volumes

**Parameters**

- **volume\_id** – The Block volume to deauthorize hosts to
- **hardware\_ids** – A List of SoftLayer\_Hardware ids
- **virtual\_guest\_ids** – A List of SoftLayer\_Virtual\_Guest ids
- **ip\_address\_ids** – A List of SoftLayer\_Network\_Subnet\_IpAddress ids

**Returns** Returns an array of SoftLayer\_Network\_Storage\_Allowed\_Host objects which have access to the given Block volume

**delete\_snapshot** (*snapshot\_id*)

Deletes the specified snapshot object.

**Parameters** **snapshot\_id** – The ID of the snapshot object to delete.

**disable\_snapshots** (*volume\_id, schedule\_type*)

Disables snapshots for a specific block volume at a given schedule

**Parameters**

- **volume\_id** (*integer*) – The id of the volume
- **schedule\_type** (*string*) – ‘HOURLY’|‘DAILY’|‘WEEKLY’

**Returns** Returns whether successfully disabled or not

**enable\_snapshots** (*volume\_id, schedule\_type, retention\_count, minute, hour, day\_of\_week, \*\*kwargs*)

Enables snapshots for a specific block volume at a given schedule

**Parameters**

- **volume\_id** (*integer*) – The id of the volume
- **schedule\_type** (*string*) – ‘HOURLY’|‘DAILY’|‘WEEKLY’
- **retention\_count** (*integer*) – Number of snapshots to be kept
- **minute** (*integer*) – Minute when to take snapshot
- **hour** (*integer*) – Hour when to take snapshot
- **day\_of\_week** (*string*) – Day when to take snapshot

**Returns** Returns whether successfully scheduled or not

**failback\_from\_replicant** (*volume\_id*)

Failback from a volume replicant.

**Parameters** **volume\_id** (*integer*) – The id of the volume

**Returns** Returns whether failback was successful or not

**failover\_to\_replicant** (*volume\_id, replicant\_id*)

Failover to a volume replicant.

**Parameters**

- **volume\_id** (*integer*) – The id of the volume
- **replicant\_id** (*integer*) – ID of replicant to failover to

**Returns** Returns whether failover was successful or not

**get\_block\_volume\_access\_list** (*volume\_id, \*\*kwargs*)

Returns a list of authorized hosts for a specified volume.

**Parameters**

- **volume\_id** – ID of volume.
- **kwargs** –

**Returns** Returns a list of authorized hosts for a specified volume.

**get\_block\_volume\_details** (*volume\_id*, *\*\*kwargs*)

Returns details about the specified volume.

**Parameters**

- **volume\_id** – ID of volume.
- **kwargs** –

**Returns** Returns details about the specified volume.

**get\_block\_volume\_snapshot\_list** (*volume\_id*, *\*\*kwargs*)

Returns a list of snapshots for the specified volume.

**Parameters**

- **volume\_id** – ID of volume.
- **kwargs** –

**Returns** Returns a list of snapshots for the specified volume.

**get\_replication\_locations** (*volume\_id*)

Acquires list of the datacenters to which a volume can be replicated.

**Parameters** **volume\_id** – The ID of the primary volume to be replicated

**Returns** Returns an array of SoftLayer\_Network\_Storage objects

**get\_replication\_partners** (*volume\_id*)

Acquires list of replicant volumes pertaining to the given volume.

**Parameters** **volume\_id** – The ID of the primary volume to be replicated

**Returns** Returns an array of SoftLayer\_Location objects

**get\_subnets\_in\_acl** (*access\_id*)

Returns a list of subnet records for the access host.

*access\_id* is the *host\_id* obtained by: `slcli block access-list <volume_id>`

**Parameters** **access\_id** (*integer*) – id of the access host

**Returns** Returns an array of SoftLayer\_Network\_Subnet objects

**list\_block\_volume\_limit** ()

Returns a list of block volume count limit.

**Returns** Returns a list of block volume count limit.

**list\_block\_volumes** (*datacenter=None*, *username=None*, *storage\_type=None*, *\*\*kwargs*)

Returns a list of block volumes.

**Parameters**

- **datacenter** – Datacenter short name (e.g.: dal09)
- **username** – Name of volume.
- **storage\_type** – Type of volume: Endurance or Performance
- **kwargs** –

**Returns** Returns a list of block volumes.

**list\_volume\_schedules** (*volume\_id*)

Lists schedules for a given volume

**Parameters** `volume_id` (*integer*) – The id of the volume

**Returns** Returns list of schedules assigned to a given volume

**order\_block\_volume** (*storage\_type, location, size, os\_type, iops=None, tier\_level=None, snapshot\_size=None, service\_offering='storage\_as\_a\_service', hourly\_billing\_flag=False*)

Places an order for a block volume.

**Parameters**

- **storage\_type** – ‘performance’ or ‘endurance’
- **location** – Datacenter in which to order iSCSI volume
- **size** – Size of the desired volume, in GB
- **os\_type** – OS Type to use for volume alignment, see help for list
- **iops** – Number of IOPs for a “Performance” order
- **tier\_level** – Tier level to use for an “Endurance” order
- **snapshot\_size** – The size of optional snapshot space, if snapshot space should also be ordered (None if not ordered)
- **service\_offering** – Requested offering package to use in the order (‘storage\_as\_a\_service’, ‘enterprise’, or ‘performance’)
- **hourly\_billing\_flag** – Billing type, monthly (False) or hourly (True), default to monthly.

**order\_duplicate\_volume** (*origin\_volume\_id, origin\_snapshot\_id=None, duplicate\_size=None, duplicate\_iops=None, duplicate\_tier\_level=None, duplicate\_snapshot\_size=None, hourly\_billing\_flag=False*)

Places an order for a duplicate block volume.

**Parameters**

- **origin\_volume\_id** – The ID of the origin volume to be duplicated
- **origin\_snapshot\_id** – Origin snapshot ID to use for duplication
- **duplicate\_size** – Size/capacity for the duplicate volume
- **duplicate\_iops** – The IOPS per GB for the duplicate volume
- **duplicate\_tier\_level** – Tier level for the duplicate volume
- **duplicate\_snapshot\_size** – Snapshot space size for the duplicate
- **hourly\_billing\_flag** – Billing type, monthly (False) or hourly (True), default to monthly.

**Returns** Returns a SoftLayer\_Container\_Product\_Order\_Receipt

**order\_modified\_volume** (*volume\_id, new\_size=None, new\_iops=None, new\_tier\_level=None*)

Places an order for modifying an existing block volume.

**Parameters**

- **volume\_id** – The ID of the volume to be modified
- **new\_size** – The new size/capacity for the volume
- **new\_iops** – The new IOPS for the volume
- **new\_tier\_level** – The new tier level for the volume



**Returns** Returns a SoftLayer\_Container\_Product\_Order\_Receipt

**order\_replicant\_volume** (*volume\_id, snapshot\_schedule, location, tier=None, os\_type=None*)  
Places an order for a replicant block volume.

**Parameters**

- **volume\_id** – The ID of the primary volume to be replicated
- **snapshot\_schedule** – The primary volume’s snapshot schedule to use for replication
- **location** – The location for the ordered replicant volume
- **tier** – The tier (IOPS per GB) of the primary volume
- **os\_type** – The OS type of the primary volume

**Returns** Returns a SoftLayer\_Container\_Product\_Order\_Receipt

**order\_snapshot\_space** (*volume\_id, capacity, tier, upgrade, \*\*kwargs*)  
Orders snapshot space for the given block volume.

**Parameters**

- **volume\_id** (*integer*) – The id of the volume
- **capacity** (*integer*) – The capacity to order, in GB
- **tier** (*float*) – The tier level of the block volume, in IOPS per GB
- **upgrade** (*boolean*) – Flag to indicate if this order is an upgrade

**Returns** Returns a SoftLayer\_Container\_Product\_Order\_Receipt

**remove\_subnets\_from\_acl** (*access\_id, subnet\_ids*)  
Removes subnet records from ACL for the access host.

access\_id is the host\_id obtained by: slcli block access-list <volume\_id>

**Parameters**

- **access\_id** (*integer*) – id of the access host
- **subnet\_ids** (*list*) – The ids of the subnets to be removed

**Returns** Returns int array of removed subnet ids

**resolve\_ids** (*identifier*)  
Takes a string and tries to resolve to a list of matching ids.

What exactly ‘identifier’ can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

**restore\_from\_snapshot** (*volume\_id, snapshot\_id*)  
Restores a specific volume from a snapshot

**Parameters**

- **volume\_id** (*integer*) – The id of the volume
- **snapshot\_id** (*integer*) – The id of the restore point

**Returns** Returns whether succesfully restored or not

**set\_credential\_password** (*access\_id, password*)  
Sets the password for an access host

**Parameters**

- **access\_id** (*integer*) – id of the access host
- **password** (*string*) – password to set

### 3.2.4 SoftLayer.cdn

CDN Manager/helpers

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.cdn.CDNManager` (*client*)  
Manage Content Delivery Networks in the account.

See product information here: <https://www.ibm.com/cloud/cdn> <https://cloud.ibm.com/docs/infrastructure/CDN?topic=CDN-about-content-delivery-networks-cdn->

**Parameters** **client** (*SoftLayer.API.BaseClient*) – the client instance

**add\_origin** (*unique\_id, origin, path, origin\_type='server', header=None, port=80, protocol='http', bucket\_name=None, file\_extensions=None, optimize\_for='web', cache\_query='include all'*)

Creates an origin path for an existing CDN.

**Parameters**

- **unique\_id** (*str*) – The unique ID associated with the CDN.
- **path** (*str*) – relative path to the domain provided, e.g. “/articles/video”
- **origin** (*str*) – ip address or hostname if *origin\_type=server*, API endpoint for your S3 object storage if *origin\_type=storage*
- **origin\_type** (*str*) – it can be ‘server’ or ‘storage’ types.
- **header** (*str*) – the edge server uses the host header to communicate with the origin. It defaults to hostname. (optional)
- **port** (*int*) – the http port number (default: 80)
- **protocol** (*str*) – the protocol of the origin (default: HTTP)
- **bucket\_name** (*str*) – name of the available resource
- **file\_extensions** (*str*) – file extensions that can be stored in the CDN, e.g. “jpg,png”
- **optimize\_for** (*str*) – performance configuration, available options: web, video, and file where:
  - ‘web’ = ‘General web delivery’
  - ‘video’ = ‘Video on demand optimization’
  - ‘file’ = ‘Large file optimization’
- **cache\_query** (*str*) – rules with the following formats: ‘include-all’, ‘ignore-all’, ‘include: space separated query-names’, ‘ignore: space separated query-names’.

**Returns** a CDN origin path object

**end\_date**

Retrieve the cdn usage metric end date.

**get\_cdn** (*unique\_id*, *\*\*kwargs*)

Retrieves the information about the CDN account object.

**Parameters**

- **unique\_id** (*str*) – The unique ID associated with the CDN.
- **\*\*kwargs** (*dict*) – header-level option (mask)

**Returns** The CDN object

**get\_origins** (*unique\_id*, *\*\*kwargs*)

Retrieves list of origin pull mappings for a specified CDN account.

**Parameters**

- **unique\_id** (*str*) – The unique ID associated with the CDN.
- **\*\*kwargs** (*dict*) – header-level options (mask, limit, etc.)

**Returns** The list of origin paths in the CDN object.

**get\_usage\_metrics** (*unique\_id*, *history=30*, *frequency='aggregate'*)

Retrieves the cdn usage metrics.

It uses the 'days' argument if start\_date and end\_date are None.

**Parameters**

- **unique\_id** (*int*) – The CDN uniqueId from which the usage metrics will be obtained.
- **history** (*int*) – Last N days, default days is 30.
- **frequency** (*str*) – It can be day, week, month and aggregate. The default is “aggregate”.

**Returns** A Container\_Network\_CdnMarketplace\_Metrics object

**list\_cdn** (*\*\*kwargs*)

Lists Content Delivery Networks for the active user.

**Parameters** **\*\*kwargs** (*dict*) – header-level options (mask, limit, etc.)

**Returns** The list of CDN objects in the account

**purge\_content** (*unique\_id*, *path*)

Purges a URL or path from the CDN.

**Parameters**

- **unique\_id** (*str*) – The unique ID associated with the CDN.
- **path** (*str*) – A string of url or path that should be purged.

**Returns** A Container\_Network\_CdnMarketplace\_Configuration\_Cache\_Purge array object

**remove\_origin** (*unique\_id*, *path*)

Removes an origin pull mapping with the given origin pull ID.

**Parameters**

- **unique\_id** (*str*) – The unique ID associated with the CDN.
- **path** (*str*) – The origin path to delete.

**Returns** A string value

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly 'identifier' can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

**start\_data**

Retrieve the cdn usage metric start date.

### 3.2.5 SoftLayer.dedicatedhost

DH Manager/helpers

**license** MIT, see License for more details.

**class** `SoftLayer.managers.dedicated_host.DedicatedHostManager` (*client*, *ordering\_manager=None*)

Manages SoftLayer Dedicated Hosts.

See product information here <https://www.ibm.com/cloud/dedicated>

**Parameters**

- **client** (*SoftLayer.API.BaseClient*) – the client instance
- **ordering\_manager** (*SoftLayer.managers.OrderingManager*) – an optional manager to handle ordering. If none is provided, one will be auto initialized.

**cancel\_guests** (*host\_id*)

Cancel all guests into the dedicated host immediately.

To cancel an specified guest use the method `VSManger.cancel_instance()`

**Parameters** **host\_id** – The ID of the dedicated host.

**Returns** The id, fqdn and status of all guests into a dictionary. The status could be 'Cancelled' or an exception message, The dictionary is empty if there isn't any guest in the dedicated host.

**Example::** # Cancel guests of dedicated host id 12345 result = mgr.cancel\_guests(12345)

**cancel\_host** (*host\_id*)

Cancel a dedicated host immediately, it fails if there are still guests in the host.

**Parameters** **host\_id** – The ID of the dedicated host to be cancelled.

**Returns** True on success or an exception

**Example::** # Cancels dedicated host id 12345 result = mgr.cancel\_host(12345)

**get\_create\_options** ()

Returns valid options for ordering a dedicated host.

**get\_host** (*host\_id*, *\*\*kwargs*)

Get details about a dedicated host.

:param integer : the host ID :returns: A dictionary containing host information.

Example:

```
# Print out host ID 12345.
dh = mgr.get_host(12345)
print dh

# Print out only name and backendRouter for instance 12345
object_mask = "mask[name,backendRouter[id]]"
dh = mgr.get_host(12345, mask=mask)
print dh
```

**get\_router\_options** (*datacenter=None, flavor=None*)

Returns available backend routers for the dedicated host.

**list\_guests** (*host\_id, tags=None, cpus=None, memory=None, hostname=None, domain=None, local\_disk=None, nic\_speed=None, public\_ip=None, private\_ip=None, \*\*kwargs*)

Retrieve a list of all virtual servers on the dedicated host.

Example:

```
# Print out a list of instances with 4 cpu cores in the host id 12345.

for vsi in mgr.list_guests(host_id=12345, cpus=4):
    print vsi['fullyQualifiedDomainName'], vsi['primaryIpAddress']

# Using a custom object-mask. Will get ONLY what is specified
object_mask = "mask[hostname,monitoringRobot[robotStatus]]"
for vsi in mgr.list_guests(mask=object_mask, cpus=4):
    print vsi
```

#### Parameters

- **host\_id** (*integer*) – the identifier of dedicated host
- **tags** (*list*) – filter based on list of tags
- **cpus** (*integer*) – filter based on number of CPUS
- **memory** (*integer*) – filter based on amount of memory
- **hostname** (*string*) – filter based on hostname
- **domain** (*string*) – filter based on domain
- **local\_disk** (*string*) – filter based on local\_disk
- **nic\_speed** (*integer*) – filter based on network speed (in MBPS)
- **public\_ip** (*string*) – filter based on public ip address
- **private\_ip** (*string*) – filter based on private ip address
- **\*\*kwargs** (*dict*) – response-level options (mask, limit, etc.)

**Returns** Returns a list of dictionaries representing the matching virtual servers

**list\_instances** (*tags=None, cpus=None, memory=None, hostname=None, disk=None, datacenter=None, \*\*kwargs*)

Retrieve a list of all dedicated hosts on the account

#### Parameters

- **tags** (*list*) – filter based on list of tags
- **cpus** (*integer*) – filter based on number of CPUS

- **memory** (*integer*) – filter based on amount of memory
- **hostname** (*string*) – filter based on hostname
- **disk** (*string*) – filter based on disk
- **datacenter** (*string*) – filter based on datacenter
- **\*\*kwargs** (*dict*) – response-level options (mask, limit, etc.)

**Returns** Returns a list of dictionaries representing the matching dedicated host.

**place\_order** (*hostname, domain, location, flavor, hourly, router=None*)

Places an order for a dedicated host.

See `get_create_options()` for valid arguments.

**Parameters**

- **hostname** (*string*) – server hostname
- **domain** (*string*) – server domain name
- **location** (*string*) – location (datacenter) name
- **hourly** (*boolean*) – True if using hourly pricing (default). False for monthly.
- **router** (*int*) – an optional value for selecting a backend router

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly ‘identifier’ can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

**verify\_order** (*hostname, domain, location, hourly, flavor, router=None*)

Verifies an order for a dedicated host.

See `place_order()` for a list of available options.

### 3.2.6 SoftLayer.dns

DNS Manager/helpers

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.dns.DNSManager` (*client*)

Manage SoftLayer DNS.

See product information here: <http://www.softlayer.com/DOMAIN-SERVICES>

**Parameters** **client** (*SoftLayer.API.BaseClient*) – the client instance

**create\_record** (*zone\_id, record, record\_type, data, ttl=60*)

Create a resource record on a domain.

**Parameters**

- **id** (*integer*) – the zone’s ID
- **record** – the name of the record to add
- **record\_type** – the type of record (A, AAAA, CNAME, TXT, etc.)
- **data** – the record’s value

- **t11** (*integer*) – the TTL or time-to-live value (default: 60)

**create\_record\_mx** (*zone\_id, record, data, ttl=60, priority=10*)

Create a mx resource record on a domain.

#### Parameters

- **id** (*integer*) – the zone’s ID
- **record** – the name of the record to add
- **data** – the record’s value
- **t11** (*integer*) – the TTL or time-to-live value (default: 60)
- **priority** (*integer*) – the priority of the target host

**create\_record\_ptr** (*record, data, ttl=60*)

Create a reverse record.

#### Parameters

- **record** – the public ip address of device for which you would like to manage reverse DNS.
- **data** – the record’s value
- **t11** (*integer*) – the TTL or time-to-live value (default: 60)

**create\_record\_srv** (*zone\_id, record, data, protocol, port, service, ttl=60, priority=20, weight=10*)

Create a resource record on a domain.

#### Parameters

- **id** (*integer*) – the zone’s ID
- **record** – the name of the record to add
- **data** – the record’s value
- **protocol** (*string*) – the protocol of the service, usually either TCP or UDP.
- **port** (*integer*) – the TCP or UDP port on which the service is to be found.
- **service** (*string*) – the symbolic name of the desired service.
- **t11** (*integer*) – the TTL or time-to-live value (default: 60)
- **priority** (*integer*) – the priority of the target host (default: 20)
- **weight** (*integer*) – relative weight for records with same priority (default: 10)

**create\_zone** (*zone, serial=None*)

Create a zone for the specified zone.

#### Parameters

- **zone** – the zone name to create
- **serial** – serial value on the zone (default: strptime(“%Y%m%d01”))

**delete\_record** (*record\_id*)

Delete a resource record by its ID.

**Parameters** **id** (*integer*) – the record’s ID

**delete\_zone** (*zone\_id*)

Delete a zone by its ID.

**Parameters** `zone_id` (*integer*) – the zone ID to delete

**dump\_zone** (*zone\_id*)

Retrieve a zone dump in BIND format.

**Parameters** `id` (*integer*) – The zone ID to dump

**edit\_record** (*record*)

Update an existing record with the options provided.

The provided dict must include an 'id' key and value corresponding to the record that should be updated.

**Parameters** `record` (*dict*) – the record to update

**edit\_zone** (*zone*)

Update an existing zone with the options provided.

The provided dict must include an 'id' key and value corresponding to the zone that should be updated.

**Parameters** `zone` (*dict*) – the zone to update

**get\_record** (*record\_id*)

Get a DNS record.

**Parameters** `id` (*integer*) – the record's ID

**get\_records** (*zone\_id*, *t11=None*, *data=None*, *host=None*, *record\_type=None*)

List, and optionally filter, records within a zone.

**Parameters**

- **zone** – the zone name in which to search.
- **t11** (*int*) – time in seconds
- **data** (*str*) – the records data
- **host** (*str*) – record's host
- **record\_type** (*str*) – the type of record

**Returns** A list of dictionaries representing the matching records within the specified zone.

**get\_zone** (*zone\_id*, *records=True*)

Get a zone and its records.

**Parameters** `zone` – the zone name

**Returns** A dictionary containing a large amount of information about the specified zone.

**list\_zones** (*\*\*kwargs*)

Retrieve a list of all DNS zones.

**Parameters** **\*\*kwargs** (*dict*) – response-level options (mask, limit, etc.)

**Returns** A list of dictionaries representing the matching zones.

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly 'identifier' can be depends on the resolvers

**Parameters** `identifier` (*string*) – identifying string

**Returns** list

**sync\_host\_record** (*zone\_id*, *hostname*, *ip\_address*, *record\_type='a'*, *t11=7200*)

For a given zone\_id, will set hostname's A record to ip\_address



**Parameters**

- **zone\_id** (*integer*) – The zone id for the domain
- **hostname** (*string*) – host part of the record
- **ip\_address** (*string*) – data part of the record
- **t1** (*integer*) – TTL for the record
- **record\_type** (*string*) – ‘a’ or ‘aaaa’

**sync\_ptr\_record** (*ptr\_domains, ip\_address, fqdn, ttl=7200*)  
Sync PTR record.

**Parameters**

- **ptr\_domains** (*dict*) – result from SoftLayer\_Virtual\_Guest.getReverseDomainRecords or SoftLayer\_Hardware\_Server.getReverseDomainRecords
- **ip\_address** (*string*) – ip address to sync with
- **fqdn** (*string*) – Fully Qualified Domain Name
- **t1** (*integer*) – TTL for the record

### 3.2.7 SoftLayer.event\_log

Network Manager/helpers

**license** MIT, see LICENSE for more details.

**class** SoftLayer.managers.event\_log.**EventLogManager** (*client*)

Provides an interface for the SoftLayer Event Log Service.

See product information here: [http://sldn.softlayer.com/reference/services/SoftLayer\\_Event\\_Log](http://sldn.softlayer.com/reference/services/SoftLayer_Event_Log)

**static build\_filter** (*date\_min=None, date\_max=None, obj\_event=None, obj\_id=None, obj\_type=None, utc\_offset=None*)

Returns a query filter that can be passed into EventLogManager.get\_event\_logs

**Parameters**

- **date\_min** (*string*) – Lower bound date in MM/DD/YYYY format
- **date\_max** (*string*) – Upper bound date in MM/DD/YYYY format
- **obj\_event** (*string*) – The name of the events we want to filter by
- **obj\_id** (*int*) – The id of the event we want to filter by
- **obj\_type** (*string*) – The type of event we want to filter by
- **utc\_offset** (*string*) – The UTC offset we want to use when converting date\_min and date\_max. (default ‘+0000’)

**Returns** dict: The generated query filter

**get\_event\_log\_types** ()

Returns a list of event log types

**Returns** List of event log types

**get\_event\_logs** (*request\_filter=None, log\_limit=20, iterator=True*)

Returns a list of event logs

Example:

```
event_mgr = SoftLayer.EventLogManager(env.client)
request_filter = event_mgr.build_filter(date_min="01/01/2019", date_max="02/
↪01/2019")
logs = event_mgr.get_event_logs(request_filter)
for log in logs:
    print("Event Name: {}".format(log['eventName']))
```

#### Parameters

- **request\_filter** (*dict*) – filter dict
- **log\_limit** (*int*) – number of results to get in one API call
- **iterator** (*bool*) – False will only make one API call for log\_limit results. True will keep making API calls until all logs have been retrieved. There may be a lot of these.

**Returns** List of event logs. If iterator=True, will return a python generator object instead.

### 3.2.8 SoftLayer.file

File Storage Manager

**license** MIT, see LICENSE for more details.

**class** SoftLayer.managers.file.**FileStorageManager** (*client*)

Manages file Storage volumes.

**authorize\_host\_to\_volume** (*volume\_id*, *hardware\_ids=None*, *virtual\_guest\_ids=None*,  
*ip\_address\_ids=None*, *subnet\_ids=None*, *\*\*kwargs*)

Authorizes hosts to File Storage Volumes

#### Parameters

- **volume\_id** – The File volume to authorize hosts to
- **hardware\_ids** – A List of SoftLayer\_Hardware ids
- **virtual\_guest\_ids** – A List of SoftLayer\_Virtual\_Guest ids
- **ip\_address\_ids** – A List of SoftLayer\_Network\_Subnet\_IpAddress ids
- **subnet\_ids** – A List of SoftLayer\_Network\_Subnet ids

**Returns** Returns an array of SoftLayer\_Network\_Storage\_Allowed\_Host objects which now have access to the given File volume

**cancel\_file\_volume** (*volume\_id*, *reason='No longer needed'*, *immediate=False*)

Cancels the given file storage volume.

#### Parameters

- **volume\_id** (*integer*) – The volume ID
- **reason** (*string*) – The reason for cancellation
- **immediate** (*boolean*) – Cancel immediately or on anniversary date

**cancel\_snapshot\_space** (*volume\_id*, *reason='No longer needed'*, *immediate=False*)

Cancels snapshot space for a given volume.

#### Parameters

- **volume\_id** (*integer*) – The volume ID

- **reason** (*string*) – The reason for cancellation
- **immediate** (*boolean*) – Cancel immediately or on anniversary date

**create\_snapshot** (*volume\_id*, *notes*=”, *\*\*kwargs*)

Creates a snapshot on the given file volume.

#### Parameters

- **volume\_id** (*integer*) – The id of the volume
- **notes** (*string*) – The notes or “name” to assign the snapshot

**Returns** Returns the id of the new snapshot

**deauthorize\_host\_to\_volume** (*volume\_id*, *hardware\_ids*=None, *virtual\_guest\_ids*=None, *ip\_address\_ids*=None, *subnet\_ids*=None, *\*\*kwargs*)

Revokes authorization of hosts to File Storage Volumes

#### Parameters

- **volume\_id** – The File volume to deauthorize hosts to
- **hardware\_ids** – A List of SoftLayer\_Hardware ids
- **virtual\_guest\_ids** – A List of SoftLayer\_Virtual\_Guest ids
- **ip\_address\_ids** – A List of SoftLayer\_Network\_Subnet\_IpAddress ids
- **subnet\_ids** – A List of SoftLayer\_Network\_Subnet ids

**Returns** Returns an array of SoftLayer\_Network\_Storage\_Allowed\_Host objects which have access to the given File volume

**delete\_snapshot** (*snapshot\_id*)

Deletes the specified snapshot object.

**Parameters** **snapshot\_id** – The ID of the snapshot object to delete.

**disable\_snapshots** (*volume\_id*, *schedule\_type*)

Disables snapshots for a specific file volume at a given schedule

#### Parameters

- **volume\_id** (*integer*) – The id of the volume
- **schedule\_type** (*string*) – ‘HOURLY’|‘DAILY’|‘WEEKLY’

**Returns** Returns whether successfully disabled or not

**enable\_snapshots** (*volume\_id*, *schedule\_type*, *retention\_count*, *minute*, *hour*, *day\_of\_week*, *\*\*kwargs*)

Enables snapshots for a specific file volume at a given schedule

#### Parameters

- **volume\_id** (*integer*) – The id of the volume
- **schedule\_type** (*string*) – ‘HOURLY’|‘DAILY’|‘WEEKLY’
- **retention\_count** (*integer*) – The number of snapshots to attempt to retain in this schedule
- **minute** (*integer*) – The minute of the hour at which HOURLY, DAILY, and WEEKLY snapshots should be taken
- **hour** (*integer*) – The hour of the day at which DAILY and WEEKLY snapshots should be taken

- **day\_of\_week** (*string/integer*) – The day of the week on which WEEKLY snapshots should be taken, either as a string ('SUNDAY') or integer ('0' is Sunday)

**Returns** Returns whether successfully scheduled or not

**failback\_from\_replicant** (*volume\_id*)

Failback from a volume replicant.

**Parameters** **volume\_id** (*integer*) – The ID of the volume

**Returns** Returns whether failback was successful or not

**failover\_to\_replicant** (*volume\_id, replicant\_id*)

Failover to a volume replicant.

**Parameters**

- **volume\_id** (*integer*) – The ID of the volume
- **replicant\_id** (*integer*) – ID of replicant to failover to

**Returns** Returns whether failover was successful or not

**get\_file\_volume\_access\_list** (*volume\_id, \*\*kwargs*)

Returns a list of authorized hosts for a specified volume.

**Parameters**

- **volume\_id** – ID of volume.
- **kwargs** –

**Returns** Returns a list of authorized hosts for a specified volume.

**get\_file\_volume\_details** (*volume\_id, \*\*kwargs*)

Returns details about the specified volume.

**Parameters**

- **volume\_id** – ID of volume.
- **kwargs** –

**Returns** Returns details about the specified volume.

**get\_file\_volume\_snapshot\_list** (*volume\_id, \*\*kwargs*)

Returns a list of snapshots for the specified volume.

**Parameters**

- **volume\_id** – ID of volume.
- **kwargs** –

**Returns** Returns a list of snapshots for the specified volume.

**get\_replication\_locations** (*volume\_id*)

Acquires list of the datacenters to which a volume can be replicated.

**Parameters** **volume\_id** – The ID of the primary volume to be replicated

**Returns** Returns an array of SoftLayer\_Network\_Storage objects

**get\_replication\_partners** (*volume\_id*)

Acquires list of replicant volumes pertaining to the given volume.

**Parameters** **volume\_id** – The ID of the primary volume to be replicated

**Returns** Returns an array of SoftLayer\_Location objects

**list\_file\_volume\_limit()**

Returns a list of file volume count limit.

**Returns** Returns a list of file volume count limit.

**list\_file\_volumes** (*datacenter=None, username=None, storage\_type=None, \*\*kwargs*)

Returns a list of file volumes.

**Parameters**

- **datacenter** – Datacenter short name (e.g.: dal09)
- **username** – Name of volume.
- **storage\_type** – Type of volume: Endurance or Performance
- **kwargs** –

**Returns** Returns a list of file volumes.

**list\_volume\_schedules** (*volume\_id*)

Lists schedules for a given volume

**Parameters** **volume\_id** (*integer*) – The id of the volume

**Returns** Returns list of schedules assigned to a given volume

**order\_duplicate\_volume** (*origin\_volume\_id, origin\_snapshot\_id=None, duplicate\_size=None, duplicate\_iops=None, duplicate\_tier\_level=None, duplicate\_snapshot\_size=None, hourly\_billing\_flag=False*)

Places an order for a duplicate file volume.

**Parameters**

- **origin\_volume\_id** – The ID of the origin volume to be duplicated
- **origin\_snapshot\_id** – Origin snapshot ID to use for duplication
- **duplicate\_size** – Size/capacity for the duplicate volume
- **duplicate\_iops** – The IOPS per GB for the duplicate volume
- **duplicate\_tier\_level** – Tier level for the duplicate volume
- **duplicate\_snapshot\_size** – Snapshot space size for the duplicate
- **hourly\_billing\_flag** – Billing type, monthly (False) or hourly (True), default to monthly.

**Returns** Returns a SoftLayer\_Container\_Product\_Order\_Receipt

**order\_file\_volume** (*storage\_type, location, size, iops=None, tier\_level=None, snapshot\_size=None, service\_offering='storage\_as\_a\_service', hourly\_billing\_flag=False*)

Places an order for a file volume.

**Parameters**

- **storage\_type** – ‘performance’ or ‘endurance’
- **location** – Name of the datacenter in which to order the volume
- **size** – Size of the desired volume, in GB
- **iops** – Number of IOPs for a “Performance” order
- **tier\_level** – Tier level to use for an “Endurance” order

- **snapshot\_size** – The size of optional snapshot space, if snapshot space should also be ordered (None if not ordered)
- **service\_offering** – Requested offering package to use in the order ('storage\_as\_a\_service', 'enterprise', or 'performance')
- **hourly\_billing\_flag** – Billing type, monthly (False) or hourly (True), default to monthly.

**order\_modified\_volume** (*volume\_id, new\_size=None, new\_iops=None, new\_tier\_level=None*)  
Places an order for modifying an existing file volume.

**Parameters**

- **volume\_id** – The ID of the volume to be modified
- **new\_size** – The new size/capacity for the volume
- **new\_iops** – The new IOPS for the volume
- **new\_tier\_level** – The new tier level for the volume

**Returns** Returns a SoftLayer\_Container\_Product\_Order\_Receipt

**order\_replicant\_volume** (*volume\_id, snapshot\_schedule, location, tier=None*)  
Places an order for a replicant file volume.

**Parameters**

- **volume\_id** – The ID of the primary volume to be replicated
- **snapshot\_schedule** – The primary volume's snapshot schedule to use for replication
- **location** – The location for the ordered replicant volume
- **tier** – The tier (IOPS per GB) of the primary volume

**Returns** Returns a SoftLayer\_Container\_Product\_Order\_Receipt

**order\_snapshot\_space** (*volume\_id, capacity, tier, upgrade, \*\*kwargs*)  
Orders snapshot space for the given file volume.

**Parameters**

- **volume\_id** (*integer*) – The ID of the volume
- **capacity** (*integer*) – The capacity to order, in GB
- **tier** (*float*) – The tier level of the file volume, in IOPS per GB
- **upgrade** (*boolean*) – Flag to indicate if this order is an upgrade

**Returns** Returns a SoftLayer\_Container\_Product\_Order\_Receipt

**resolve\_ids** (*identifier*)  
Takes a string and tries to resolve to a list of matching ids.

What exactly 'identifier' can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

**restore\_from\_snapshot** (*volume\_id, snapshot\_id*)  
Restores a specific volume from a snapshot

**Parameters**

- **volume\_id** (*integer*) – The ID of the volume

- **snapshot\_id** (*integer*) – The id of the restore point

**Returns** Returns whether successfully restored or not

### 3.2.9 SoftLayer.firewall

Firewall Manager/helpers

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.firewall.FirewallManager` (*client*)  
Manages SoftLayer firewalls

See product information here: <http://www.softlayer.com/firewalls>

**Parameters** **client** (*SoftLayer.API.BaseClient*) – the client instance

**add\_standard\_firewall** (*server\_id, is\_virt=True*)

Creates a firewall for the specified virtual/hardware server.

**Parameters**

- **server\_id** (*int*) – The ID of the server to create the firewall for
- **is\_virt** (*bool*) – If true, will create the firewall for a virtual server, otherwise for a hardware server.

**Returns** A dictionary containing the standard virtual server firewall order

**add\_vlan\_firewall** (*vlan\_id, ha\_enabled=False*)

Creates a firewall for the specified vlan.

**Parameters**

- **vlan\_id** (*int*) – The ID of the vlan to create the firewall for
- **ha\_enabled** (*bool*) – If True, an HA firewall will be created

**Returns** A dictionary containing the VLAN firewall order

**cancel\_firewall** (*firewall\_id, dedicated=False*)

Cancels the specified firewall.

**Parameters**

- **firewall\_id** (*int*) – Firewall ID to be cancelled.
- **dedicated** (*bool*) – If true, the firewall instance is dedicated, otherwise, the firewall instance is shared.

**edit\_dedicated\_fw\_rules** (*firewall\_id, rules*)

Edit the rules for dedicated firewall.

**Parameters**

- **firewall\_id** (*integer*) – the instance ID of the dedicated firewall
- **rules** (*list*) – the rules to be pushed on the firewall as defined by `SoftLayer_Network_Firewall_Update_Request_Rule`

**edit\_standard\_fw\_rules** (*firewall\_id, rules*)

Edit the rules for standard firewall.

**Parameters**

- **firewall\_id** (*integer*) – the instance ID of the standard firewall

- **rules** (*list*) – the rules to be pushed on the firewall as defined by SoftLayer\_Network\_Firewall\_Update\_Request\_Rule

**get\_dedicated\_fw\_rules** (*firewall\_id*)

Get the rules of a dedicated firewall.

**Parameters** **firewall\_id** (*integer*) – the instance ID of the dedicated firewall

**Returns** A list of the rules.

**get\_dedicated\_package** (*ha\_enabled=False*)

Retrieves the dedicated firewall package.

**Parameters** **ha\_enabled** (*bool*) – True if HA is to be enabled on the firewall False for No HA

**Returns** A dictionary containing the dedicated virtual server firewall package

**get\_firewalls** ()

Returns a list of all firewalls on the account.

**Returns** A list of firewalls on the current account.

**get\_standard\_fw\_rules** (*firewall\_id*)

Get the rules of a standard firewall.

**Parameters** **firewall\_id** (*integer*) – the instance ID of the standard firewall

**Returns** A list of the rules.

**get\_standard\_package** (*server\_id, is\_virt=True*)

Retrieves the standard firewall package for the virtual server.

**Parameters**

- **server\_id** (*int*) – The ID of the server to create the firewall for
- **is\_virt** (*bool*) – True if the ID provided is for a virtual server, False for a server

**Returns** A dictionary containing the standard virtual server firewall package

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly ‘identifier’ can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

`SoftLayer.managers.firewall.has_firewall(vlan)`

Helper to determine whether or not a VLAN has a firewall.

**Parameters** **vlan** (*dict*) – A dictionary representing a VLAN

**Returns** True if the VLAN has a firewall, false if it doesn't.

### 3.2.10 SoftLayer.hardware

Hardware Manager/helpers

**license** MIT, see LICENSE for more details.



**class** SoftLayer.managers.hardware.**HardwareManager** (*client, ordering\_manager=None*)  
 Manage SoftLayer hardware servers.

Example:

```
# Initialize the Manager.
# env variables. These can also be specified in ~/.softlayer,
# or passed directly to SoftLayer.Client()
# SL_USERNAME = YOUR_USERNAME
# SL_API_KEY = YOUR_API_KEY
import SoftLayer
client = SoftLayer.Client()
mgr = SoftLayer.HardwareManager(client)
```

See product information here: <http://www.softlayer.com/bare-metal-servers>

#### Parameters

- **client** (*SoftLayer.API.BaseClient*) – the client instance
- **ordering\_manager** (*SoftLayer.managers.OrderingManager*) – an optional manager to handle ordering. If none is provided, one will be auto initialized.

**cancel\_hardware** (*hardware\_id, reason='unneeded', comment='', immediate=False*)

Cancels the specified dedicated server.

Example:

```
# Cancels hardware id 1234
result = mgr.cancel_hardware(hardware_id=1234)
```

#### Parameters

- **hardware\_id** (*int*) – The ID of the hardware to be cancelled.
- **reason** (*string*) – The reason code for the cancellation. This should come from *get\_cancellation\_reasons()*.
- **comment** (*string*) – An optional comment to include with the cancellation.
- **immediate** (*bool*) – If set to True, will automatically update the cancelation ticket to request the resource be reclaimed asap. This request still has to be reviewed by a human

**Returns** True on success or an exception

**change\_port\_speed** (*hardware\_id, public, speed*)

Allows you to change the port speed of a server's NICs.

#### Parameters

- **hardware\_id** (*int*) – The ID of the server
- **public** (*bool*) – Flag to indicate which interface to change. True (default) means the public interface. False indicates the private interface.
- **speed** (*int*) – The port speed to set.

**Warning:** A port speed of 0 will disable the interface.

Example:

```
#change the Public interface to 10Mbps on instance 12345
result = mgr.change_port_speed(hardware_id=12345,
                               public=True, speed=10)
# result will be True or an Exception
```

**edit** (*hardware\_id*, *userdata=None*, *hostname=None*, *domain=None*, *notes=None*, *tags=None*)

Edit hostname, domain name, notes, user data of the hardware.

Parameters set to None will be ignored and not attempted to be updated.

#### Parameters

- **hardware\_id** (*integer*) – the instance ID to edit
- **userdata** (*string*) – user data on the hardware to edit. If none exist it will be created
- **hostname** (*string*) – valid hostname
- **domain** (*string*) – valid domain name
- **notes** (*string*) – notes about this particular hardware
- **tags** (*string*) – tags to set on the hardware as a comma separated list. Use the empty string to remove all tags.

Example:

```
# Change the hostname on instance 12345 to 'something'
result = mgr.edit(hardware_id=12345 , hostname="something")
#result will be True or an Exception
```

**get\_bandwidth\_allocation** (*instance\_id*)

Combines `getBandwidthAllotmentDetail()` and `getBillingCycleBandwidthUsage()`

**get\_bandwidth\_data** (*instance\_id*, *start\_date=None*, *end\_date=None*, *direction=None*,  
*rollup=3600*)

Gets bandwidth data for a server

Will get averaged bandwidth data for a given time period. If you use a rollup over 3600 be aware that the API will bump your start/end date to align with how data is stored. For example if you have a rollup of 86400 your start\_date will be bumped to 00:00. If you are not using a time in the start/end date fields, this won't really matter.

#### Parameters

- **instance\_id** (*int*) – Hardware Id to get data for
- **start\_date** (*date*) – Date to start pulling data for.
- **end\_date** (*date*) – Date to finish pulling data for
- **direction** (*string*) – Can be either 'public', 'private', or None for both.
- **rollup** (*int*) – 300, 600, 1800, 3600, 43200 or 86400 seconds to average data over.

**get\_cancellation\_reasons** ()

Returns a dictionary of valid cancellation reasons.

These can be used when cancelling a dedicated server via `cancel_hardware()`.

**get\_create\_options** ()

Returns valid options for ordering hardware.

**get\_hardware** (*hardware\_id*, *\*\*kwargs*)

Get details about a hardware device.

**Parameters** `id` (*integer*) – the hardware ID

**Returns** A dictionary containing a large amount of information about the specified server.

Example:

```
object_mask = "mask[id,networkVlans[vlanNumber]]"
# Object masks are optional
result = mgr.get_hardware(hardware_id=1234,mask=object_mask)
```

**get\_tracking\_id** (*instance\_id*)

Returns the Metric Tracking Object Id for a hardware server

**Parameters** `instance_id` (*int*) – Id of the hardware server

**list\_hardware** (*tags=None, cpus=None, memory=None, hostname=None, domain=None, datacenter=None, nic\_speed=None, public\_ip=None, private\_ip=None, \*\*kwargs*)

List all hardware (servers and bare metal computing instances).

**param list tags** filter based on tags

**param integer cpus** filter based on number of CPUS

**param integer memory** filter based on amount of memory in gigabytes

**param string hostname** filter based on hostname

**param string domain** filter based on domain

**param string datacenter** filter based on datacenter

**param integer nic\_speed** filter based on network speed (in MBPS)

**param string public\_ip** filter based on public ip address

**param string private\_ip** filter based on private ip address

**param dict \*\*kwargs** response-level options (mask, limit, etc.)

**returns** Returns a list of dictionaries representing the matching hardware. This list will contain both dedicated servers and bare metal computing instances

Example:

```
# Using a custom object-mask. Will get ONLY what is specified
# These will stem from the SoftLayer_Hardware_Server datatype
object_mask = "mask[hostname,monitoringRobot[robotStatus]]"
result = mgr.list_hardware(mask=object_mask)
```

**place\_order** (*\*\*kwargs*)

Places an order for a piece of hardware.

See `get_create_options()` for valid arguments.

**Parameters**

- **size** (*string*) – server size name or presetId
- **hostname** (*string*) – server hostname
- **domain** (*string*) – server domain name
- **location** (*string*) – location (datacenter) name
- **os** (*string*) – operating system name
- **port\_speed** (*int*) – Port speed in Mbps

- **ssh\_keys** (*list*) – list of ssh key ids
- **post\_uri** (*string*) – The URI of the post-install script to run after reload
- **hourly** (*boolean*) – True if using hourly pricing (default). False for monthly.
- **no\_public** (*boolean*) – True if this server should only have private interfaces
- **extras** (*list*) – List of extra feature names

**reflash\_firmware** (*hardware\_id, ipmi=True, raid\_controller=True, bios=True*)

Reflash hardware firmware.

This will cause the server to be unavailable for ~60 minutes. The firmware will not be upgraded but rather reflashed to the version installed.

#### Parameters

- **hardware\_id** (*int*) – The ID of the hardware to have its firmware reflashed.
- **ipmi** (*bool*) – Reflash the ipmi firmware.
- **raid\_controller** (*bool*) – Reflash the raid controller firmware.
- **bios** (*bool*) – Reflash the bios firmware.

Example:

```
# Check the servers active transactions to see progress
result = mgr.reflash_firmware(hardware_id=1234)
```

**reload** (*hardware\_id, post\_uri=None, ssh\_keys=None*)

Perform an OS reload of a server with its current configuration.

#### Parameters

- **hardware\_id** (*integer*) – the instance ID to reload
- **post\_uri** (*string*) – The URI of the post-install script to run after reload
- **ssh\_keys** (*list*) – The SSH keys to add to the root user

**rescue** (*hardware\_id*)

Reboot a server into the a rescue kernel.

**Parameters** **instance\_id** (*integer*) – the server ID to rescue

Example:

```
result = mgr.rescue(1234)
```

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly 'identifier' can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

**update\_firmware** (*hardware\_id, ipmi=True, raid\_controller=True, bios=True, hard\_drive=True*)

Update hardware firmware.

This will cause the server to be unavailable for ~20 minutes.

#### Parameters

- **hardware\_id** (*int*) – The ID of the hardware to have its firmware updated.

- **ipmi** (*bool*) – Update the ipmi firmware.
- **raid\_controller** (*bool*) – Update the raid controller firmware.
- **bios** (*bool*) – Update the bios firmware.
- **hard\_drive** (*bool*) – Update the hard drive firmware.

Example:

```
# Check the servers active transactions to see progress
result = mgr.update_firmware(hardware_id=1234)
```

**verify\_order** (*\*\*kwargs*)

Verifies an order for a piece of hardware.

See `place_order()` for a list of available options.

**wait\_for\_ready** (*instance\_id, limit=14400, delay=10, pending=False*)

Determine if a Server is ready.

A server is ready when no transactions are running on it.

#### Parameters

- **instance\_id** (*int*) – The instance ID with the pending transaction
- **limit** (*int*) – The maximum amount of seconds to wait.
- **delay** (*int*) – The number of seconds to sleep before checks. Defaults to 10.

### 3.2.11 SoftLayer.image

Image Manager/helpers

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.image.ImageManager` (*client*)

Manages SoftLayer server images.

See product information here: [https://console.ibm.com/docs/infrastructure/image-templates/image\\_index.html](https://console.ibm.com/docs/infrastructure/image-templates/image_index.html)

**Parameters** **client** (`SoftLayer.API.BaseClient`) – the client instance

**delete\_image** (*image\_id*)

Deletes the specified image.

**Parameters** **image\_id** (*int*) – The ID of the image.

**edit** (*image\_id, name=None, note=None, tag=None*)

Edit image related details.

#### Parameters

- **image\_id** (*int*) – The ID of the image
- **name** (*string*) – Name of the Image.
- **note** (*string*) – Note of the image.
- **tag** (*string*) – Tags of the image to be updated to.

**export\_image\_to\_uri** (*image\_id, uri, ibm\_api\_key=None*)

Export image into the given object storage

**Parameters**

- **image\_id** (*int*) – The ID of the image
- **uri** (*string*) – The URI for object storage of the format `swift://<objectStorageAccount>@<cluster>/<container>/<objectPath>` or `cos://<regionName>/<bucketName>/<objectPath>` if using IBM Cloud Object Storage
- **ibm\_api\_key** (*string*) – Ibm Api Key needed to communicate with IBM Cloud Object Storage

**get\_image** (*image\_id*, *\*\*kwargs*)

Get details about an image.

**Parameters**

- **image** (*int*) – The ID of the image.
- **\*\*kwargs** (*dict*) – response-level options (mask, limit, etc.)

**import\_image\_from\_uri** (*name*, *uri*, *os\_code=None*, *note=None*, *ibm\_api\_key=None*,  
*root\_key\_crn=None*, *wrapped\_dek=None*, *cloud\_init=False*,  
*byol=False*, *is\_encrypted=False*)

Import a new image from object storage.

**Parameters**

- **name** (*string*) – Name of the new image
- **uri** (*string*) – The URI for an object storage object (.vhd/.iso file) of the format: `swift://<objectStorageAccount>@<cluster>/<container>/<objectPath>` or (.vhd/.iso/.raw file) of the format: `cos://<regionName>/<bucketName>/<objectPath>` if using IBM Cloud Object Storage
- **os\_code** (*string*) – The reference code of the operating system
- **note** (*string*) – Note to add to the image
- **ibm\_api\_key** (*string*) – Ibm Api Key needed to communicate with ICOS and your KMS
- **root\_key\_crn** (*string*) – CRN of the root key in your KMS. Go to your KMS (Key Protect or Hyper Protect) provider to get the CRN for your root key. An example CRN: `crn:v1:bluemix:public:hs-crypto:us-south:acctID:serviceID:key:keyID` Used only when `is_encrypted` is True.
- **wrapped\_dek** (*string*) – Wrapped Data Encryption Key provided by your KMS. Used only when `is_encrypted` is True.
- **cloud\_init** (*boolean*) – Specifies if image is cloud-init
- **byol** (*boolean*) – Specifies if image is bring your own license
- **is\_encrypted** (*boolean*) – Specifies if image is encrypted

**list\_private\_images** (*guid=None*, *name=None*, *\*\*kwargs*)

List all private images.

**Parameters**

- **guid** (*string*) – filter based on GUID
- **name** (*string*) – filter based on name
- **\*\*kwargs** (*dict*) – response-level options (mask, limit, etc.)

**list\_public\_images** (*guid=None, name=None, \*\*kwargs*)

List all public images.

**Parameters**

- **guid** (*string*) – filter based on GUID
- **name** (*string*) – filter based on name
- **\*\*kwargs** (*dict*) – response-level options (mask, limit, etc.)

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly ‘identifier’ can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

### 3.2.12 SoftLayer.ipsec

IPSec VPN Manager

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.ipsec.IPSECManager` (*client*)

Manage SoftLayer IPSEC VPN tunnel contexts.

This provides helpers to manage IPSEC contexts, private and remote subnets, and NAT translations.

**Parameters**

- **client** (*SoftLayer.API.BaseClient*) – the client instance
- **account** (*SoftLayer.API.BaseClient*) – account service client
- **context** (*SoftLayer.API.BaseClient*) – tunnel context client
- **customer\_subnet** (*SoftLayer.API.BaseClient*) – remote subnet client

**add\_internal\_subnet** (*context\_id, subnet\_id*)

Add an internal subnet to a tunnel context.

**Parameters**

- **context\_id** (*int*) – The id-value representing the context instance.
- **subnet\_id** (*int*) – The id-value representing the internal subnet.

**Return bool** True if internal subnet addition was successful.

**add\_remote\_subnet** (*context\_id, subnet\_id*)

Adds a remote subnet to a tunnel context.

**Parameters**

- **context\_id** (*int*) – The id-value representing the context instance.
- **subnet\_id** (*int*) – The id-value representing the remote subnet.

**Return bool** True if remote subnet addition was successful.

**add\_service\_subnet** (*context\_id, subnet\_id*)

Adds a service subnet to a tunnel context.

**Parameters**

- **context\_id** (*int*) – The id-value representing the context instance.
- **subnet\_id** (*int*) – The id-value representing the service subnet.

**Return bool** True if service subnet addition was successful.

**apply\_configuration** (*context\_id*)

Requests network configuration for a tunnel context.

**Parameters** **context\_id** (*int*) – The id-value representing the context instance.

**Return bool** True if the configuration request was successfully queued.

**create\_remote\_subnet** (*account\_id, identifier, cidr*)

Creates a remote subnet on the given account.

**Parameters**

- **account\_id** (*string*) – The account identifier.
- **identifier** (*string*) – The network identifier of the remote subnet.
- **cidr** (*string*) – The CIDR value of the remote subnet.

**Return dict** Mapping of properties for the new remote subnet.

**create\_translation** (*context\_id, static\_ip, remote\_ip, notes*)

Creates an address translation on a tunnel context/

**Parameters**

- **context\_id** (*int*) – The id-value representing the context instance.
- **static\_ip** (*string*) – The IP address value representing the internal side of the translation entry,
- **remote\_ip** (*string*) – The IP address value representing the remote side of the translation entry,
- **notes** (*string*) – The notes to supply with the translation entry,

**Return dict** Mapping of properties for the new translation entry.

**delete\_remote\_subnet** (*subnet\_id*)

Deletes a remote subnet from the current account.

**Parameters** **subnet\_id** (*string*) – The id-value representing the remote subnet.

**Return bool** True if subnet deletion was successful.

**get\_translation** (*context\_id, translation\_id*)

Retrieves a translation entry for the given id values.

**Parameters**

- **context\_id** (*int*) – The id-value representing the context instance.
- **translation\_id** (*int*) – The id-value representing the translation instance.

**Return dict** Mapping of properties for the translation entry.

**Raises** *SoftLayerAPIError* – If a translation cannot be found.

**get\_translations** (*context\_id*)

Retrieves all translation entries for a tunnel context.

**Parameters** **context\_id** (*int*) – The id-value representing the context instance.

**Return list(dict)** Translations associated with the given context



**get\_tunnel\_context** (*context\_id*, *\*\*kwargs*)

Retrieves the network tunnel context instance.

**Parameters** **context\_id** (*int*) – The id-value representing the context instance.

**Return dict** Mapping of properties for the tunnel context.

**Raises** *SoftLayerAPIError* – If a context cannot be found.

**get\_tunnel\_contexts** (*\*\*kwargs*)

Retrieves network tunnel module context instances.

**Return list(dict)** Contexts associated with the current account.

**remove\_internal\_subnet** (*context\_id*, *subnet\_id*)

Remove an internal subnet from a tunnel context.

**Parameters**

- **context\_id** (*int*) – The id-value representing the context instance.
- **subnet\_id** (*int*) – The id-value representing the internal subnet.

**Return bool** True if internal subnet removal was successful.

**remove\_remote\_subnet** (*context\_id*, *subnet\_id*)

Removes a remote subnet from a tunnel context.

**Parameters**

- **context\_id** (*int*) – The id-value representing the context instance.
- **subnet\_id** (*int*) – The id-value representing the remote subnet.

**Return bool** True if remote subnet removal was successful.

**remove\_service\_subnet** (*context\_id*, *subnet\_id*)

Removes a service subnet from a tunnel context.

**Parameters**

- **context\_id** (*int*) – The id-value representing the context instance.
- **subnet\_id** (*int*) – The id-value representing the service subnet.

**Return bool** True if service subnet removal was successful.

**remove\_translation** (*context\_id*, *translation\_id*)

Removes a translation entry from a tunnel context.

**Parameters**

- **context\_id** (*int*) – The id-value representing the context instance.
- **translation\_id** (*int*) – The id-value representing the translation.

**Return bool** True if translation entry removal was successful.

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly ‘identifier’ can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns list**

**update\_translation** (*context\_id*, *translation\_id*, *static\_ip=None*, *remote\_ip=None*, *notes=None*)

Updates an address translation entry using the given values.

**Parameters**

- **context\_id** (*int*) – The id-value representing the context instance.
- **template** (*dict*) – A key-value mapping of translation properties.
- **static\_ip** (*string*) – The static IP address value to update.
- **remote\_ip** (*string*) – The remote IP address value to update.
- **notes** (*string*) – The notes value to update.

**Return bool** True if the update was successful.

```
update_tunnel_context (context_id, friendly_name=None, remote_peer=None,  
                        preshared_key=None, phase1_auth=None,  
                        phase1_crypto=None, phase1_dh=None, phase1_key_ttl=None,  
                        phase2_auth=None, phase2_crypto=None, phase2_dh=None,  
                        phase2_forward_secrecy=None, phase2_key_ttl=None)
```

Updates a tunnel context using the given values.

**Parameters**

- **context\_id** (*string*) – The id-value representing the context.
- **friendly\_name** (*string*) – The friendly name value to update.
- **remote\_peer** (*string*) – The remote peer IP address value to update.
- **preshared\_key** (*string*) – The preshared key value to update.
- **phase1\_auth** (*string*) – The phase 1 authentication value to update.
- **phase1\_crypto** (*string*) – The phase 1 encryption value to update.
- **phase1\_dh** (*string*) – The phase 1 diffie hellman group value to update.
- **phase1\_key\_ttl** (*string*) – The phase 1 key life value to update.
- **phase2\_auth** (*string*) – The phase 2 authentication value to update.
- **phase2\_crypto** (*string*) – The phase 2 encryption value to update.
- **phase2\_df** (*string*) – The phase 2 diffie hellman group value to update.
- **phase2\_forward\_secriecy** (*string*) – The phase 2 perfect forward secrecy value to update.
- **phase2\_key\_ttl** (*string*) – The phase 2 key life value to update.

**Return bool** True if the update was successful.

### 3.2.13 SoftLayer.load\_balancer

Load Balancer Manager/helpers

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.load_balancer.LoadBalancerManager` (*client*)  
Manages SoftLayer load balancers.

See product information here: <https://www.ibm.com/cloud/load-balancer>

**Parameters** **client** (*SoftLayer.API.BaseClient*) – the client instance

**add\_lb\_17\_pool** (*identifier, pool, members, health, session*)  
Creates a new l7 pool for a LBaaS instance

- [https://sldn.softlayer.com/reference/services/SoftLayer\\_Network\\_LBaaS\\_L7Pool/createL7Pool/](https://sldn.softlayer.com/reference/services/SoftLayer_Network_LBaaS_L7Pool/createL7Pool/)
- <https://cloud.ibm.com/docs/infrastructure/loadbalancer-service?topic=loadbalancer-service-api-reference>

### Parameters

- **identifier** – UUID of the LBaaS instance
- **SoftLayer\_Network\_LBaaS\_L7Pool** (*pool*) – Description of the pool
- **SoftLayer\_Network\_LBaaS\_L7Member** [] (*members*) – Array of servers with their address, port, weight
- **SoftLayer\_Network\_LBaaS\_L7HealthMonitor** (*monitor*) – A health monitor
- **SoftLayer\_Network\_LBaaS\_L7SessionAffinity** (*session*) – Whether to use affinity

**add\_lb\_listener** (*identifier, listener*)

Adds or update a listener to a LBaaS instance

When using this to update a listener, just include the 'listenerUuid' in the listener object See the following for listener configuration options [https://sldn.softlayer.com/reference/datatypes/SoftLayer\\_Network\\_LBaaS\\_LoadBalancerProtocolConfiguration/](https://sldn.softlayer.com/reference/datatypes/SoftLayer_Network_LBaaS_LoadBalancerProtocolConfiguration/)

### Parameters

- **identifier** – UUID of the LBaaS instance
- **listener** – Object with all listener configurations

**add\_lb\_member** (*identifier, service\_info*)

Adds a member to a LBaaS instance

[https://sldn.softlayer.com/reference/services/SoftLayer\\_Network\\_LBaaS\\_Member/deleteLoadBalancerMembers/](https://sldn.softlayer.com/reference/services/SoftLayer_Network_LBaaS_Member/deleteLoadBalancerMembers/) :param identifier: UUID of the LBaaS instance :param service\_info: datatypes/SoftLayer\_Network\_LBaaS\_LoadBalancerServerInstanceInfo

**cancel\_lbaas** (*uuid*)

Cancels a LBaaS instance.

[https://sldn.softlayer.com/reference/services/SoftLayer\\_Network\\_LBaaS\\_LoadBalancer/cancelLoadBalancer/](https://sldn.softlayer.com/reference/services/SoftLayer_Network_LBaaS_LoadBalancer/cancelLoadBalancer/) :param uuid string: UUID of the LBaaS instance to cancel

**del\_lb\_l7\_pool** (*identifier*)

Deletes a l7 pool

**Parameters** **identifier** – Id of the L7Pool

**delete\_lb\_member** (*identifier, member\_id*)

Removes a member from a LBaaS instance

[https://sldn.softlayer.com/reference/services/SoftLayer\\_Network\\_LBaaS\\_Member/deleteLoadBalancerMembers/](https://sldn.softlayer.com/reference/services/SoftLayer_Network_LBaaS_Member/deleteLoadBalancerMembers/) :param identifier: UUID of the LBaaS instance :param member\_id: Member UUID to remove.

**get\_adc** (*identifier, mask=None*)

Returns a netscaler object.

**Returns** SoftLayer\_Network\_Application\_Delivery\_Controller.

**get\_adcs** (*mask=None*)

Returns a list of all netscalers.

**Returns** SoftLayer\_Network\_Application\_Delivery\_Controller[].

**get\_lb** (*identifier, mask=None*)

Returns a IBM Cloud LoadBalancer

**Returns** SoftLayer\_Network\_LBaaS\_LoadBalancer

**get\_lbaas** (*mask=None*)

Returns a list of IBM Cloud Loadbalancers

**Returns** SoftLayer\_Network\_LBaaS\_LoadBalancer[]

**get\_lbaas\_uuid\_id** (*identifier*)

Gets a LBaaS uuid, id. Since sometimes you need one or the other.

**Parameters identifier** – either the LB Id, or UUID, this function will return both.

**Return (uuid, id)**

**lbaas\_order\_options** ()

Gets the options to order a LBaaS instance.

**order\_lbaas** (*datacenter, name, desc, protocols, subnet\_id, public=False, verify=False*)

Allows to order a Load Balancer

**Parameters**

- **datacenter** – Shortname for the SoftLayer datacenter to order in.
- **name** – Identifier for the new LB.
- **desc** – Optional description for the lb.
- **protocols** – [https://sldn.softlayer.com/reference/datatypes/SoftLayer\\_Network\\_LBaaS\\_Listener/](https://sldn.softlayer.com/reference/datatypes/SoftLayer_Network_LBaaS_Listener/)
- **subnet\_id** – Id of the subnet for this new LB to live on.
- **public** – Use Public side for the backend.
- **verify** – Don't actually order if True.

**remove\_lb\_listener** (*identifier, listener*)

Removes a listener to a LBaaS instance

**Parameters**

- **identifier** – UUID of the LBaaS instance
- **listener** – UUID of the Listner to be removed.

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly 'identifier' can be depends on the resolvers

**Parameters identifier** (*string*) – identifying string

**Returns list**

**update\_lb\_health\_monitors** (*uuid, checks*)

calls SoftLayer\_Network\_LBaaS\_HealthMonitor::updateLoadBalancerHealthMonitors()

- [updateLoadBalancerHealthMonitors](#)

- `SoftLayer_Network_LBaaS_LoadBalancerHealthMonitorConfiguration`

#### Parameters

- **uuid** – loadBalancerUuid
- **list** (*checks*) – `SoftLayer_Network_LBaaS_LoadBalancerHealthMonitorConfiguration[]`

### 3.2.14 SoftLayer.metadata

Metadata Manager/helpers

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.metadata.MetadataManager` (*client=None, timeout=5*)  
Provides an interface for the SoftLayer metadata service.

See product information here: [http://sldn.softlayer.com/reference/services/SoftLayer\\_Resource\\_Metadata](http://sldn.softlayer.com/reference/services/SoftLayer_Resource_Metadata)

This provides metadata about the resource it is called from. See `METADATA_ATTRIBUTES` for full list of attributes.

Usage:

```
>>> import SoftLayer
>>> client = SoftLayer.create_client_from_env()
>>> from SoftLayer import MetadataManager
>>> meta = MetadataManager(client)
>>> meta.get('datacenter')
'dal105'
>>> meta.get('fqdn')
'test.example.com'
```

**Parameters** **client** (`SoftLayer.API.BaseClient`) – the client instance

**get** (*name, param=None*)  
Retrieve a metadata attribute.

#### Parameters

- **name** (*string*) – name of the attribute to retrieve. See *attrs*
- **param** – Required parameter for some attributes

**private\_network** (*\*\*kwargs*)  
Returns details about the private network.

#### Parameters

- **router** (*boolean*) – True to return router details
- **vlan** (*boolean*) – True to return vlan details
- **vlan\_ids** (*boolean*) – True to return vlan\_ids

**public\_network** (*\*\*kwargs*)  
Returns details about the public network.

#### Parameters

- **router** (*boolean*) – True to return router details

- **vlangs** (*boolean*) – True to return vlan details
- **vlan\_ids** (*boolean*) – True to return vlan\_ids

metadata.METADATA\_ATTRIBUTES = dict\_keys(['backend\_mac', 'datacenter', 'datacenter\_id', 'd

### 3.2.15 SoftLayer.network

Network Manager/helpers

**license** MIT, see LICENSE for more details.

**class** SoftLayer.managers.network.**NetworkManager** (*client*)

Manage SoftLayer network objects: VLANs, subnets, IPs and rwhois

See product information here: <http://www.softlayer.com/networking>

**Parameters** **client** (*SoftLayer.API.BaseClient*) – the client instance

**add\_global\_ip** (*version=4, test\_order=False*)

Adds a global IP address to the account.

#### Parameters

- **version** (*int*) – Specifies whether this is IPv4 or IPv6
- **test\_order** (*bool*) – If true, this will only verify the order.

**add\_securitygroup\_rule** (*group\_id, remote\_ip=None, remote\_group=None, direction=None, ethertype=None, port\_max=None, port\_min=None, protocol=None*)

Add a rule to a security group

#### Parameters

- **group\_id** (*int*) – The ID of the security group to add this rule to
- **remote\_ip** (*str*) – The remote IP or CIDR to enforce the rule on
- **remote\_group** (*int*) – The remote security group ID to enforce the rule on
- **direction** (*str*) – The direction to enforce (egress or ingress)
- **ethertype** (*str*) – The ethertype to enforce (IPv4 or IPv6)
- **port\_max** (*int*) – The upper port bound to enforce (icmp code if the protocol is icmp)
- **port\_min** (*int*) – The lower port bound to enforce (icmp type if the protocol is icmp)
- **protocol** (*str*) – The protocol to enforce (icmp, udp, tcp)

**add\_securitygroup\_rules** (*group\_id, rules*)

Add rules to a security group

#### Parameters

- **group\_id** (*int*) – The ID of the security group to add the rules to
- **rules** (*list*) – The list of rule dictionaries to add

**add\_subnet** (*subnet\_type, quantity=None, endpoint\_id=None, version=4, test\_order=False*)

Orders a new subnet

#### Parameters

- **subnet\_type** (*str*) – Type of subnet to add: private, public, global,static
- **quantity** (*int*) – Number of IPs in the subnet

- **endpoint\_id** (*int*) – id for the subnet to be placed into
- **version** (*int*) – 4 for IPv4, 6 for IPv6
- **test\_order** (*bool*) – If true, this will only verify the order.

**assign\_global\_ip** (*global\_ip\_id, target*)

Assigns a global IP address to a specified target.

**Parameters**

- **global\_ip\_id** (*int*) – The ID of the global IP being assigned
- **target** (*string*) – The IP address to assign

**attach\_securitygroup\_component** (*group\_id, component\_id*)

Attaches a network component to a security group.

**Parameters**

- **group\_id** (*int*) – The ID of the security group
- **component\_id** (*int*) – The ID of the network component to attach

**attach\_securitygroup\_components** (*group\_id, component\_ids*)

Attaches network components to a security group.

**Parameters**

- **group\_id** (*int*) – The ID of the security group
- **component\_ids** (*list*) – The IDs of the network components to attach

**cancel\_global\_ip** (*global\_ip\_id*)

Cancels the specified global IP address.

**Parameters** **id** (*int*) – The ID of the global IP to be cancelled.

**cancel\_subnet** (*subnet\_id*)

Cancels the specified subnet.

**Parameters** **subnet\_id** (*int*) – The ID of the subnet to be cancelled.

**create\_securitygroup** (*name=None, description=None*)

Creates a security group.

**Parameters**

- **name** (*string*) – The name of the security group
- **description** (*string*) – The description of the security group

**delete\_securitygroup** (*group\_id*)

Deletes the specified security group.

**Parameters** **group\_id** (*int*) – The ID of the security group

**detach\_securitygroup\_component** (*group\_id, component\_id*)

Detaches a network component from a security group.

**Parameters**

- **group\_id** (*int*) – The ID of the security group
- **component\_id** (*int*) – The ID of the component to detach

**detach\_securitygroup\_components** (*group\_id, component\_ids*)

Detaches network components from a security group.

**Parameters**

- **group\_id** (*int*) – The ID of the security group
- **component\_ids** (*list*) – The IDs of the network components to detach

**edit\_rwhois** (*abuse\_email=None, address1=None, address2=None, city=None, company\_name=None, country=None, first\_name=None, last\_name=None, postal\_code=None, private\_residence=None, state=None*)  
Edit rwhois record.

**edit\_securitygroup** (*group\_id, name=None, description=None*)  
Edit security group details.

**Parameters**

- **group\_id** (*int*) – The ID of the security group
- **name** (*string*) – The name of the security group
- **description** (*string*) – The description of the security group

**edit\_securitygroup\_rule** (*group\_id, rule\_id, remote\_ip=None, remote\_group=None, direction=None, ethertype=None, port\_max=None, port\_min=None, protocol=None*)  
Edit a security group rule.

**Parameters**

- **group\_id** (*int*) – The ID of the security group the rule belongs to
- **rule\_id** (*int*) – The ID of the rule to edit
- **remote\_ip** (*str*) – The remote IP or CIDR to enforce the rule on
- **remote\_group** (*int*) – The remote security group ID to enforce the rule on
- **direction** (*str*) – The direction to enforce (egress or ingress)
- **ethertype** (*str*) – The ethertype to enforce (IPv4 or IPv6)
- **port\_max** (*str*) – The upper port bound to enforce
- **port\_min** (*str*) – The lower port bound to enforce
- **protocol** (*str*) – The protocol to enforce (icmp, udp, tcp)

**get\_event\_logs\_by\_request\_id** (*request\_id*)  
Gets all event logs by the given request id

**Parameters** **request\_id** (*string*) – The request id we want to filter on

**get\_nas\_credentials** (*identifier, \*\*kwargs*)  
Returns a list of IDs of VLANs which match the given VLAN name.

**Parameters** **instance\_id** (*integer*) – the instance ID

**Returns** A dictionary containing a large amount of information about the specified instance.

**get\_rwhois** ()  
Returns the RWhois information about the current account.

**Returns** A dictionary containing the account's RWhois information.

**get\_securitygroup** (*group\_id, \*\*kwargs*)  
Returns the information about the given security group.

**Parameters** **id** (*string*) – The ID for the security group



**Returns** A dictionary of information about the security group

**get\_subnet** (*subnet\_id*, *\*\*kwargs*)

Returns information about a single subnet.

**Parameters** *id* (*string*) – Either the ID for the subnet or its network identifier

**Returns** A dictionary of information about the subnet

**get\_vlan** (*vlan\_id*)

Returns information about a single VLAN.

**Parameters** *id* (*int*) – The unique identifier for the VLAN

**Returns** A dictionary containing a large amount of information about the specified VLAN.

**ip\_lookup** (*ip\_address*)

Looks up an IP address and returns network information about it.

**Parameters** *ip\_address* (*string*) – An IP address. Can be IPv4 or IPv6

**Returns** A dictionary of information about the IP

**list\_global\_ips** (*version=None*, *identifier=None*, *\*\*kwargs*)

Returns a list of all global IP address records on the account.

**Parameters**

- **version** (*int*) – Only returns IPs of this version (4 or 6)
- **identifier** (*string*) – If specified, the list will only contain the global ips matching this network identifier.

**list\_securitygroup\_rules** (*group\_id*)

List security group rules associated with a security group.

**Parameters** *group\_id* (*int*) – The security group to list rules for

**list\_securitygroups** (*\*\*kwargs*)

List security groups.

**list\_subnets** (*identifier=None*, *datacenter=None*, *version=0*, *subnet\_type=None*, *network\_space=None*, *\*\*kwargs*)

Display a list of all subnets on the account.

This provides a quick overview of all subnets including information about data center residence and the number of devices attached.

**Parameters**

- **identifier** (*string*) – If specified, the list will only contain the subnet matching this network identifier.
- **datacenter** (*string*) – If specified, the list will only contain subnets in the specified data center.
- **version** (*int*) – Only returns subnets of this version (4 or 6).
- **subnet\_type** (*string*) – If specified, it will only returns subnets of this type.
- **network\_space** (*string*) – If specified, it will only returns subnets with the given address space label.
- **\*\*kwargs** (*dict*) – response-level options (mask, limit, etc.)

**list\_vlans** (*datacenter=None, vlan\_number=None, name=None, \*\*kwargs*)

Display a list of all VLANs on the account.

This provides a quick overview of all VLANs including information about data center residence and the number of devices attached.

**Parameters**

- **datacenter** (*string*) – If specified, the list will only contain VLANs in the specified data center.
- **vlan\_number** (*int*) – If specified, the list will only contain the VLAN matching this VLAN number.
- **name** (*int*) – If specified, the list will only contain the VLAN matching this VLAN name.
- **\*\*kwargs** (*dict*) – response-level options (mask, limit, etc.)

**remove\_securitygroup\_rule** (*group\_id, rule\_id*)

Remove a rule from a security group.

**Parameters**

- **group\_id** (*int*) – The ID of the security group
- **rule\_id** (*int*) – The ID of the rule to remove

**remove\_securitygroup\_rules** (*group\_id, rules*)

Remove rules from a security group.

**Parameters**

- **group\_id** (*int*) – The ID of the security group
- **rules** (*list*) – The list of IDs to remove

**resolve\_global\_ip\_ids** (*identifier*)

Resolve global ip ids.

**resolve\_subnet\_ids** (*identifier*)

Resolve subnet ids.

**resolve\_vlan\_ids** (*identifier*)

Resolve VLAN ids.

**summary\_by\_datacenter** ()

Summary of the networks on the account, grouped by data center.

The resultant dictionary is primarily useful for statistical purposes. It contains count information rather than raw data. If you want raw information, see the `list_vlans()` method instead.

**Returns** A dictionary keyed by data center with the data containing a set of counts for subnets, hardware, virtual servers, and other objects residing within that data center.

**unassign\_global\_ip** (*global\_ip\_id*)

Unassigns a global IP address from a target.

**Parameters** **id** (*int*) – The ID of the global IP being unassigned

## 3.2.16 SoftLayer.ordering

Ordering Manager

**license** MIT, see LICENSE for more details.

**class** SoftLayer.managers.ordering.**OrderingManager** (*client*)

Manager to help ordering via the SoftLayer API.

**Parameters** *client* (*SoftLayer.API.BaseClient*) – the client instance

**static filter\_outlet\_packages** (*packages*)

Remove packages designated as OUTLET.

Those type of packages must be handled in a different way, and they are not supported at the moment.

**Parameters** *packages* – Dictionary of packages. Name and description keys must be present in each of them.

**generate\_order** (*package\_keyname, location, item\_keynames, complex\_type=None, hourly=True, preset\_keyname=None, extras=None, quantity=1*)

Generates an order with the given package and prices.

This function takes in parameters needed for an order and generates an order dictionary. This dictionary can then be used in either verify or placeOrder().

#### Parameters

- **package\_keyname** (*str*) – The keyname for the package being ordered
- **location** (*str*) – The datacenter location string for ordering (Ex: DALLAS13)
- **item\_keynames** (*list*) – The list of item keyname strings to order. To see list of possible keynames for a package, use list\_items() (or *slcli order item-list*)
- **complex\_type** (*str*) – The complex type to send with the order. Typically begins with *SoftLayer\_Container\_Product\_Order\_*.
- **hourly** (*bool*) – If true, uses hourly billing, otherwise uses monthly billing
- **preset\_keyname** (*string*) – If needed, specifies a preset to use for that package. To see a list of possible keynames for a package, use list\_preset() (or *slcli order preset-list*)
- **extras** (*dict*) – The extra data for the order in dictionary format. Example: A VSI order requires hostname and domain to be set, so extras will look like the following: {'virtualGuests': [{'hostname': 'test', 'domain': 'softlayer.com'}]}
- **quantity** (*int*) – The number of resources to order

**generate\_order\_template** (*quote\_id, extra, quantity=1*)

Generate a complete order template.

#### Parameters

- **quote\_id** (*int*) – ID of target quote
- **extra** (*dictionary*) – Overrides for the defaults of *SoftLayer\_Container\_Product\_Order*
- **quantity** (*int*) – Number of items to order.

**get\_item\_capacity** (*items, item\_keynames*)

Get item capacity.

**static get\_item\_price\_id** (*core, prices*)

get item price id

**get\_item\_prices** (*package\_id*)

Get item prices.

Retrieve a SoftLayer\_Product\_Package item prices record.

**Parameters** *package\_id* (*int*) – package identifier.

**Returns** A list of price IDs associated with the given package.

**get\_location\_id** (*location*)

Finds the location ID of a given datacenter

This is mostly used so either a dc name, or regions keyname can be used when ordering :param str location: Region Keyname (DALLAS13) or datacenter name (dal13) :returns: integer id of the datacenter

**static get\_only\_active\_packages** (*packages*)

Return only active packages.

If a package is active, it is eligible for ordering This will inspect the ‘isActive’ property on the provided packages

**Parameters packages** – Dictionary of packages, isActive key must be present

**get\_order\_container** (*quote\_id*)

Generate an order container from a quote object.

**Parameters quote\_id** – ID number of target quote

**get\_package\_by\_key** (*package\_keyname, mask=None*)

Get a single package with a given key.

If no packages are found, returns None

**Parameters**

- **package\_keyname** – string representing the package key name we are interested in.
- **mask** (*string*) – Mask to specify the properties we want to retrieve

**get\_package\_by\_type** (*package\_type, mask=None*)

Get a single package of a given type.

Syntactic sugar to retrieve a single package of a given type. If multiple packages share the given type, this will return the first one returned by the API. If no packages are found, returns None

**Parameters package\_type** (*string*) – representing the package type key name we are interested in

**get\_package\_id\_by\_type** (*package\_type*)

Return the package ID of a Product Package with a given type.

**Parameters package\_type** (*string*) – representing the package type key name we are interested in

**Raises ValueError** – when no package of the given type is found

**get\_packages\_of\_type** (*package\_types, mask=None*)

Get packages that match a certain type.

Each ordering package has a type, so return all packages that match the types we are looking for

**Parameters**

- **package\_types** (*list*) – List of strings representing the package type keynames we are interested in.
- **mask** (*string*) – Mask to specify the properties we want to retrieve

**get\_preset\_by\_key** (*package\_keyname, preset\_keyname, mask=None*)

Gets a single preset with the given key.

**get\_preset\_prices** (*preset*)

Get preset item prices.

Retrieve a SoftLayer\_Product\_Package\_Preset record.

**Parameters** **preset** (*int*) – preset identifier.

**Returns** A list of price IDs associated with the given `preset_id`.

**get\_price\_id\_list** (*package\_keyname, item\_keynames, core=None*)

Converts a list of item keynames to a list of price IDs.

This function is used to convert a list of item keynames into a list of price IDs that are used in the Product\_Order `verifyOrder()` and `placeOrder()` functions.

**Parameters**

- **package\_keyname** (*str*) – The package associated with the prices
- **item\_keynames** (*list*) – A list of item keyname strings
- **core** (*str*) – preset guest core capacity.

**Returns** A list of price IDs associated with the given item keynames in the given package

**get\_quote\_details** (*quote\_id*)

Retrieve quote details.

**Parameters** **quote\_id** – ID number of target quote

**get\_quotes** ()

Retrieve a list of active quotes.

**Returns** a list of SoftLayer\_Billing\_Order\_Quote

**list\_categories** (*package\_keyname, \*\*kwargs*)

List the categories for the given package.

**Parameters** **package\_keyname** (*str*) – The package for which to get the categories.

**Returns** List of categories associated with the package

**list\_items** (*package\_keyname, \*\*kwargs*)

List the items for the given package.

**Parameters** **package\_keyname** (*str*) – The package for which to get the items.

**Returns** List of items in the package

**list\_packages** (*\*\*kwargs*)

List active packages.

**Returns** List of active packages.

**list\_presets** (*package\_keyname, \*\*kwargs*)

Gets active presets for the given package.

**Parameters** **package\_keyname** (*str*) – The package for which to get presets

**Returns** A list of package presets that can be used for ordering

**order\_quote** (*quote\_id, extra*)

Places an order using a quote

```
extras = {
    'hardware': {'hostname': 'test', 'domain': 'testing.com'},
    'quantity': 2
}
manager = ordering.OrderingManager(env.client)
result = manager.order_quote(12345, extras)
```

### Parameters

- **quote\_id** (*int*) – ID for the target quote
- **extra** (*dictionary*) – Overrides for the defaults of SoftLayer\_Container\_Product\_Order
- **quantity** (*int*) – Quantity to override default

**package\_locations** (*package\_keyname*)

List datacenter locations for a package keyname

**Parameters** **package\_keyname** (*str*) – The package for which to get the items.

**Returns** List of locations a package is orderable in

**place\_order** (*package\_keyname, location, item\_keynames, complex\_type=None, hourly=True, preset\_keyname=None, extras=None, quantity=1*)

Places an order with the given package and prices.

This function takes in parameters needed for an order and places the order.

### Parameters

- **package\_keyname** (*str*) – The keyname for the package being ordered
- **location** (*str*) – The datacenter location string for ordering (Ex: DALLAS13)
- **item\_keynames** (*list*) – The list of item keyname strings to order. To see list of possible keynames for a package, use `list_items()` (or `slcli order item-list`)
- **complex\_type** (*str*) – The complex type to send with the order. Typically begins with `SoftLayer_Container_Product_Order_`.
- **hourly** (*bool*) – If true, uses hourly billing, otherwise uses monthly billing
- **preset\_keyname** (*string*) – If needed, specifies a preset to use for that package. To see a list of possible keynames for a package, use `list_preset()` (or `slcli order preset-list`)
- **extras** (*dict*) – The extra data for the order in dictionary format. Example: A VSI order requires hostname and domain to be set, so extras will look like the following: `{'virtualGuests': [{'hostname': 'test', 'domain': 'softlayer.com'}]}`
- **quantity** (*int*) – The number of resources to order

**place\_quote** (*package\_keyname, location, item\_keynames, complex\_type=None, preset\_keyname=None, extras=None, quantity=1, quote\_name=None, send\_email=False*)

Place a quote with the given package and prices.

This function takes in parameters needed for an order and places the quote.

### Parameters

- **package\_keyname** (*str*) – The keyname for the package being ordered
- **location** (*str*) – The datacenter location string for ordering (Ex: DALLAS13)

- **item\_keynames** (*list*) – The list of item keyname strings to order. To see list of possible keynames for a package, use `list_items()` (or *slcli order item-list*)
- **complex\_type** (*str*) – The complex type to send with the order. Typically begins with *SoftLayer\_Container\_Product\_Order\_.*
- **preset\_keyname** (*string*) – If needed, specifies a preset to use for that package. To see a list of possible keynames for a package, use `list_preset()` (or *slcli order preset-list*)
- **extras** (*dict*) – The extra data for the order in dictionary format. Example: A VSI order requires hostname and domain to be set, so extras will look like the following: `{'virtualGuests': [{'hostname': 'test', 'domain': 'softlayer.com'}]}`
- **quantity** (*int*) – The number of resources to order
- **quote\_name** (*string*) – A custom name to be assigned to the quote (optional).
- **send\_email** (*bool*) – This flag indicates that the quote should be sent to the email address associated with the account or order.

**verify\_order** (*package\_keyname, location, item\_keynames, complex\_type=None, hourly=True, preset\_keyname=None, extras=None, quantity=1*)

Verifies an order with the given package and prices.

This function takes in parameters needed for an order and verifies the order to ensure the given items are compatible with the given package.

#### Parameters

- **package\_keyname** (*str*) – The keyname for the package being ordered
- **location** (*str*) – The datacenter location string for ordering (Ex: DALLAS13)
- **item\_keynames** (*list*) – The list of item keyname strings to order. To see list of possible keynames for a package, use `list_items()` (or *slcli order item-list*)
- **complex\_type** (*str*) – The complex type to send with the order. Typically begins with *SoftLayer\_Container\_Product\_Order\_.*
- **hourly** (*bool*) – If true, uses hourly billing, otherwise uses monthly billing
- **preset\_keyname** (*string*) – If needed, specifies a preset to use for that package. To see a list of possible keynames for a package, use `list_preset()` (or *slcli order preset-list*)
- **extras** (*dict*) – The extra data for the order in dictionary format. Example: A VSI order requires hostname and domain to be set, so extras will look like the following: `{'virtualGuests': [{'hostname': 'test', 'domain': 'softlayer.com'}]}`
- **quantity** (*int*) – The number of resources to order

**verify\_quote** (*quote\_id, extra*)

Verifies that a quote order is valid.

```
extras = {
    'hardware': {'hostname': 'test', 'domain': 'testing.com'},
    'quantity': 2
}
manager = ordering.OrderingManager(env.client)
result = manager.verify_quote(12345, extras)
```

#### Parameters

- **quote\_id** (*int*) – ID for the target quote

- **extra** (*dictionary*) – Overrides for the defaults of SoftLayer\_Container\_Product\_Order
- **quantity** (*int*) – Quantity to override default

### 3.2.17 SoftLayer.sshkey

SSH Key Manager/helpers

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.sshkey.SshKeyManager` (*client*)

Manages account SSH keys in SoftLayer.

See product information here: <https://knowledgelayer.softlayer.com/procedure/ssh-keys>

**Parameters** **client** (*SoftLayer.API.BaseClient*) – the client instance

**add\_key** (*key, label, notes=None*)

Adds a new SSH key to the account.

**Parameters**

- **key** (*string*) – The SSH key to add
- **label** (*string*) – The label for the key
- **notes** (*string*) – Additional notes for the key

**Returns** A dictionary of the new key's information.

**delete\_key** (*key\_id*)

Permanently deletes an SSH key from the account.

**Parameters** **key\_id** (*int*) – The ID of the key to delete

**edit\_key** (*key\_id, label=None, notes=None*)

Edits information about an SSH key.

**Parameters**

- **key\_id** (*int*) – The ID of the key to edit
- **label** (*string*) – The new label for the key
- **notes** (*string*) – Notes to set or change on the key

**Returns** A Boolean indicating success or failure

**get\_key** (*key\_id*)

Returns full information about a single SSH key.

**Parameters** **key\_id** (*int*) – The ID of the key to retrieve

**Returns** A dictionary of information about the key

**list\_keys** (*label=None*)

Lists all SSH keys on the account.

**Parameters** **label** (*string*) – Filter list based on SSH key label

**Returns** A list of dictionaries with information about each key



**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly 'identifier' can be depends on the resolvers

**Parameters** *identifier* (*string*) – identifying string

**Returns** list

### 3.2.18 SoftLayer.ssl

SSL Manager/helpers

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.ssl.SSLManager` (*client*)

Manages SSL certificates in SoftLayer.

See product information here: <http://www.softlayer.com/ssl-certificates>

Example:

```
# Initialize the Manager.
# env variables. These can also be specified in ~/.softlayer,
# or passed directly to SoftLayer.Client()
# SL_USERNAME = YOUR_USERNAME
# SL_API_KEY = YOUR_API_KEY
import SoftLayer
client = SoftLayer.Client()
mgr = SoftLayer.SSLManager(client)
```

**Parameters** *client* (*SoftLayer.API.BaseClient*) – the client instance

**add\_certificate** (*certificate*)

Creates a new certificate.

**Parameters** *certificate* (*dict*) – A dictionary representing the parts of the certificate. See [sldn.softlayer.com](http://sldn.softlayer.com) for more info.

Example:

```
cert = ??
result = mgr.add_certificate(certificate=cert)
```

**edit\_certificate** (*certificate*)

Updates a certificate with the included options.

The provided dict must include an 'id' key and value corresponding to the certificate ID that should be updated.

**Parameters** *certificate* (*dict*) – the certificate to update.

Example:

```
# Updates the cert id 1234
cert['id'] = 1234
cert['certificate'] = ??
result = mgr.edit_certificate(certificate=cert)
```

**get\_certificate** (*cert\_id*)

Gets a certificate with the ID specified.

**Parameters** **cert\_id** (*integer*) – the certificate ID to retrieve

Example:

```
cert = mgr.get_certificate(cert_id=1234)
print(cert)
```

**list\_certs** (*method='all'*)

List all certificates.

**Parameters** **method** (*string*) – The type of certificates to list. Options are ‘all’, ‘expired’, and ‘valid’.

**Returns** A list of dictionaries representing the requested SSL certs.

Example:

```
# Get all valid SSL certs
certs = mgr.list_certs(method='valid')
print(certs)
```

**remove\_certificate** (*cert\_id*)

Removes a certificate.

**Parameters** **cert\_id** (*integer*) – a certificate ID to remove

Example:

```
# Removes certificate with id 1234
result = mgr.remove_certificate(cert_id = 1234)
```

### 3.2.19 SoftLayer.ticket

Ticket Manager/helpers

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.ticket.TicketManager` (*client*)

Manages SoftLayer support tickets.

See product information here: <http://www.softlayer.com/support>

**Parameters** **client** (*SoftLayer.API.BaseClient*) – the client instance

**attach\_hardware** (*ticket\_id=None, hardware\_id=None*)

Attach hardware to a ticket.

**Parameters**

- **ticket\_id** (*integer*) – the id of the ticket to attach to
- **hardware\_id** (*integer*) – the id of the hardware to attach

**Returns** dict – The new ticket attachment

**attach\_virtual\_server** (*ticket\_id=None, virtual\_id=None*)

Attach a virtual server to a ticket.

**Parameters**

- **ticket\_id** (*integer*) – the id of the ticket to attach to
- **virtual\_id** (*integer*) – the id of the virtual server to attach

**Returns** dict – The new ticket attachment

**create\_ticket** (*title=None, body=None, subject=None, priority=None*)

Create a new ticket.

**Parameters**

- **title** (*string*) – title for the new ticket
- **body** (*string*) – body for the new ticket
- **subject** (*integer*) – id of the subject to be assigned to the ticket
- **priority** (*integer*) – Value from 1 (highest) to 4 (lowest)

**detach\_hardware** (*ticket\_id=None, hardware\_id=None*)

Detach hardware from a ticket.

**Parameters**

- **ticket\_id** – the id of the ticket to detach from
- **hardware\_id** – the id of the hardware to detach

**Returns** bool – Whether the detachment was successful

**detach\_virtual\_server** (*ticket\_id=None, virtual\_id=None*)

Detach a virtual server from a ticket.

**Parameters**

- **ticket\_id** – the id of the ticket to detach from
- **virtual\_id** – the id of the virtual server to detach

**Returns** bool – Whether the detachment was successful

**get\_ticket** (*ticket\_id*)

Get details about a ticket.

**Parameters** **ticket\_id** (*integer*) – the ticket ID

**Returns** dict – information about the specified ticket

**list\_subjects** ()

List all ticket subjects.

**list\_tickets** (*open\_status=True, closed\_status=True*)

List all tickets.

**Parameters**

- **open\_status** (*boolean*) – include open tickets
- **closed\_status** (*boolean*) – include closed tickets

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly 'identifier' can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

**update\_ticket** (*ticket\_id=None, body=None*)

Update a ticket.

**Parameters**

- **ticket\_id** (*integer*) – the id of the ticket to update
- **body** (*string*) – entry to update in the ticket

**upload\_attachment** (*ticket\_id=None, file\_path=None, file\_name=None*)

Upload an attachment to a ticket.

**Parameters**

- **ticket\_id** (*integer*) – the id of the ticket to upload the attachment to
- **file\_path** (*string*) – The path of the attachment to be uploaded
- **file\_name** (*string*) – The name of the attachment shown in the ticket

**Returns** dict – The uploaded attachment

### 3.2.20 SoftLayer.vs

VS Manager/helpers

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.managers.vs.VSManager` (*client, ordering\_manager=None*)  
Manages SoftLayer Virtual Servers.

See product information here: <http://www.softlayer.com/virtual-servers>

Example:

```
# Initialize the VSManager.
# env variables. These can also be specified in ~/.softlayer,
# or passed directly to SoftLayer.Client()
# SL_USERNAME = YOUR_USERNAME
# SL_API_KEY = YOUR_API_KEY
import SoftLayer
client = SoftLayer.Client()
mgr = SoftLayer.VSManager(client)
```

**Parameters**

- **client** (*SoftLayer.API.BaseClient*) – the client instance
- **ordering\_manager** (*SoftLayer.managers.OrderingManager*) – an optional manager to handle ordering. If none is provided, one will be auto initialized.

**cancel\_instance** (*instance\_id*)

Cancel an instance immediately, deleting all its data.

**Parameters** **instance\_id** (*integer*) – the instance ID to cancel

Example:

```
# Cancels instance 12345
mgr.cancel_instance(12345)
```

**capture** (*instance\_id*, *name*, *additional\_disks=False*, *notes=None*)

Capture one or all disks from a VS to a SoftLayer image.

Parameters set to None will be ignored and not attempted to be updated.

#### Parameters

- **instance\_id** (*integer*) – the instance ID to edit
- **name** (*string*) – name assigned to the image
- **additional\_disks** (*bool*) – set to true to include all additional attached storage devices
- **notes** (*string*) – notes about this particular image

**Returns** dictionary – information about the capture transaction.

**Example::** name = “Testing Images” notes = “Some notes about this image” result = mgr.capture(instance\_id=12345, name=name, notes=notes)

**change\_port\_speed** (*instance\_id*, *public*, *speed*)

Allows you to change the port speed of a virtual server’s NICs.

Example:

```
#change the Public interface to 10Mbps on instance 12345
result = mgr.change_port_speed(instance_id=12345,
                               public=True, speed=10)
# result will be True or an Exception
```

#### Parameters

- **instance\_id** (*int*) – The ID of the VS
- **public** (*bool*) – Flag to indicate which interface to change. True (default) means the public interface. False indicates the private interface.
- **speed** (*int*) – The port speed to set.

**Warning:** A port speed of 0 will disable the interface.

**create\_instance** (\*\**kwargs*)

Creates a new virtual server instance.

**Warning:** This will add charges to your account

Example:

```
new_vsi = {
    'domain': u'test01.labs.sftlyr.ws',
    'hostname': u'minion05',
    'datacenter': u'hkg02',
    'flavor': 'BL1_1X2X100'
    'dedicated': False,
    'private': False,
```

(continues on next page)

(continued from previous page)

```
'os_code' : u'UBUNTU_LATEST',
'hourly': True,
'ssh_keys': [1234],
'disks': ('100', '25'),
'local_disk': True,
'tags': 'test, pleaseCancel',
'public_security_groups': [12, 15]
}

vsi = mgr.create_instance(**new_vsi)
# vsi will have the newly created vsi details if done properly.
print vsi
```

### Parameters

- **cpus** (*int*) – The number of virtual CPUs to include in the instance.
- **memory** (*int*) – The amount of RAM to order.
- **hourly** (*bool*) – Flag to indicate if this server should be billed hourly (default) or monthly.
- **hostname** (*string*) – The hostname to use for the new server.
- **domain** (*string*) – The domain to use for the new server.
- **local\_disk** (*bool*) – Flag to indicate if this should be a local disk (default) or a SAN disk.
- **datacenter** (*string*) – The short name of the data center in which the VS should reside.
- **os\_code** (*string*) – The operating system to use. Cannot be specified if `image_id` is specified.
- **image\_id** (*int*) – The GUID of the image to load onto the server. Cannot be specified if `os_code` is specified.
- **dedicated** (*bool*) – Flag to indicate if this should be housed on adedicated or shared host (default). This will incur a fee on your account.
- **public\_vlan** (*int*) – The ID of the public VLAN on which you want this VS placed.
- **public\_security\_groups** (*list*) – The list of security group IDs to apply to the public interface
- **private\_security\_groups** (*list*) – The list of security group IDs to apply to the private interface
- **private\_vlan** (*int*) – The ID of the private VLAN on which you want this VS placed.
- **disks** (*list*) – A list of disk capacities for this server.
- **post\_uri** (*string*) – The URI of the post-install script to run after reload
- **private** (*bool*) – If true, the VS will be provisioned only with access to the private network. Defaults to false
- **ssh\_keys** (*list*) – The SSH keys to add to the root user
- **nic\_speed** (*int*) – The port speed to set
- **tags** (*string*) – tags to set on the VS as a comma separated list

- **flavor** (*string*) – The key name of the public virtual server flavor being ordered.
- **host\_id** (*int*) – The host id of a dedicated host to provision a dedicated host virtual server on.

### **create\_instances** (*config\_list*)

Creates multiple virtual server instances.

This takes a list of dictionaries using the same arguments as `create_instance()`.

**Warning:** This will add charges to your account

Example:

```
# Define the instance we want to create.
new_vsi = {
    'domain': u'test01.labs.sftlyr.ws',
    'hostname': u'minion05',
    'datacenter': u'hkg02',
    'flavor': 'BL1_1X2X100'
    'dedicated': False,
    'private': False,
    'os_code' : u'UBUNTU_LATEST',
    'hourly': True,
    'ssh_keys': [1234],
    'disks': ('100','25'),
    'local_disk': True,
    'tags': 'test, pleaseCancel',
    'public_security_groups': [12, 15]
}

# using .copy() so we can make changes to individual nodes
instances = [new_vsi.copy(), new_vsi.copy(), new_vsi.copy()]

# give each its own hostname, not required.
instances[0]['hostname'] = "multi-test01"
instances[1]['hostname'] = "multi-test02"
instances[2]['hostname'] = "multi-test03"

vsi = mgr.create_instances(config_list=instances)
#vsi will be a dictionary of all the new virtual servers
print vsi
```

### **edit** (*instance\_id*, *userdata=None*, *hostname=None*, *domain=None*, *notes=None*, *tags=None*)

Edit hostname, domain name, notes, and/or the user data of a VS.

Parameters set to None will be ignored and not attempted to be updated.

#### Parameters

- **instance\_id** (*integer*) – the instance ID to edit
- **userdata** (*string*) – user data on VS to edit. If none exist it will be created
- **hostname** (*string*) – valid hostname
- **domain** (*string*) – valid domain namem
- **notes** (*string*) – notes about this particular VS

- **tags** (*string*) – tags to set on the VS as a comma separated list. Use the empty string to remove all tags.

**Returns** bool – True or an Exception

**Example::** # Change the hostname on instance 12345 to ‘something’ result = mgr.edit(instance\_id=12345, hostname=”something”) #result will be True or an Exception

**get\_bandwidth\_allocation** (*instance\_id*)

Combines getBandwidthAllotmentDetail() and getBillingCycleBandwidthUsage()

**get\_bandwidth\_data** (*instance\_id*, *start\_date=None*, *end\_date=None*, *direction=None*, *rollup=3600*)

Gets bandwidth data for a server

Will get averaged bandwidth data for a given time period. If you use a rollup over 3600 be aware that the API will bump your start/end date to align with how data is stored. For example if you have a rollup of 86400 your start\_date will be bumped to 00:00. If you are not using a time in the start/end date fields, this won’t really matter.

#### Parameters

- **instance\_id** (*int*) – Hardware Id to get data for
- **start\_date** (*date*) – Date to start pulling data for.
- **end\_date** (*date*) – Date to finish pulling data for
- **direction** (*string*) – Can be either ‘public’, ‘private’, or None for both.
- **rollup** (*int*) – 300, 600, 1800, 3600, 43200 or 86400 seconds to average data over.

**get\_create\_options** ()

Retrieves the available options for creating a VS.

**Returns** A dictionary of creation options.

Example:

```
# Prints out the create option dictionary
options = mgr.get_create_options()
print(options)
```

**get\_instance** (*instance\_id*, *\*\*kwargs*)

Get details about a virtual server instance.

**Parameters** **instance\_id** (*integer*) – the instance ID

**Returns** A dictionary containing a large amount of information about the specified instance.

Example:

```
# Print out instance ID 12345.
vsi = mgr.get_instance(12345)
print vsi

# Print out only FQDN and primaryIP for instance 12345
object_mask = "mask[fullyQualifiedDomainName,primaryIpAddress]"
vsi = mgr.get_instance(12345, mask=mask)
print vsi
```



**get\_summary\_data\_usage** (*instance\_id*, *start\_date=None*, *end\_date=None*, *valid\_type=None*, *summary\_period=None*)

Retrieve the usage information of a virtual server.

#### Parameters

- **instance\_id** (*string*) – a string identifier used to resolve ids
- **start\_date** (*string*) – the start data to retrieve the vs usage information
- **end\_date** (*string*) – the start data to retrieve the vs usage information
- **string valid\_type** (*string*) – the Metric\_Data\_Type keyName.
- **summary\_period** (*int*) – summary period.

**get\_tracking\_id** (*instance\_id*)

Returns the Metric Tracking Object Id for a hardware server

**Parameters** **instance\_id** (*int*) – Id of the hardware server

**list\_instances** (*hourly=True*, *monthly=True*, *tags=None*, *cpus=None*, *memory=None*, *hostname=None*, *domain=None*, *local\_disk=None*, *datacenter=None*, *nic\_speed=None*, *public\_ip=None*, *private\_ip=None*, *transient=None*, *\*\*kwargs*)

Retrieve a list of all virtual servers on the account.

Example:

```
# Print out a list of hourly instances in the DAL05 data center.
for vsi in mgr.list_instances(hourly=True, datacenter='dal05'):
    print vsi['fullyQualifiedDomainName'], vsi['primaryIpAddress']

# Using a custom object-mask. Will get ONLY what is specified
object_mask = "mask[hostname,monitoringRobot[robotStatus]]"
for vsi in mgr.list_instances(mask=object_mask, hourly=True):
    print vsi
```

#### Parameters

- **hourly** (*boolean*) – include hourly instances
- **monthly** (*boolean*) – include monthly instances
- **tags** (*list*) – filter based on list of tags
- **cpus** (*integer*) – filter based on number of CPUS
- **memory** (*integer*) – filter based on amount of memory
- **hostname** (*string*) – filter based on hostname
- **domain** (*string*) – filter based on domain
- **local\_disk** (*string*) – filter based on local\_disk
- **datacenter** (*string*) – filter based on datacenter
- **nic\_speed** (*integer*) – filter based on network speed (in MBPS)
- **public\_ip** (*string*) – filter based on public ip address
- **private\_ip** (*string*) – filter based on private ip address
- **transient** (*boolean*) – filter on transient or non-transient instances
- **\*\*kwargs** (*dict*) – response-level options (mask, limit, etc.)

**Returns** Returns a list of dictionaries representing the matching virtual servers

**order\_guest** (*guest\_object*, *test=False*)

Uses Product\_Order::placeOrder to create a virtual guest.

Useful when creating a virtual guest with options not supported by Virtual\_Guest::createObject specifically ipv6 support.

**Parameters** **guest\_object** (*dictionary*) – See SoftLayer.CLI.virt.create.\_parse\_create\_args

Example:

```
new_vsi = {
    'domain': u'test01.labs.sftlyr.ws',
    'hostname': u'minion05',
    'datacenter': u'hkg02',
    'flavor': 'BL1_1X2X100'
    'dedicated': False,
    'private': False,
    'transient': False,
    'os_code' : u'UBUNTU_LATEST',
    'hourly': True,
    'ssh_keys': [1234],
    'disks': ('100', '25'),
    'local_disk': True,
    'tags': 'test, pleaseCancel',
    'public_security_groups': [12, 15],
    'ipv6': True
}

vsi = mgr.order_guest(new_vsi)
# vsi will have the newly created vsi receipt.
# vsi['orderDetails']['virtualGuests'] will be an array of created Guests
print vsi
```

**reload\_instance** (*instance\_id*, *post\_uri=None*, *ssh\_keys=None*, *image\_id=None*)

Perform an OS reload of an instance.

#### Parameters

- **instance\_id** (*integer*) – the instance ID to reload
- **post\_url** (*string*) – The URI of the post-install script to run after reload
- **ssh\_keys** (*list*) – The SSH keys to add to the root user
- **image\_id** (*int*) – The GUID of the image to load onto the server

**Warning:** This will reformat the primary drive. Post-provision script MUST be HTTPS for it to be executed.

Example:

```
# Reload instance ID 12345 then run a custom post-provision script.
# Post-provision script MUST be HTTPS for it to be executed.
post_uri = 'https://somehost.com/bootstrap.sh'
vsi = mgr.reload_instance(12345, post_uri=post_url)
```

**rescue** (*instance\_id*)

Reboot a VSI into the Xen rescue kernel.

**Parameters** *instance\_id* (*integer*) – the instance ID to rescue

**Returns** bool – True or an Exception

**Example::** # Puts instance 12345 into rescue mode result = mgr.rescue(instance\_id=12345)

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly ‘identifier’ can be depends on the resolvers

**Parameters** *identifier* (*string*) – identifying string

**Returns** list

**set\_tags** (*tags*, *guest\_id*)

Sets tags on a guest with a retry decorator

Just calls guest.setTags, but if it fails from an APIError will retry

**upgrade** (*instance\_id*, *cpus=None*, *memory=None*, *nic\_speed=None*, *public=True*, *preset=None*)

Upgrades a VS instance.

Example:

```
# Upgrade instance 12345 to 4 CPUs and 4 GB of memory
import SoftLayer
client = SoftLayer.create_client_from_env()
mgr = SoftLayer.VSManager(client)
mgr.upgrade(12345, cpus=4, memory=4)
```

**Parameters**

- **instance\_id** (*int*) – Instance id of the VS to be upgraded
- **cpus** (*int*) – The number of virtual CPUs to upgrade to of a VS instance.
- **preset** (*string*) – preset assigned to the vsi
- **memory** (*int*) – RAM of the VS to be upgraded to.
- **nic\_speed** (*int*) – The port speed to set
- **public** (*bool*) – CPU will be in Private/Public Node.

**Returns** bool

**verify\_create\_instance** (*\*\*kwargs*)

Verifies an instance creation command.

Without actually placing an order. See [create\\_instance\(\)](#) for a list of available options.

Example:

```
new_vsi = {
    'domain': u'test01.labs.sftlyr.ws',
    'hostname': u'minion05',
    'datacenter': u'hkg02',
    'flavor': 'BL1_1X2X100'
    'dedicated': False,
```

(continues on next page)

(continued from previous page)

```

'private': False,
'transient': False,
'os_code' : u'UBUNTU_LATEST',
'hourly': True,
'ssh_keys': [1234],
'disks': ('100','25'),
'local_disk': True,
'tags': 'test, pleaseCancel',
'public_security_groups': [12, 15]
}

vsi = mgr.verify_create_instance(**new_vsi)
# vsi will be a SoftLayer_Container_Product_Order_Virtual_Guest
# if your order is correct. Otherwise you will get an exception
print vsi

```

**wait\_for\_ready** (*instance\_id*, *limit=3600*, *delay=10*, *pending=False*)

Determine if a VS is ready and available.

In some cases though, that can mean that no transactions are running. The default arguments imply a VS is operational and ready for use by having network connectivity and remote access is available. Setting *pending=True* will ensure future API calls against this instance will not error due to pending transactions such as OS Reloads and cancellations.

#### Parameters

- **instance\_id** (*int*) – The instance ID with the pending transaction
- **limit** (*int*) – The maximum amount of seconds to wait.
- **delay** (*int*) – The number of seconds to sleep before checks. Defaults to 10.
- **pending** (*bool*) – Wait for pending transactions not related to provisioning or reloads such as monitoring.

Example:

```

# Will return once vsi 12345 is ready, or after 10 checks
ready = mgr.wait_for_ready(12345, 10)

```

**wait\_for\_transaction** (*instance\_id*, *limit*, *delay=10*)

Waits on a VS transaction for the specified amount of time.

This is really just a wrapper for `wait_for_ready(pending=True)`. Provided for backwards compatibility.

#### Parameters

- **instance\_id** (*int*) – The instance ID with the pending transaction
- **limit** (*int*) – The maximum amount of time to wait.
- **delay** (*int*) – The number of seconds to sleep before checks. Defaults to 10.

### 3.2.21 SoftLayer.vs\_capacity

Reserved Capacity Manager and helpers

**license** MIT, see License for more details.

**class** SoftLayer.managers.vs\_capacity.CapacityManager(*client*, *ordering\_manager=None*)

Manages SoftLayer Reserved Capacity Groups.

Product Information

- [https://console.bluemix.net/docs/vsi/vsi\\_about\\_reserved.html](https://console.bluemix.net/docs/vsi/vsi_about_reserved.html)
- [https://softlayer.github.io/reference/services/SoftLayer\\_Virtual\\_ReservedCapacityGroup/](https://softlayer.github.io/reference/services/SoftLayer_Virtual_ReservedCapacityGroup/)
- [https://softlayer.github.io/reference/services/SoftLayer\\_Virtual\\_ReservedCapacityGroup\\_Instance/](https://softlayer.github.io/reference/services/SoftLayer_Virtual_ReservedCapacityGroup_Instance/)

**Parameters**

- **client** (*SoftLayer.API.BaseClient*) – the client instance
- **ordering\_manager** (*SoftLayer.managers.OrderingManager*) – an optional manager to handle ordering. If none is provided, one will be auto initialized.

**create** (*name, backend\_router\_id, flavor, instances, test=False*)

Orders a Virtual\_ReservedCapacityGroup

**Parameters**

- **name** (*string*) – Name for the new reserved capacity
- **backend\_router\_id** (*int*) – This selects the pod. See create\_options for a list
- **flavor** (*string*) – Capacity KeyName, see create\_options for a list
- **instances** (*int*) – Number of guest this capacity can support
- **test** (*bool*) – If True, don't actually order, just test.

**create\_guest** (*capacity\_id, test, guest\_object*)

Turns an empty Reserve Capacity into a real Virtual Guest

**Parameters**

- **capacity\_id** (*int*) – ID of the RESERVED\_CAPACITY\_GROUP to create this guest into
- **test** (*bool*) – True will use verifyOrder, False will use placeOrder
- **guest\_object** (*dictionary*) – Below is the minimum info you need to send in guest\_object = { 'domain': 'test.com', 'hostname': 'A1538172419', 'os\_code': 'UBUNTU\_LATEST\_64', 'primary\_disk': '25', }

**get\_available\_routers** (*dc=None*)

Pulls down all backendRouterIds that are available

**Parameters** *dc* (*string*) – A specific location to get routers for, like 'dal13'.

**Returns list** A list of locations where RESERVED\_CAPACITY can be ordered.

**get\_create\_options** ()

List available reserved capacity plans

**get\_object** (*identifier, mask=None*)

Get a Reserved Capacity Group

**Parameters**

- **identifier** (*int*) – Id of the SoftLayer\_Virtual\_ReservedCapacityGroup
- **mask** (*string*) – override default object Mask

**list** ()

List Reserved Capacities

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly 'identifier' can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

### 3.2.22 SoftLayer.vs\_placement

Placement Group Manager

**license** MIT, see License for more details.

**class** `SoftLayer.managers.vs_placement.PlacementManager` (*client*)

Manages SoftLayer Reserved Capacity Groups.

Product Information

- [https://console.test.cloud.ibm.com/docs/vsi/vsi\\_placegroup.html#placement-groups](https://console.test.cloud.ibm.com/docs/vsi/vsi_placegroup.html#placement-groups)
- [https://softlayer.github.io/reference/services/SoftLayer\\_Account/getPlacementGroups/](https://softlayer.github.io/reference/services/SoftLayer_Account/getPlacementGroups/)
- [https://softlayer.github.io/reference/services/SoftLayer\\_Virtual\\_PlacementGroup\\_Rule/](https://softlayer.github.io/reference/services/SoftLayer_Virtual_PlacementGroup_Rule/)

Existing instances cannot be added to a placement group. You can only add a virtual server instance to a placement group at provisioning. To remove an instance from a placement group, you must delete or reclaim the instance.

**Parameters** **client** (*SoftLayer.API.BaseClient*) – the client instance

**create** (*placement\_object*)

Creates a placement group

A `placement_object` is defined as:

```
placement_object = {
    'backendRouterId': 12345,
    'name': 'Test Name',
    'ruleId': 12345
}
```

- [https://softlayer.github.io/reference/datatypes/SoftLayer\\_Virtual\\_PlacementGroup/](https://softlayer.github.io/reference/datatypes/SoftLayer_Virtual_PlacementGroup/)

**Parameters** **placement\_object** (*dictionary*) –

**delete** (*group\_id*)

Deletes a PlacementGroup

Placement group must be empty to be deleted. [https://softlayer.github.io/reference/services/SoftLayer\\_Virtual\\_PlacementGroup/deleteObject](https://softlayer.github.io/reference/services/SoftLayer_Virtual_PlacementGroup/deleteObject)

**get\_all\_rules** ()

Returns all available rules for creating a placement group

**get\_backend\_router\_id\_from\_hostname** (*hostname*)

Finds the backend router Id that matches the hostname given

No way to use an objectFilter to find a backendRouter, so we have to search the hard way.

**get\_object** (*group\_id, mask=None*)

Returns a PlacementGroup Object

[https://softlayer.github.io/reference/services/SoftLayer\\_Virtual\\_PlacementGroup/getObject](https://softlayer.github.io/reference/services/SoftLayer_Virtual_PlacementGroup/getObject)

**get\_routers** ()

Calls SoftLayer\_Virtual\_PlacementGroup::getAvailableRouters()

**get\_rule\_id\_from\_name** (*name*)

Finds the rule that matches name.

SoftLayer\_Virtual\_PlacementGroup\_Rule.getAllObjects doesn't support objectFilters.

**list** (*mask=None*)

List existing placement groups

Calls SoftLayer\_Account::getPlacementGroups

**resolve\_ids** (*identifier*)

Takes a string and tries to resolve to a list of matching ids.

What exactly 'identifier' can be depends on the resolvers

**Parameters** **identifier** (*string*) – identifying string

**Returns** list

If you need more power or functionality than the managers provide, you can make direct API calls as well.

### 3.3 Making API Calls

For full control over your account and services, you can directly call the SoftLayer API. The SoftLayer API client for python leverages SoftLayer's XML-RPC API. It supports authentication, object masks, object filters, limits, offsets, and retrieving objects by id. The following section assumes you have an initialized client named 'client'.

The best way to test our setup is to call the `getObject` method on the `SoftLayer_Account` service.

```
client.call('Account', 'getObject')
```

For a more complex example we'll retrieve a support ticket with id 123456 along with the ticket's updates, the user it's assigned to, the servers attached to it, and the datacenter those servers are in. To retrieve our extra information using an `object mask`.

Retrieve a ticket using object masks.

```
ticket = client.call('Ticket', 'getObject',
                    id=123456, mask="updates, assignedUser, attachedHardware.datacenter")
```

Now add an update to the ticket with `Ticket.addUpdate`. This uses a parameter, which translate to positional arguments in the order that they appear in the API docs.

```
update = client.call('Ticket', 'addUpdate', {'entry' : 'Hello!'}, id=123456)
```

Let's get a listing of virtual guests using the domain example.com

```
client.call('Account', 'getVirtualGuests',
           filter={'virtualGuests': {'domain': {'operation': 'example.com'}}})
```

This call gets tickets created between the beginning of March 1, 2013 and March 15, 2013. More information on [Object Filters](#).

**NOTE** The *value* field for *startDate* and *endDate* is in *[/]*, if you do not put the date in brackets the filter will not work.

```
client.call('Account', 'getTickets',
           filter={
               'tickets': {
                   'createDate': {
                       'operation': 'betweenDate',
                       'options': [
                           {'name': 'startDate', 'value': ['03/01/2013 0:0:0']},
                           {'name': 'endDate', 'value': ['03/15/2013 23:59:59']}
                       ]
                   }
               }
           })
```

SoftLayer's XML-RPC API also allows for pagination.

```
from pprint import pprint

page1 = client.call('Account', 'getVirtualGuests', limit=10, offset=0) # Page 1
page2 = client.call('Account', 'getVirtualGuests', limit=10, offset=10) # Page 2

#Automatic Pagination (v5.5.3+), default limit is 100
result = client.call('Account', 'getVirtualGuests', iter=True, limit=10)
pprint(result)

# Using a python generator, default limit is 100
results = client.iter_call('Account', 'getVirtualGuests', limit=10)
for result in results:
    pprint(result)
```

**NOTE** *client.call(iter=True)* will pull all results, then return. *client.iter\_call()* will return a generator, and only make API calls as you iterate over the results.

Here's how to create a new Cloud Compute Instance using `SoftLayer_Virtual_Guest.createObject`. Be warned, this call actually creates an hourly virtual server so this will have billing implications.

```
client.call('Virtual_Guest', 'createObject', {
    'hostname': 'myhostname',
    'domain': 'example.com',
    'startCpus': 1,
    'maxMemory': 1024,
    'hourlyBillingFlag': 'true',
    'operatingSystemReferenceCode': 'UBUNTU_LATEST',
    'localDiskFlag': 'false'
})
```



## 3.4 Debugging

If you ever need to figure out what exact API call the client is making, you can do the following:

*NOTE* the `print_reproducible` method produces different output for REST and XML-RPC endpoints. If you are using REST, this will produce a CURL call. If you are using XML-RPC, it will produce some pure python code you can use outside of the SoftLayer library.

```
# Setup the client as usual
client = SoftLayer.Client()
# Create an instance of the DebugTransport, which logs API calls
debugger = SoftLayer.DebugTransport(client.transport)
# Set that as the default client transport
client.transport = debugger
# Make your API call
client.call('Account', 'getObject')

# Print out the reproducible call
for call in client.transport.get_last_calls():
    print(client.transport.print_reproducible(call))
```

## 3.5 API Reference

### 3.5.1 SoftLayer Python API Client

SoftLayer API bindings

Usage:

```
>>> import SoftLayer
>>> client = SoftLayer.create_client_from_env(username="username",
                                             api_key="api_key")
>>> resp = client.call('Account', 'getObject')
>>> resp['companyName']
'Your Company'
```

**license** MIT, see LICENSE for more details.

**class** `SoftLayer.BaseClient` (*auth=None, transport=None*)

Base SoftLayer API client.

#### Parameters

- **auth** – auth driver that looks like `SoftLayer.auth.AuthenticationBase`
- **transport** – An object that's callable with this signature: `transport(SoftLayer.transports.Request)`

**authenticate\_with\_password** (*username, password, security\_question\_id=None, security\_question\_answer=None*)

Performs Username/Password Authentication

#### Parameters

- **username** (*string*) – your SoftLayer username
- **password** (*string*) – your SoftLayer password

- **security\_question\_id** (*int*) – The security question id to answer
- **security\_question\_answer** (*string*) – The answer to the security question

**call** (*service, method, \*args, \*\*kwargs*)

Make a SoftLayer API call.

#### Parameters

- **method** – the method to call on the service
- **\*args** – (optional) arguments for the remote call
- **id** – (optional) id for the resource
- **mask** – (optional) object mask
- **filter** (*dict*) – (optional) filter dict
- **headers** (*dict*) – (optional) optional XML-RPC headers
- **compress** (*boolean*) – (optional) Enable/Disable HTTP compression
- **raw\_headers** (*dict*) – (optional) HTTP transport headers
- **limit** (*int*) – (optional) return at most this many results
- **offset** (*int*) – (optional) offset results by this many
- **iter** (*boolean*) – (optional) if True, returns a generator with the results
- **verify** (*bool*) – verify SSL cert
- **cert** – client certificate path

#### Usage:

```
>>> import SoftLayer
>>> client = SoftLayer.create_client_from_env()
>>> client.call('Account', 'getVirtualGuests', mask="id", limit=10)
[...]
```

**iter\_call** (*service, method, \*args, \*\*kwargs*)

A generator that deals with paginating through results.

#### Parameters

- **service** – the name of the SoftLayer API service
- **method** – the method to call on the service
- **limit** (*integer*) – result size for each API call (defaults to 100)
- **\*args** – same optional arguments that `Service.call` takes
- **\*\*kwargs** – same optional keyword arguments that `Service.call` takes

`SoftLayer.create_client_from_env` (*username=None, api\_key=None, endpoint\_url=None, timeout=None, auth=None, config\_file=None, proxy=None, user\_agent=None, transport=None, verify=True*)

Creates a SoftLayer API client using your environment.

Settings are loaded via keyword arguments, environmental variables and config file.

#### Parameters

- **username** – an optional API username if you wish to bypass the package’s built-in username
- **api\_key** – an optional API key if you wish to bypass the package’s built in API key
- **endpoint\_url** – the API endpoint base URL you wish to connect to. Set this to `API_PRIVATE_ENDPOINT` to connect via SoftLayer’s private network.
- **proxy** – proxy to be used to make API calls
- **timeout** (*integer*) – timeout for API requests
- **auth** – an object which responds to `get_headers()` to be inserted into the xml-rpc headers. Example: *BasicAuthentication*
- **config\_file** – A path to a configuration file used to load settings
- **user\_agent** – an optional User Agent to report when making API calls if you wish to bypass the packages built in User Agent string
- **transport** – An object that’s callable with this signature: `transport(SoftLayer.transports.Request)`
- **verify** (*bool*) – decide to verify the server’s SSL/TLS cert. DO NOT SET TO FALSE WITHOUT UNDERSTANDING THE IMPLICATIONS.

Usage:

```
>>> import SoftLayer
>>> client = SoftLayer.create_client_from_env()
>>> resp = client.call('Account', 'getObject')
>>> resp['companyName']
'Your Company'
```

`SoftLayer.Client` (*\*\*kwargs*)

Get a SoftLayer API Client using environmental settings.

Deprecated in favor of `create_client_from_env()`

**class** `SoftLayer.BasicAuthentication` (*username, api\_key*)

Token-based authentication class.

#### Parameters

- **str** (*api\_key*) – a user’s username
- **str** – a user’s API key

**get\_request** (*request*)

Sets token-based auth headers.

**exception** `SoftLayer.SoftLayerError`

The base SoftLayer error.

**exception** `SoftLayer.SoftLayerAPIError` (*fault\_code, fault\_string, \*args*)

SoftLayerAPIError is an exception raised during API errors.

Provides `faultCode` and `faultString` properties.

**class** `SoftLayer.SoftLayerListResult` (*items=None, total\_count=0*)

A SoftLayer API list result.

**total\_count** = `None`

total count of items that exist on the server. This is useful when paginating through a large list of objects.



---

## Command-line Interface

---

The SoftLayer command line interface is available via the *slcli* command available in your *PATH*. The *slcli* command is a reference implementation of SoftLayer API bindings for python and how to efficiently make API calls. See the *Usage Examples* section to see how to discover all of the functionality not fully documented here.

### 4.1 Configuration Setup

To update the configuration, you can use *slcli setup*.

```
$ slcli setup
Username []: username
API Key or Password []:
Endpoint (public|private|custom): public
:.....:
:      Name : Value                                     :
:.....:
:      Username : username                                     :
:      API Key  : oyVmeipYQCNrjVS4rF9bHWV7D75S6palFghF1384v7mwRCbHTfuJ8qRORIqoVnha :
:      Endpoint URL : https://api.softlayer.com/xmlrpc/v3.1/           :
:.....:
Are you sure you want to write settings to "/home/me/.softlayer"? [y/N]: y
```

To check the configuration, you can use *slcli config show*.

```
$ slcli config show
:.....:
:      Name : Value                                     :
:.....:
:      Username : username                                     :
:      API Key  : oyVmeipYQCNrjVS4rF9bHWV7D75S6palFghF1384v7mwRCbHTfuJ8qRORIqoVnha :
:      Endpoint URL : https://api.softlayer.com/xmlrpc/v3.1/           :
:.....:
```

If you are using an account created from the <https://cloud.ibm.com> portal, your username will be literally *apikey*, and use the key provided. [How to create an IBM apikey](#)

To see more about the config file format, see [Configuration File](#).

## 4.2 Usage Examples

To discover the available commands, simply type *slcli*.

```
$ slcli
Usage: slcli [OPTIONS] COMMAND [ARGS]...

SoftLayer Command-line Client

Options:
  --format [table|raw|json|jsonraw] Output format [default: raw]
  -C, --config PATH                 Config file location [default: ~/.softlayer]
  -v, --verbose                      Sets the debug noise level, specify multiple
↳times for more verbosity.
  --proxy TEXT                       HTTP[S] proxy to be use to make API calls
  -y, --really / --not-really        Confirm all prompt actions
  --demo / --no-demo                 Use demo data instead of actually making API
↳calls
  --version                          Show the version and exit.
  -h, --help                          Show this message and exit.

Commands:
  block                Block Storage.
  call-api             Call arbitrary API endpoints.
  cdn                  Content Delivery Network.
  config              CLI configuration.
  dedicatedhost       Dedicated Host.
  dns                 Domain Name System.
  event-log           Event Logs.
  file                File Storage.
  firewall            Firewalls.
  globalip            Global IP addresses.
  hardware            Hardware servers.
  image               Compute images.
  ipsec               IPSEC VPN
  loadbal             Load balancers.
  metadata            Find details about this machine.
  nas                 Network Attached Storage.
  object-storage      Object Storage.
  order               View and order from the catalog.
  report              Reports.
  rwhois              Referral Whois.
  securitygroup       Network security groups.
  setup               Edit configuration.
  shell               Enters a shell for slcli.
  sshkey              SSH Keys.
  ssl                 SSL Certificates.
  subnet              Network subnets.
  summary             Account summary.
  ticket              Support tickets.
  user                Manage Users.
```

(continues on next page)

(continued from previous page)

```
virtual      Virtual Servers.
vlan         Network VLANs.
```

```
To use most commands your SoftLayer username and api_key need to be
configured. The easiest way to do that is to use: 'slcli setup'
```

As you can see, there are a number of commands/sections. To look at the list of subcommands for virtual servers type *slcli vs*. For example:

```
$ slcli vs
Usage: slcli vs [OPTIONS] COMMAND [ARGS]...

Virtual Servers.

Options:
  --help Show this message and exit.

Commands:
  cancel      Cancel virtual servers.
  capture     Capture SoftLayer image.
  create      Order/create virtual servers.
  create-options Virtual server order options.
  credentials List virtual server credentials.
  detail     Get details for a virtual server.
  dns-sync   Sync DNS records.
  edit       Edit a virtual server's details.
  list       List virtual servers.
  network    Manage network settings.
  pause      Pauses an active virtual server.
  power_off  Power off an active virtual server.
  power_on   Power on a virtual server.
  ready      Check if a virtual server is ready.
  reboot     Reboot an active virtual server.
  reload     Reload operating system on a virtual server.
  rescue     Reboot into a rescue image.
  resume     Resumes a paused virtual server.
  upgrade    Upgrade a virtual server.
```

Finally, we can make an actual call. Let's list out the virtual servers on our account by using *slcli vs list*.

Example:

```
$ slcli vs list
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:
↪.....:.....:
:   id   : datacenter :      host          : cores : memory :   primary_ip   : _
↪backend_ip : active_transaction :
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:
↪.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:
: 1234567 :   sjc01   : test.example.com  :    4   :    4G   :    12.34.56    : 65.
↪43.21   :          -          :
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:
↪.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:
```

Most commands will take in additional options/arguments. To see all available actions, use *-help*.

```
$ slcli vs list --help
Usage: slcli vs list [OPTIONS]

List virtual servers.

Options:
  --sortby [guid|hostname|primary_ip|backend_ip|datacenter]
                                Column to sort by
  -c, --cpu INTEGER             Number of CPU cores
  -D, --domain TEXT             Domain portion of the FQDN
  -d, --datacenter TEXT         Datacenter shortname
  -H, --hostname TEXT           Host portion of the FQDN
  -m, --memory INTEGER          Memory in mebibytes
  -n, --network TEXT            Network port speed in Mbps
  --hourly                      Show only hourly instances
  --monthly                      Show only monthly instances
  --tags TEXT                    Show instances that have one of these comma-
                                separated tags
  --help                          Show this message and exit.
```



To see exactly what API call is being made by the SLCLI, you can use the verbose option.

A single `-v` will show a simple version of the API call, along with some statistics

```
slcli -v vs detail 74397127
Calling: SoftLayer_Virtual_Guest::getObject(id=74397127, mask='id,globalIdentifier,
↳fullyQualifiedDomainName,hostname,domain', filter='None', args=(), limit=None,
↳offset=None)
Calling: SoftLayer_Virtual_Guest::getReverseDomainRecords(id=77460683, mask='',
↳filter='None', args=(), limit=None, offset=None)
:.....:
:      name      :                      value                      :
:.....:
: execution_time :                2.020334s                :
:  api_calls    :      SoftLayer_Virtual_Guest::getObject (1.515583s) :
:              :      SoftLayer_Virtual_Guest::getReverseDomainRecords (0.494480s) :
:  version     :                softlayer-python/v5.7.2                :
: python_version :                3.7.3 (default, Mar 27 2019, 09:23:15) :
:              :                [Clang 10.0.1 (clang-1001.0.46.3)]      :
: library_location : /Users/chris/Code/py3/lib/python3.7/site-packages/SoftLayer :
:.....:
:.....:
```

Using `-vv` will print out some API call details in the summary as well.

```
slcli -vv account summary
Calling: SoftLayer_Account::getObject(id=None, mask='mask[ nextInvoiceTotalAmount,
↳pendingInvoice[invoiceTotalAmount], blockDeviceTemplateGroupCount,
↳dedicatedHostCount, domainCount, hardwareCount, networkStorageCount,
↳openTicketCount, networkVlanCount, subnetCount, userCount, virtualGuestCount ]',
↳filter='None', args=(), limit=None, offset=None)
:.....:
:      name      :                      value                      :
:.....:
: execution_time :                0.921271s                :
:  api_calls    :      SoftLayer_Account::getObject (0.911208s)        :
:.....:
:.....:
```

(continues on next page)

(continued from previous page)

```

:   version      :                softlayer-python/v5.7.2                :
: python_version :                3.7.3 (default, Mar 27 2019, 09:23:15)    :
:               :                [Clang 10.0.1 (clang-1001.0.46.3)]        :
: library_location : /Users/chris/Code/py3/lib/python3.7/site-packages/SoftLayer :
:.....:.....:
:.....:.....:
:   :                SoftLayer_Account::getObject                :
:.....:.....:
:   id           :                None                                  :
: mask          :                mask[                                :
:               :                nextInvoiceTotalAmount,                :
:               :                pendingInvoice[invoiceTotalAmount],            :
:               :                blockDeviceTemplateGroupCount,                  :
:               :                dedicatedHostCount,                              :
:               :                domainCount,                                    :
:               :                hardwareCount,                                  :
:               :                networkStorageCount,                            :
:               :                openTicketCount,                                :
:               :                networkVlanCount,                               :
:               :                subnetCount,                                    :
:               :                userCount,                                      :
:               :                virtualGuestCount                               :
: filter        :                None                                  :
: limit         :                None                                  :
: offset        :                None                                  :
:.....:.....:

```

Using `-vvv` will print out the exact API that can be used without the `softlayer-python` framework, A simple python code snippet for XML-RPC, a curl call for REST API calls. This is dependant on the endpoint you are using in the config file.

```

slcli -vvv account summary
curl -u $SSL_USER:$SSL_APIKEY -X GET -H "Accept: */*" -H "Accept-Encoding: gzip,
↪deflate, compress" 'https://api.softlayer.com/rest/v3.1/SoftLayer_Account/
↪getObject.json?objectMask=mask%5B%0A+++++++nextInvoiceTotalAmount%2C
↪%0A+++++++pendingInvoice%5BinvoiceTotalAmount%5D%2C
↪%0A+++++++blockDeviceTemplateGroupCount%2C%0A+++++++dedicatedHostCount%2C
↪%0A+++++++domainCount%2C%0A+++++++hardwareCount%2C
↪%0A+++++++networkStorageCount%2C%0A+++++++openTicketCount%2C
↪%0A+++++++networkVlanCount%2C%0A+++++++subnetCount%2C
↪%0A+++++++userCount%2C%0A+++++++virtualGuestCount%0A+++++++%5D'

```

### 6.1 Account Commands

#### 6.1.1 account summary

Prints some various bits of information about an account

```
account summary [OPTIONS]
```

#### 6.1.2 account events

Summary and acknowledgement of upcoming and ongoing maintenance events

```
account events [OPTIONS]
```

#### Options

##### **--ack-all**

Acknowledge every upcoming event. Doing so will turn off the popup in the control portal

#### 6.1.3 account event-detail

Details of a specific event, and ability to acknowledge event.

```
account event-detail [OPTIONS] IDENTIFIER
```

## Options

### **--ack**

Acknowledge Event. Doing so will turn off the popup in the control portal

## Arguments

### **IDENTIFIER**

Required argument

## 6.1.4 account invoices

List invoices

```
account invoices [OPTIONS]
```

## Options

### **--limit** <limit>

How many invoices to get back. [default: 50]

### **--closed**

Include invoices with a CLOSED status. [default: False]

### **--all**

Return ALL invoices. There may be a lot of these. [default: False]

## 6.1.5 account invoice-detail

Invoice details

```
account invoice-detail [OPTIONS] IDENTIFIER
```

## Options

### **--details**

Shows a very detailed list of charges [default: False]

## Arguments

### **IDENTIFIER**

Required argument

## 6.2 Autoscale Commands

These commands were added in version 5.8.1

For making changes to the triggers or the autoscale group itself, see the [Autoscale Portal](#)

- [Autoscale Product](#)
- [Autoscale Documentation](#)
- [Autoscale Portal](#)

### 6.2.1 autoscale list

List AutoScale Groups.

```
autoscale list [OPTIONS]
```

### 6.2.2 autoscale detail

Get details of an Autoscale groups.

```
autoscale detail [OPTIONS] IDENTIFIER
```

#### Arguments

##### **IDENTIFIER**

Required argument

### 6.2.3 autoscale scale

Scales an Autoscale group. Bypasses a scale group's cooldown period.

```
autoscale scale [OPTIONS] IDENTIFIER
```

#### Options

##### **--up, --down**

'-up' adds guests, '-down' removes guests.

##### **--by, --to**

'-by' will add/remove the specified number of guests. '-to' will add/remove a number of guests to get the group's guest count to the specified number. [required]

##### **--amount** <amount>

Number of guests for the scale action. [required]

#### Arguments

##### **IDENTIFIER**

Required argument

## 6.2.4 autoscale logs

Retrieve logs for an autoscale group

```
autoscale logs [OPTIONS] IDENTIFIER
```

### Options

**-d, --date-min** <date\_min>  
Earliest date to retrieve logs for.

### Arguments

**IDENTIFIER**  
Required argument

## 6.2.5 autoscale tag

Tags all guests in an autoscale group.

**-tags** "Use, quotes, if you, want whitespace"  
**-tags** Otherwise,Just,commas

```
autoscale tag [OPTIONS] IDENTIFIER
```

### Options

**-g, --tags** <tags>  
Tags to set for each guest in this group. Existing tags are overwritten. An empty string will remove all tags

### Arguments

**IDENTIFIER**  
Required argument

## 6.2.6 autoscale edit

Edits an Autoscale group.

```
autoscale edit [OPTIONS] IDENTIFIER
```

### Options

**--name** <name>  
Scale group's name.

**--min** <minimum>  
Set the minimum number of guests

- max** <maximum>  
Set the maximum number of guests
- userdata** <userdata>  
User defined metadata string
- F, --userfile** <userfile>  
Read userdata from a file
- cpu** <cpu>  
Number of CPUs for new guests (existing not effected)
- memory** <memory>  
RAM in MB or GB for new guests (existing not effected)

## Arguments

- IDENTIFIER**  
Required argument

## 6.3 Block Commands

### 6.3.1 block access-authorize

Authorizes hosts to access a given volume

```
block access-authorize [OPTIONS] VOLUME_ID
```

## Options

- h, --hardware-id** <hardware\_id>  
The id of one SoftLayer\_Hardware to authorize
- v, --virtual-id** <virtual\_id>  
The id of one SoftLayer\_Virtual\_Guest to authorize
- i, --ip-address-id** <ip\_address\_id>  
The id of one SoftLayer\_Network\_Subnet\_IPAddress to authorize
- ip-address** <ip\_address>  
An IP address to authorize

## Arguments

- VOLUME\_ID**  
Required argument

### 6.3.2 block access-list

List ACLs.

```
block access-list [OPTIONS] VOLUME_ID
```

### Options

- sortby** <sortby>  
Column to sort by
- columns** <columns>  
Columns to display. Options: id, name, type, private\_ip\_address, source\_subnet, host\_iqn, username, password, allowed\_host\_id

### Arguments

**VOLUME\_ID**  
Required argument

## 6.3.3 block access-revoke

Revokes authorization for hosts accessing a given volume

```
block access-revoke [OPTIONS] VOLUME_ID
```

### Options

- h, --hardware-id** <hardware\_id>  
The id of one SoftLayer\_Hardware to revoke authorization
- v, --virtual-id** <virtual\_id>  
The id of one SoftLayer\_Virtual\_Guest to revoke authorization
- i, --ip-address-id** <ip\_address\_id>  
The id of one SoftLayer\_Network\_Subnet\_IpAddress to revoke authorization
- ip-address** <ip\_address>  
An IP address to revoke authorization

### Arguments

**VOLUME\_ID**  
Required argument

## 6.3.4 block access-password

Changes a password for a volume's access.

access id is the allowed\_host\_id from slcli block access-list

```
block access-password [OPTIONS] ACCESS_ID
```



## Options

**-p, --password** <password>  
Password you want to set, this command will fail if the password is not strong

## Arguments

**ACCESS\_ID**  
Required argument

### 6.3.5 block replica-failback

Failback a block volume from the given replicant volume.

```
block replica-failback [OPTIONS] VOLUME_ID
```

## Arguments

**VOLUME\_ID**  
Required argument

### 6.3.6 block replica-failover

Failover a block volume to the given replicant volume.

```
block replica-failover [OPTIONS] VOLUME_ID
```

## Options

**--replicant-id** <replicant\_id>  
ID of the replicant volume

## Arguments

**VOLUME\_ID**  
Required argument

### 6.3.7 block replica-order

Order a block storage replica volume.

```
block replica-order [OPTIONS] VOLUME_ID
```

## Options

- s, --snapshot-schedule** <snapshot\_schedule>  
Snapshot schedule to use for replication, (INTERVAL | HOURLY | DAILY | WEEKLY) [required]  
**Options** INTERVAL|HOURLY|DAILY|WEEKLY
- l, --location** <location>  
Short name of the data center for the replicant (e.g.: dal09) [required]
- tier** <tier>  
Endurance Storage Tier (IOPS per GB) of the primary volume for which a replicant is ordered [optional]  
**Options** 0.25|2|4|10
- os-type** <os\_type>  
Operating System Type (e.g.: LINUX) of the primary volume for which a replica is ordered [optional]  
**Options** HYPER\_V|LINUX|VMWARE|WINDOWS\_2008|WINDOWS\_GPT|WINDOWSIXEN

## Arguments

- VOLUME\_ID**  
Required argument

### 6.3.8 block replica-partners

List existing replicant volumes for a block volume.

```
block replica-partners [OPTIONS] VOLUME_ID
```

## Options

- sortby** <sortby>  
Column to sort by
- columns** <columns>  
Columns to display. Options: ID, Username, Account ID, Capacity (GB), Hardware ID, Guest ID, Host ID

## Arguments

- VOLUME\_ID**  
Required argument

### 6.3.9 block replica-locations

List suitable replication datacenters for the given volume.

```
block replica-locations [OPTIONS] VOLUME_ID
```

## Options

- sortby** <sortby>  
Column to sort by
- columns** <columns>  
Columns to display. Options: ID, Long Name, Short Name

## Arguments

- VOLUME\_ID**  
Required argument

### 6.3.10 block snapshot-cancel

Cancel existing snapshot space for a given volume.

```
block snapshot-cancel [OPTIONS] VOLUME_ID
```

## Options

- reason** <reason>  
An optional reason for cancellation
- immediate**  
Cancels the snapshot space immediately instead of on the billing anniversary

## Arguments

- VOLUME\_ID**  
Required argument

### 6.3.11 block snapshot-create

Creates a snapshot on a given volume

```
block snapshot-create [OPTIONS] VOLUME_ID
```

## Options

- n, --notes** <notes>  
Notes to set on the new snapshot

## Arguments

- VOLUME\_ID**  
Required argument

### 6.3.12 block snapshot-delete

Deletes a snapshot on a given volume

```
block snapshot-delete [OPTIONS] SNAPSHOT_ID
```

#### Arguments

##### **SNAPSHOT\_ID**

Required argument

### 6.3.13 block snapshot-disable

Disables snapshots on the specified schedule for a given volume

```
block snapshot-disable [OPTIONS] VOLUME_ID
```

#### Options

**--schedule-type** <schedule\_type>  
Snapshot schedule [INTERVAL|HOURLY|DAILY|WEEKLY] [required]

#### Arguments

##### **VOLUME\_ID**

Required argument

### 6.3.14 block snapshot-enable

Enables snapshots for a given volume on the specified schedule

```
block snapshot-enable [OPTIONS] VOLUME_ID
```

#### Options

**--schedule-type** <schedule\_type>  
Snapshot schedule [INTERVAL|HOURLY|DAILY|WEEKLY] [required]

**--retention-count** <retention\_count>  
Number of snapshots to retain [required]

**--minute** <minute>  
Minute of the day when snapshots should be taken

**--hour** <hour>  
Hour of the day when snapshots should be taken

**--day-of-week** <day\_of\_week>  
Day of the week when snapshots should be taken

## Arguments

**VOLUME\_ID**

Required argument

### 6.3.15 block snapshot-schedule-list

Lists snapshot schedules for a given volume

```
block snapshot-schedule-list [OPTIONS] VOLUME_ID
```

## Arguments

**VOLUME\_ID**

Required argument

### 6.3.16 block snapshot-list

List block storage snapshots.

```
block snapshot-list [OPTIONS] VOLUME_ID
```

## Options

**--sortby** <sortby>

Column to sort by

**--columns** <columns>

Columns to display. Options: id, name, created, size\_bytes

## Arguments

**VOLUME\_ID**

Required argument

### 6.3.17 block snapshot-order

Order snapshot space for a block storage volume.

```
block snapshot-order [OPTIONS] VOLUME_ID
```

## Options

**--capacity** <capacity>

Size of snapshot space to create in GB [required]

**--tier** <tier>

Endurance Storage Tier (IOPS per GB) of the block volume for which space is ordered [optional, and only valid for endurance storage volumes]

**Options** 0.25|2|4|10

**--upgrade**

Flag to indicate that the order is an upgrade

## Arguments

**VOLUME\_ID**

Required argument

### 6.3.18 block snapshot-restore

Restore block volume using a given snapshot

```
block snapshot-restore [OPTIONS] VOLUME_ID
```

## Options

**-s, --snapshot-id** <snapshot\_id>

The id of the snapshot which will be used to restore the block volume

## Arguments

**VOLUME\_ID**

Required argument

### 6.3.19 block volume-cancel

Cancel an existing block storage volume.

```
block volume-cancel [OPTIONS] VOLUME_ID
```

## Options

**--reason** <reason>

An optional reason for cancellation

**--immediate**

Cancels the block storage volume immediately instead of on the billing anniversary

## Arguments

**VOLUME\_ID**

Required argument

### 6.3.20 block volume-count

List number of block storage volumes per datacenter.

```
block volume-count [OPTIONS]
```

#### Options

- d, --datacenter** <datacenter>  
Datacenter shortname
- sortby** <sortby>  
Column to sort by

### 6.3.21 block volume-detail

Display details for a specified volume.

```
block volume-detail [OPTIONS] VOLUME_ID
```

#### Arguments

- VOLUME\_ID**  
Required argument

### 6.3.22 block volume-duplicate

Order a duplicate block storage volume.

```
block volume-duplicate [OPTIONS] ORIGIN_VOLUME_ID
```

#### Options

- o, --origin-snapshot-id** <origin\_snapshot\_id>  
ID of an origin volume snapshot to use for duplication.
- c, --duplicate-size** <duplicate\_size>  
Size of duplicate block volume in GB. **\*If no size is specified, the size of the origin volume will be used.\*** Potential Sizes: [20, 40, 80, 100, 250, 500, 1000, 2000, 4000, 8000, 12000] Minimum: [the size of the origin volume]
- i, --duplicate-iops** <duplicate\_iops>  
Performance Storage IOPS, between 100 and 6000 in multiples of 100 [only used for performance volumes] **\*If no IOPS value is specified, the IOPS value of the origin volume will be used.\*** Requirements: [If IOPS/GB for the origin volume is less than 0.3, IOPS/GB for the duplicate must also be less than 0.3. If IOPS/GB for the origin volume is greater than or equal to 0.3, IOPS/GB for the duplicate must also be greater than or equal to 0.3.]
- t, --duplicate-tier** <duplicate\_tier>  
Endurance Storage Tier (IOPS per GB) [only used for endurance volumes] **\*If no tier is specified, the tier of the origin volume will be used.\*** Requirements: [If IOPS/GB for the origin volume is 0.25, IOPS/GB for the

duplicate must also be 0.25. If IOPS/GB for the origin volume is greater than 0.25, IOPS/GB for the duplicate must also be greater than 0.25.]

**Options** 0.25|2|4|10

**-s, --duplicate-snapshot-size** <duplicate\_snapshot\_size>

The size of snapshot space to order for the duplicate. **\*If no snapshot space size is specified, the snapshot space size of the origin block volume will be used.\*** Input “0” for this parameter to order a duplicate volume with no snapshot space.

**--billing** <billing>

Optional parameter for Billing rate (default to monthly)

**Options** hourly|monthly

## Arguments

**ORIGIN\_VOLUME\_ID**

Required argument

### 6.3.23 block volume-list

List block storage.

```
block volume-list [OPTIONS]
```

#### Options

**-u, --username** <username>

Volume username

**-d, --datacenter** <datacenter>

Datacenter shortname

**--storage-type** <storage\_type>

Type of storage volume

**Options** performance|endurance

**--sortby** <sortby>

Column to sort by

**--columns** <columns>

Columns to display. Options: id, username, datacenter, storage\_type, capacity\_gb, bytes\_used, iops, ip\_addr, lunId, active\_transactions, rep\_partner\_count, created\_by

### 6.3.24 block volume-modify

Modify an existing block storage volume.

```
block volume-modify [OPTIONS] VOLUME_ID
```



## Options

- c, --new-size** <new\_size>  
New Size of block volume in GB. **\*If no size is given, the original size of volume is used.\*** Potential Sizes: [20, 40, 80, 100, 250, 500, 1000, 2000, 4000, 8000, 12000] Minimum: [the original size of the volume]
- i, --new-iops** <new\_iops>  
Performance Storage IOPS, between 100 and 6000 in multiples of 100 [only for performance volumes] **\*If no IOPS value is specified, the original IOPS value of the volume will be used.\*** Requirements: [If original IOPS/GB for the volume is less than 0.3, new IOPS/GB must also be less than 0.3. If original IOPS/GB for the volume is greater than or equal to 0.3, new IOPS/GB for the volume must also be greater than or equal to 0.3.]
- t, --new-tier** <new\_tier>  
Endurance Storage Tier (IOPS per GB) [only for endurance volumes] **\*If no tier is specified, the original tier of the volume will be used.\*** Requirements: [If original IOPS/GB for the volume is 0.25, new IOPS/GB for the volume must also be 0.25. If original IOPS/GB for the volume is greater than 0.25, new IOPS/GB for the volume must also be greater than 0.25.]

**Options** 0.25|2|4|10

## Arguments

- VOLUME\_ID**  
Required argument

### 6.3.25 block volume-order

Order a block storage volume.

Valid size and iops options can be found here: <https://console.bluemix.net/docs/infrastructure/BlockStorage/index.html#provisioning>

```
block volume-order [OPTIONS]
```

## Options

- storage-type** <storage\_type>  
Type of block storage volume [required]  
**Options** performancelendurance
- size** <size>  
Size of block storage volume in GB. [required]
- iops** <iops>  
Performance Storage IOPS. Options vary based on storage size. [required for storage-type performance]
- tier** <tier>  
Endurance Storage Tier (IOP per GB) [required for storage-type endurance]  
**Options** 0.25|2|4|10
- os-type** <os\_type>  
Operating System [required]  
**Options** HYPER\_V|LINUX|VMWARE|WINDOWS\_2008|WINDOWS\_GPT|WINDOWS|XEN

- location** <location>  
Datacenter short name (e.g.: dal09) [required]
- snapshot-size** <snapshot\_size>  
Optional parameter for ordering snapshot space along with endurance block storage; specifies the size (in GB) of snapshot space to order
- service-offering** <service\_offering>  
The service offering package to use for placing the order. [optional, default is 'storage\_as\_a\_service']. enterprise and performance are deprecated  
**Options** storage\_as\_a\_service|enterprise|performance
- billing** <billing>  
Optional parameter for Billing rate (default to monthly)  
**Options** hourly|monthly

### 6.3.26 block volume-set-lun-id

Set the LUN ID on an existing block storage volume.

The LUN ID only takes effect during the Host Authorization process. It is recommended (but not necessary) to de-authorize all hosts before using this method. See *block access-revoke*.

VOLUME\_ID - the volume ID on which to set the LUN ID.

LUN\_ID - recommended range is an integer between 0 and 255. Advanced users can use an integer between 0 and 4095.

```
block volume-set-lun-id [OPTIONS] VOLUME_ID LUN_ID
```

#### Arguments

**VOLUME\_ID**  
Required argument

**LUN\_ID**  
Required argument

### 6.3.27 block volume-limits

List number of block storage volumes limit per datacenter.

```
block volume-limits [OPTIONS]
```

#### Options

**--sortby** <sortby>  
Column to sort by

### 6.3.28 block subnets-list

List block storage assigned subnets for the given host id.

access\_id is the host\_id obtained by: `slcli block access-list <volume_id>`

```
block subnets-list [OPTIONS] ACCESS_ID
```

#### Arguments

##### ACCESS\_ID

Required argument

### 6.3.29 block subnets-assign

Assign block storage subnets to the given host id.

access\_id is the host\_id obtained by: `slcli block access-list <volume_id>`

SoftLayer\_Account::iscsiisolationdisabled must be False to use this command

```
block subnets-assign [OPTIONS] ACCESS_ID
```

#### Options

**--subnet-id** <subnet\_id>

ID of the subnets to assign; e.g.: `--subnet-id 1234`

#### Arguments

##### ACCESS\_ID

Required argument

### 6.3.30 block subnets-remove

Remove block storage subnets for the given host id.

access\_id is the host\_id obtained by: `slcli block access-list <volume_id>`

SoftLayer\_Account::iscsiisolationdisabled must be False to use this command

```
block subnets-remove [OPTIONS] ACCESS_ID
```

#### Options

**--subnet-id** <subnet\_id>

ID of the subnets to remove; e.g.: `--subnet-id 1234`

## Arguments

### **ACCESS\_ID**

Required argument

## 6.4 Interacting with CDN

### 6.4.1 cdn detail

Detail a CDN Account.

```
cdn detail [OPTIONS] UNIQUE_ID
```

#### Options

**--history** <history>

Bandwidth, Hits, Ratio counted over history number of days ago. 89 is the maximum.

#### Arguments

### **UNIQUE\_ID**

Required argument

### 6.4.2 cdn list

List all CDN accounts.

```
cdn list [OPTIONS]
```

#### Options

**--sortby** <sortby>

Column to sort by

**Options** unique\_id|domain|origin|vendor|cname|status

### 6.4.3 cdn origin-add

Create an origin path for an existing CDN mapping.

For more information see the following documentation:

<https://cloud.ibm.com/docs/infrastructure/CDN?topic=CDN-manage-your-cdn#adding-origin-path-details>

```
cdn origin-add [OPTIONS] UNIQUE_ID ORIGIN PATH
```

## Options

- t, --origin-type** <origin\_type>  
The origin type. [default: server]  
**Options** server|storage
- H, --header** <header>  
The host header to communicate with the origin.
- b, --bucket-name** <bucket\_name>  
The name of the available resource [required if --origin-type=storage]
- p, --port** <port>  
The http port number. [default: 80]
- P, --protocol** <protocol>  
The protocol used by the origin. [default: http]
- o, --optimize-for** <optimize\_for>  
Performance configuration [default: web]  
**Options** web|videofile
- e, --extensions** <extensions>  
File extensions that can be stored in the CDN, example: 'jpg, png, pdf'
- c, --cache-query** <cache\_query>  
Cache query rules with the following formats: 'ignore-all', 'include: <query-names>', 'ignore: <query-names>' [default: include-all]

## Arguments

- UNIQUE\_ID**  
Required argument
- ORIGIN**  
Required argument
- PATH**  
Required argument

### 6.4.4 cdn origin-list

List origin path for an existing CDN mapping.

```
cdn origin-list [OPTIONS] UNIQUE_ID
```

## Arguments

- UNIQUE\_ID**  
Required argument

## 6.4.5 cdn origin-remove

Removes an origin path for an existing CDN mapping.

```
cdn origin-remove [OPTIONS] UNIQUE_ID ORIGIN_PATH
```

### Arguments

**UNIQUE\_ID**

Required argument

**ORIGIN\_PATH**

Required argument

## 6.4.6 cdn purge

Creates a purge record and also initiates the purge call.

**Example:** `slcli cdn purge 9779455 /article/file.txt`

For more information see the following documentation:

<https://cloud.ibm.com/docs/infrastructure/CDN?topic=CDN-manage-your-cdn#purging-cached-content>

```
cdn purge [OPTIONS] UNIQUE_ID PATH
```

### Arguments

**UNIQUE\_ID**

Required argument

**PATH**

Required argument

## 6.5 Call API

### 6.5.1 call-api

Call arbitrary API endpoints with the given SERVICE and METHOD.

Example:

```
slcli call-api Account getObject
slcli call-api Account getVirtualGuests --limit=10 --mask=id,hostname
slcli call-api Virtual_Guest getObject --id=12345
slcli call-api Metric_Tracking_Object getBandwidthData --id=1234 \
    "2015-01-01 00:00:00" "2015-01-1 12:00:00" public
slcli call-api Account getVirtualGuests \
    -f 'virtualGuests.datacenter.name=dal05' \
    -f 'virtualGuests.maxCpu=4' \
    --mask=id,hostname,datacenter.name,maxCpu
slcli call-api Account getVirtualGuests \
    -f 'virtualGuests.datacenter.name IN dal05,sng01'
```

```
call-api [OPTIONS] SERVICE METHOD [PARAMETERS]...
```

## Options

- id** <\_id>  
Init parameter
- f, --filter** <\_filters>  
Object filters. This should be of the form: 'property=value' or 'nested.property=value'. Complex filters like betweenDate are not currently supported. (multiple occurrence permitted)
- mask** <mask>  
String-based object mask
- limit** <limit>  
Result limit
- offset** <offset>  
Result offset
- output-python, --no-output-python**  
Show python example code instead of executing the call

## Arguments

- SERVICE**  
Required argument
- METHOD**  
Required argument
- PARAMETERS**  
Optional argument(s)

## 6.6 Shell

### 6.7 MetaData

Used to retrieve information about the server making the API call. Can be called with an un-authenticated API call.

#### 6.7.1 metadata

Find details about the machine making these API calls.

Table 1: Choices

backend_ip
backend_mac
datacenter
datacenter_id
fqdn
frontend_mac
id
ip
network
provision_state
tags
user_data

```
metadata [OPTIONS] [backend_ip|backend_mac|datacenter|datacenter_id|fqdn|front
end_mac|id|ip|network|provision_state|tags|user_data]
```

## Arguments

### PROP

Required argument

## 6.8 Config

Creating an IBMID apikey IBMID for services

Creating a SoftLayer apikey

### 6.8.1 config setup

Setup the ~/.softlayer file with username and apikey.

Set the username to ‘apikey’ for cloud.ibm.com accounts.

```
config setup [OPTIONS]
```

### 6.8.2 config show

Show current configuration.

```
config show [OPTIONS]
```

### 6.8.3 setup

Setup the ~/.softlayer file with username and apikey.

Set the username to ‘apikey’ for cloud.ibm.com accounts.



```
setup [OPTIONS]
```

## 6.9 Dedicated Host Commands

### 6.9.1 dedicatedhost list

List dedicated host.

```
dedicatedhost list [OPTIONS]
```

#### Options

- c, --cpu** <cpu>  
Number of CPU cores
- tag** <tag>  
Filter by tags (multiple occurrence permitted)
- sortby** <sortby>  
Column to sort by [default: name]
- columns** <columns>  
Columns to display. [options: datacenter, created\_by, tags] [default: id,name,cpuCount,diskCapacity,memoryCapacity,datacenter,guestCount]
- d, --datacenter** <datacenter>  
Datacenter shortname
- H, --name** <name>  
Host portion of the FQDN
- m, --memory** <memory>  
Memory capacity in mebibytes
- D, --disk** <disk>  
Disk capacity

### 6.9.2 dedicatedhost create

Order/create a dedicated host.

```
dedicatedhost create [OPTIONS]
```

#### Options

- H, --hostname** <hostname>  
Host portion of the FQDN [required]
- r, --router** <router>  
Router hostname ex. fcr02a.dal13
- D, --domain** <domain>  
Domain portion of the FQDN [required]

- d, --datacenter** <datacenter>  
Datacenter shortname [required]
- f, --flavor** <flavor>  
Dedicated Virtual Host flavor [required]
- billing** <billing>  
Billing rate [default: hourly]  
**Options** hourly|monthly
- verify**  
Verify dedicatedhost without creating it.
- t, --template** <template>  
A template file that defaults the command-line options
- export** <export>  
Exports options to a template file

### 6.9.3 dedicatedhost create-options

host order options for a given dedicated host.

To get a list of available backend routers see example: `slcli dh create-options --datacenter dal05 --flavor 56_CORES_X_242_RAM_X_1_4_TB`

```
dedicatedhost create-options [OPTIONS]
```

#### Options

- d, --datacenter** <datacenter>  
Router hostname (requires --flavor) ex. ams01
- f, --flavor** <flavor>  
Dedicated Virtual Host flavor (requires --datacenter) ex. 56\_CORES\_X\_242\_RAM\_X\_1\_4\_TB

### 6.9.4 dedicatedhost detail

Get details for a virtual server.

```
dedicatedhost detail [OPTIONS] IDENTIFIER
```

#### Options

- price**  
Show associated prices
- guests**  
Show guests on dedicated host

## Arguments

### IDENTIFIER

Required argument

## 6.9.5 dedicatedhost cancel

Cancel a dedicated host server immediately

```
dedicatedhost cancel [OPTIONS] IDENTIFIER
```

## Arguments

### IDENTIFIER

Required argument

## 6.9.6 dedicatedhost cancel-guests

Cancel all virtual guests of the dedicated host immediately.

Use the 'slcli vs cancel' command to cancel an specific guest

```
dedicatedhost cancel-guests [OPTIONS] IDENTIFIER
```

## Arguments

### IDENTIFIER

Required argument

## 6.9.7 dedicatedhost list-guests

List guests which are in a dedicated host server.

```
dedicatedhost list-guests [OPTIONS] IDENTIFIER
```

## Options

- c, --cpu** <cpu>  
Number of CPU cores
- D, --domain** <domain>  
Domain portion of the FQDN
- H, --hostname** <hostname>  
Host portion of the FQDN
- m, --memory** <memory>  
Memory in mebibytes
- tag** <tag>  
Filter by tags (multiple occurrence permitted)

**--sortby** <sortby>  
Column to sort by [default: hostname]

**--columns** <columns>  
Columns to display. [options: guid, cpu, memory, datacenter, primary\_ip, backend\_ip, created\_by, power\_state, tags] [default: id,hostname,domain,primary\_ip,backend\_ip,power\_state]

## Arguments

**IDENTIFIER**  
Required argument

## 6.10 DNS Management

### 6.10.1 dns import

Import zone based off a BIND zone file.

```
dns import [OPTIONS] ZONEFILE
```

## Options

**--dry-run**  
Don't actually create records

## Arguments

**ZONEFILE**  
Required argument

### 6.10.2 dns record-add

Add resource record.

Each resource record contains a RECORD and DATA property, defining a resource's name and it's target data. Domains contain multiple types of resource records so it can take one of the following values: A, AAAA, CNAME, MX, SPF, SRV, and PTR.

About reverse records (PTR), the RECORD value must to be the public Ip Address of device you would like to manage reverse DNS.

```
slcli dns record-add 10.10.8.21 PTR myhost.com --ttl=900
```

Examples:

```
slcli dns record-add myhost.com A 192.168.1.10 --zone=foobar.com --ttl=900
```

```
slcli dns record-add myhost.com AAAA 2001:DB8::1 --zone=foobar.com
```

```
slcli dns record-add 192.168.1.2 MX 192.168.1.10 --zone=foobar.com --priority=11 --ttl=1800
```

```
slcli dns record-add myhost.com TXT "txt-verification=rXOxyZounZs87oacJSKvbUSIQ"  
--zone=2223334
```

```
slcli dns record-add myhost.com SPF "v=spf1 include:_spf.google.com ~all" --zone=2223334
slcli dns record-add myhost.com SRV 192.168.1.10 --zone=2223334 --service=foobar --port=80 --protocol=TCP
```

```
dns record-add [OPTIONS] RECORD RECORD_TYPE DATA
```

## Options

- zone** <zone>  
Zone name or identifier that the resource record will be associated with. Required for all record types except PTR
- ttl** <ttl>  
TTL value in seconds, such as 86400 [default: 900]
- priority** <priority>  
The priority of the target host. (MX or SRV type only) [default: 10]
- protocol** <protocol>  
The protocol of the service, usually either TCP or UDP. (SRV type only) [default: tcp]
- Options** tcpludptls
- port** <port>  
The TCP/UDP/TLS port on which the service is to be found. (SRV type only)
- service** <service>  
The symbolic name of the desired service. (SRV type only)
- weight** <weight>  
Relative weight for records with same priority. (SRV type only) [default: 5]

## Arguments

- RECORD**  
Required argument
- RECORD\_TYPE**  
Required argument
- DATA**  
Required argument

### 6.10.3 dns record-edit

Update DNS record.

```
dns record-edit [OPTIONS] ZONE_ID
```

## Options

- by-record** <by\_record>  
Edit by host record, such as www

- by-id** <by\_id>  
Edit a single record by its ID
- data** <data>  
Record data, such as an IP address
- ttl** <ttl>  
TTL value in seconds, such as 86400

## Arguments

- ZONE\_ID**  
Required argument

### 6.10.4 dns record-list

List all records in a zone.

```
dns record-list [OPTIONS] ZONE
```

## Options

- data** <data>  
Record data, such as an IP address
- record** <record>  
Host record, such as www
- ttl** <ttl>  
TTL value in seconds, such as 86400
- type** <type>  
Record type, such as A or CNAME

## Arguments

- ZONE**  
Required argument

### 6.10.5 dns record-remove

Remove resource record.

```
dns record-remove [OPTIONS] RECORD_ID
```

## Arguments

- RECORD\_ID**  
Required argument

## 6.10.6 dns zone-create

Create a zone.

```
dns zone-create [OPTIONS] ZONE
```

### Arguments

**ZONE**

Required argument

## 6.10.7 dns zone-delete

Delete zone.

```
dns zone-delete [OPTIONS] ZONE
```

### Arguments

**ZONE**

Required argument

## 6.10.8 dns zone-list

List all zones.

```
dns zone-list [OPTIONS]
```

## 6.10.9 dns zone-print

Print zone in BIND format.

```
dns zone-print [OPTIONS] ZONE
```

### Arguments

**ZONE**

Required argument

## 6.11 Event-Log Commands

### 6.11.1 event-log get

Get Event Logs

**Example:** `slcli event-log get -d 01/01/2019 -D 02/01/2019 -t User -l 10`

```
event-log get [OPTIONS]
```

## Options

- d, --date-min** <date\_min>  
The earliest date we want to search for event logs in mm/dd/yyyy format.
- D, --date-max** <date\_max>  
The latest date we want to search for event logs in mm/dd/yyyy format.
- e, --obj-event** <obj\_event>  
The event we want to get event logs for
- i, --obj-id** <obj\_id>  
The id of the object we want to get event logs for
- t, --obj-type** <obj\_type>  
The type of the object we want to get event logs for
- z, --utc-offset** <utc\_offset>  
UTC Offset for searching with dates. +/-HHMM format [default: -0000]
- metadata, --no-metadata**  
Display metadata if present [default: False]
- l, --limit** <limit>  
Total number of result to return. -1 to return ALL, there may be a LOT of these. [default: 50]

There are usually quite a few events on an account, so be careful when using the `-limit -1` option. The command will automatically break requests out into smaller sub-requests, but this command may take a very long time to complete. It will however print out data as it comes in.

## 6.11.2 event-log types

### Get Event Log Types

```
event-log types [OPTIONS]
```

Currently the types are as follows, more may be added in the future.

```

:.....:
:      types      :
:.....:
:      Account    :
:      CDN        :
:      User       :
: Bare Metal Instance :
: API Authentication :
:      Server     :
:      CCI        :
:      Image      :
:      Bluemix LB  :
:      Facility    :
: Cloud Object Storage :
:      Security Group :
:.....:

```



## 6.12 File Commands

### 6.12.1 file access-authorize

Authorizes hosts to access a given volume

```
file access-authorize [OPTIONS] VOLUME_ID
```

#### Options

- h, --hardware-id** <hardware\_id>  
The id of one SoftLayer\_Hardware to authorize
- v, --virtual-id** <virtual\_id>  
The id of one SoftLayer\_Virtual\_Guest to authorize
- i, --ip-address-id** <ip\_address\_id>  
The id of one SoftLayer\_Network\_Subnet\_IPAddress to authorize
- ip-address** <ip\_address>  
An IP address to authorize
- s, --subnet-id** <subnet\_id>  
The id of one SoftLayer\_Network\_Subnet to authorize

#### Arguments

**VOLUME\_ID**  
Required argument

### 6.12.2 file access-list

List ACLs.

```
file access-list [OPTIONS] VOLUME_ID
```

#### Options

- sortby** <sortby>  
Column to sort by
- columns** <columns>  
Columns to display. Options: id, name, type, private\_ip\_address, source\_subnet, host\_iqn, username, password, allowed\_host\_id

#### Arguments

**VOLUME\_ID**  
Required argument

### 6.12.3 file access-revoke

Revokes authorization for hosts accessing a given volume

```
file access-revoke [OPTIONS] VOLUME_ID
```

#### Options

- h, --hardware-id** <hardware\_id>  
The id of one SoftLayer\_Hardware to revoke authorization
- v, --virtual-id** <virtual\_id>  
The id of one SoftLayer\_Virtual\_Guest to revoke authorization
- i, --ip-address-id** <ip\_address\_id>  
The id of one SoftLayer\_Network\_Subnet\_IpAddress to revoke authorization
- ip-address** <ip\_address>  
An IP address to revoke authorization
- s, --subnet-id** <subnet\_id>  
The id of one SoftLayer\_Network\_Subnet to revoke authorization

#### Arguments

**VOLUME\_ID**  
Required argument

### 6.12.4 file replica-failback

Failback a file volume from the given replicant volume.

```
file replica-failback [OPTIONS] VOLUME_ID
```

#### Arguments

**VOLUME\_ID**  
Required argument

### 6.12.5 file replica-failover

Failover a file volume to the given replicant volume.

```
file replica-failover [OPTIONS] VOLUME_ID
```

#### Options

- replicant-id** <replicant\_id>  
ID of the replicant volume

## Arguments

**VOLUME\_ID**

Required argument

### 6.12.6 file replica-order

Order a file storage replica volume.

```
file replica-order [OPTIONS] VOLUME_ID
```

## Options

**-s, --snapshot-schedule** <snapshot\_schedule>

Snapshot schedule to use for replication, (INTERVAL | HOURLY | DAILY | WEEKLY) [required]

**Options** INTERVAL|HOURLY|DAILY|WEEKLY**-l, --location** <location>

Short name of the data center for the replicant (e.g.: dal09) [required]

**--tier** <tier>

Endurance Storage Tier (IOPS per GB) of the primary volume for which a replicant is ordered [optional]

**Options** 0.25|2|4|10

## Arguments

**VOLUME\_ID**

Required argument

### 6.12.7 file replica-partners

List existing replicant volumes for a file volume.

```
file replica-partners [OPTIONS] VOLUME_ID
```

## Options

**--sortby** <sortby>

Column to sort by

**--columns** <columns>

Columns to display. Options: ID, Username, Account ID, Capacity (GB), Hardware ID, Guest ID, Host ID

## Arguments

**VOLUME\_ID**

Required argument

## 6.12.8 file replica-locations

List suitable replication datacenters for the given volume.

```
file replica-locations [OPTIONS] VOLUME_ID
```

### Options

- sortby** <sortby>  
Column to sort by
- columns** <columns>  
Columns to display. Options: ID, Long Name, Short Name

### Arguments

**VOLUME\_ID**  
Required argument

## 6.12.9 file snapshot-cancel

Cancel existing snapshot space for a given volume.

```
file snapshot-cancel [OPTIONS] VOLUME_ID
```

### Options

- reason** <reason>  
An optional reason for cancellation
- immediate**  
Cancels the snapshot space immediately instead of on the billing anniversary

### Arguments

**VOLUME\_ID**  
Required argument

## 6.12.10 file snapshot-create

Creates a snapshot on a given volume

```
file snapshot-create [OPTIONS] VOLUME_ID
```

### Options

- n, --notes** <notes>  
Notes to set on the new snapshot

## Arguments

### **VOLUME\_ID**

Required argument

## 6.12.11 file snapshot-delete

Deletes a snapshot on a given volume

```
file snapshot-delete [OPTIONS] SNAPSHOT_ID
```

## Arguments

### **SNAPSHOT\_ID**

Required argument

## 6.12.12 file snapshot-disable

Disables snapshots on the specified schedule for a given volume

```
file snapshot-disable [OPTIONS] VOLUME_ID
```

## Options

**--schedule-type** <schedule\_type>  
Snapshot schedule [INTERVAL|HOURLY|DAILY|WEEKLY] [required]

## Arguments

### **VOLUME\_ID**

Required argument

## 6.12.13 file snapshot-enable

Enables snapshots for a given volume on the specified schedule

```
file snapshot-enable [OPTIONS] VOLUME_ID
```

## Options

**--schedule-type** <schedule\_type>  
Snapshot schedule [INTERVAL|HOURLY|DAILY|WEEKLY] [required]

**--retention-count** <retention\_count>  
Number of snapshots to retain [required]

**--minute** <minute>  
Minute of the day when snapshots should be taken

- hour** <hour>  
Hour of the day when snapshots should be taken
- day-of-week** <day\_of\_week>  
Day of the week when snapshots should be taken

## Arguments

- VOLUME\_ID**  
Required argument

### 6.12.14 file snapshot-list

List file storage snapshots.

```
file snapshot-list [OPTIONS] VOLUME_ID
```

## Options

- sortby** <sortby>  
Column to sort by
- columns** <columns>  
Columns to display. Options: id, name, created, size\_bytes

## Arguments

- VOLUME\_ID**  
Required argument

### 6.12.15 file snapshot-order

Order snapshot space for a file storage volume.

```
file snapshot-order [OPTIONS] VOLUME_ID
```

## Options

- capacity** <capacity>  
Size of snapshot space to create in GB [required]
- tier** <tier>  
Endurance Storage Tier (IOPS per GB) of the file volume for which space is ordered [optional, and only valid for endurance storage volumes]  
**Options** 0.25|2|4|10
- upgrade**  
Flag to indicate that the order is an upgrade

## Arguments

### **VOLUME\_ID**

Required argument

## 6.12.16 file snapshot-restore

Restore file volume using a given snapshot

```
file snapshot-restore [OPTIONS] VOLUME_ID
```

## Options

**-s, --snapshot-id** <snapshot\_id>

The id of the snapshot which will be used to restore the block volume

## Arguments

### **VOLUME\_ID**

Required argument

## 6.12.17 file volume-cancel

Cancel an existing file storage volume.

```
file volume-cancel [OPTIONS] VOLUME_ID
```

## Options

**--reason** <reason>

An optional reason for cancellation

**--immediate**

Cancels the file storage volume immediately instead of on the billing anniversary

## Arguments

### **VOLUME\_ID**

Required argument

## 6.12.18 file volume-count

List number of file storage volumes per datacenter.

```
file volume-count [OPTIONS]
```

## Options

- d, --datacenter** <datacenter>  
Datacenter shortname
- sortby** <sortby>  
Column to sort by

### 6.12.19 file volume-detail

Display details for a specified volume.

```
file volume-detail [OPTIONS] VOLUME_ID
```

## Arguments

- VOLUME\_ID**  
Required argument

### 6.12.20 file volume-duplicate

Order a duplicate file storage volume.

```
file volume-duplicate [OPTIONS] ORIGIN_VOLUME_ID
```

## Options

- o, --origin-snapshot-id** <origin\_snapshot\_id>  
ID of an origin volume snapshot to use for duplication.
- c, --duplicate-size** <duplicate\_size>  
Size of duplicate file volume in GB. **\*If no size is specified, the size of the origin volume will be used.\***  
Minimum: [the size of the origin volume]
- i, --duplicate-iops** <duplicate\_iops>  
Performance Storage IOPS, between 100 and 6000 in multiples of 100 [only used for performance volumes] **\*If no IOPS value is specified, the IOPS value of the origin volume will be used.\*** Requirements: [If IOPS/GB for the origin volume is less than 0.3, IOPS/GB for the duplicate must also be less than 0.3. If IOPS/GB for the origin volume is greater than or equal to 0.3, IOPS/GB for the duplicate must also be greater than or equal to 0.3.]
- t, --duplicate-tier** <duplicate\_tier>  
Endurance Storage Tier (IOPS per GB) [only used for endurance volumes] **\*If no tier is specified, the tier of the origin volume will be used.\*** Requirements: [If IOPS/GB for the origin volume is 0.25, IOPS/GB for the duplicate must also be 0.25. If IOPS/GB for the origin volume is greater than 0.25, IOPS/GB for the duplicate must also be greater than 0.25.]  
**Options** 0.25|2|4|10
- s, --duplicate-snapshot-size** <duplicate\_snapshot\_size>  
The size of snapshot space to order for the duplicate. **\*If no snapshot space size is specified, the snapshot space size of the origin file volume will be used.\*** Input "0" for this parameter to order a duplicate volume with no snapshot space.



**--billing** <billing>  
Optional parameter for Billing rate (default to monthly)  
**Options** hourly|monthly

## Arguments

**ORIGIN\_VOLUME\_ID**  
Required argument

### 6.12.21 file volume-list

List file storage.

```
file volume-list [OPTIONS]
```

## Options

**-u, --username** <username>  
Volume username

**-d, --datacenter** <datacenter>  
Datacenter shortname

**--storage-type** <storage\_type>  
Type of storage volume  
**Options** performance|endurance

**--sortby** <sortby>  
Column to sort by

**--columns** <columns>  
Columns to display. Options: id, username, datacenter, storage\_type, capacity\_gb, bytes\_used, ip\_addr, active\_transactions, mount\_addr, rep\_partner\_count, created\_by

### 6.12.22 file volume-modify

Modify an existing file storage volume.

```
file volume-modify [OPTIONS] VOLUME_ID
```

## Options

**-c, --new-size** <new\_size>  
New Size of file volume in GB. **\*If no size is given, the original size of volume is used.\*** Potential Sizes: [20, 40, 80, 100, 250, 500, 1000, 2000, 4000, 8000, 12000] Minimum: [the original size of the volume]

**-i, --new-iops** <new\_iops>  
Performance Storage IOPS, between 100 and 6000 in multiples of 100 [only for performance volumes] **\*If no IOPS value is specified, the original IOPS value of the volume will be used.\*** Requirements: [If original IOPS/GB for the volume is less than 0.3, new IOPS/GB must also be less than 0.3. If original IOPS/GB for the volume is greater than or equal to 0.3, new IOPS/GB for the volume must also be greater than or equal to 0.3.]

**-t, --new-tier** <new\_tier>  
Endurance Storage Tier (IOPS per GB) [only for endurance volumes] **\*If no tier is specified, the original tier of the volume will be used.\*** Requirements: [If original IOPS/GB for the volume is 0.25, new IOPS/GB for the volume must also be 0.25. If original IOPS/GB for the volume is greater than 0.25, new IOPS/GB for the volume must also be greater than 0.25.]

**Options** 0.25|2|4|10

## Arguments

**VOLUME\_ID**  
Required argument

### 6.12.23 file volume-order

Order a file storage volume.

Valid size and iops options can be found here: <https://console.bluemix.net/docs/infrastructure/FileStorage/index.html#provisioning>

```
file volume-order [OPTIONS]
```

## Options

**--storage-type** <storage\_type>  
Type of file storage volume [required]

**Options** performancelendurance

**--size** <size>  
Size of file storage volume in GB [required]

**--iops** <iops>  
Performance Storage IOPs. Options vary based on storage size. [required for storage-type performance]

**--tier** <tier>  
Endurance Storage Tier (IOP per GB) [required for storage-type endurance]

**Options** 0.25|2|4|10

**--location** <location>  
Datacenter short name (e.g.: dal09) [required]

**--snapshot-size** <snapshot\_size>  
Optional parameter for ordering snapshot space along with endurance file storage; specifies the size (in GB) of snapshot space to order

**--service-offering** <service\_offering>  
The service offering package to use for placing the order. [optional, default is 'storage\_as\_a\_service']. enterprise and performance are depreciated

**Options** storage\_as\_a\_service|enterprise|performance

**--billing** <billing>  
Optional parameter for Billing rate (default to monthly)

**Options** hourly|monthly

### 6.12.24 file volume-limits

List number of block storage volumes limit per datacenter.

```
file volume-limits [OPTIONS]
```

#### Options

**--sortby** <sortby>  
Column to sort by

### 6.12.25 file snapshot-schedule-list

Lists snapshot schedules for a given volume

```
file snapshot-schedule-list [OPTIONS] VOLUME_ID
```

#### Arguments

**VOLUME\_ID**  
Required argument

## 6.13 Firewall Management

### 6.13.1 firewall add

Create new firewall.

**TARGET**: Id of the server the firewall will protect

```
firewall add [OPTIONS] TARGET
```

#### Options

**--firewall-type** <firewall\_type>  
Firewall type [required]

**Options** vslvlanserver

**--ha, --high-availability**  
High available firewall option

#### Arguments

**TARGET**  
Required argument

### 6.13.2 firewall cancel

Cancels a firewall.

```
firewall cancel [OPTIONS] IDENTIFIER
```

#### Arguments

**IDENTIFIER**

Required argument

### 6.13.3 firewall detail

Detail firewall.

```
firewall detail [OPTIONS] IDENTIFIER
```

#### Arguments

**IDENTIFIER**

Required argument

### 6.13.4 firewall edit

Edit firewall rules.

```
firewall edit [OPTIONS] IDENTIFIER
```

#### Arguments

**IDENTIFIER**

Required argument

### 6.13.5 firewall list

List firewalls.

```
firewall list [OPTIONS]
```

## 6.14 Global IP Addresses

### 6.14.1 globalip assign

Assigns the global IP to a target.

```
globalip assign [OPTIONS] IDENTIFIER TARGET
```

### Arguments

**IDENTIFIER**

Required argument

**TARGET**

Required argument

## 6.14.2 globalip cancel

Cancel global IP.

```
globalip cancel [OPTIONS] IDENTIFIER
```

### Arguments

**IDENTIFIER**

Required argument

## 6.14.3 globalip create

Creates a global IP.

```
globalip create [OPTIONS]
```

### Options

**-v6, --ipv6**

Order a IPv6 IP

**--test** <test>

test order

## 6.14.4 globalip list

List all global IPs.

```
globalip list [OPTIONS]
```

### Options

**--ip-version** <ip\_version>

Display only IPv4

**Options** v4|v6

## 6.14.5 globalip unassign

Unassigns a global IP from a target.

```
globalip unassign [OPTIONS] IDENTIFIER
```

### Arguments

#### **IDENTIFIER**

Required argument

## 6.15 Interacting with Hardware

### 6.15.1 hw bandwidth

Bandwidth data over date range. Bandwidth is listed in GB

Using just a date might get you times off by 1 hour, use T00:01 to get just the specific days data Timezones can also be included with the YYYY-MM-DDTHH:mm:ss.00000-HH:mm format.

Due to some rounding and date alignment details, results here might be slightly different than results in the control portal.

Example:

```
slcli hw bandwidth 1234 -s 2019-05-01T00:01 -e 2019-05-02T00:00:01.00000-12:00
```

```
hw bandwidth [OPTIONS] IDENTIFIER
```

### Options

- s, --start\_date** <start\_date>  
Start Date YYYY-MM-DD, YYYY-MM-DDTHH:mm:ss, [required]
- e, --end\_date** <end\_date>  
End Date YYYY-MM-DD, YYYY-MM-DDTHH:mm:ss [required]
- p, --summary\_period** <summary\_period>  
300, 600, 1800, 3600, 43200 or 86400 seconds [default: 3600]
- q, --quite\_summary**  
Only show the summary table [default: False]

### Arguments

#### **IDENTIFIER**

Required argument

### 6.15.2 hw cancel-reasons

Display a list of cancellation reasons.

```
hw cancel-reasons [OPTIONS]
```

### 6.15.3 hw cancel

Cancel a dedicated server.

```
hw cancel [OPTIONS] IDENTIFIER
```

#### Options

**--immediate**

          Cancels the server immediately (instead of on the billing anniversary)

**--comment** <comment>

          An optional comment to add to the cancellation ticket

**--reason** <reason>

          An optional cancellation reason. See cancel-reasons for a list of available options

#### Arguments

**IDENTIFIER**

          Required argument

### 6.15.4 hw create-options

Server order options for a given chassis.

```
hw create-options [OPTIONS]
```

### 6.15.5 hw create

Order/create a dedicated server.

```
hw create [OPTIONS]
```

#### Options

**-H, --hostname** <hostname>

          Host portion of the FQDN [required]

**-D, --domain** <domain>

          Domain portion of the FQDN [required]

**-s, --size** <size>

          Hardware size [required]

**-o, --os** <os>  
OS install code [required]

**-d, --datacenter** <datacenter>  
Datacenter shortname [required]

**--port-speed** <port\_speed>  
Port speeds [required]

**--billing** <billing>  
Billing rate [default: hourly]  
**Options** hourly|monthly

**-i, --postinstall** <postinstall>  
Post-install script to download

**-k, --key** <key>  
SSH keys to add to the root user (multiple occurrence permitted)

**--no-public**  
Private network only

**-e, --extra** <extra>  
Extra options (multiple occurrence permitted)

**--test**  
Do not actually create the server

**-t, --template** <template>  
A template file that defaults the command-line options

**--export** <export>  
Exports options to a template file

**--wait** <wait>  
Wait until the server is finished provisioning for up to X seconds before returning

Provides some basic functionality to order a server. *slcli order* has a more full featured method of ordering servers. This command only supports the FAST\_PROVISION type.

## 6.15.6 hw credentials

List server credentials.

```
hw credentials [OPTIONS] IDENTIFIER
```

### Arguments

**IDENTIFIER**  
Required argument

## 6.15.7 hw detail

Get details for a hardware device.

```
hw detail [OPTIONS] IDENTIFIER
```



## Options

- passwords**  
Show passwords (check over your shoulder!)
- price**  
Show associated prices

## Arguments

- IDENTIFIER**  
Required argument

### 6.15.8 hw edit

Edit hardware details.

```
hw edit [OPTIONS] IDENTIFIER
```

## Options

- D, --domain <domain>**  
Domain portion of the FQDN
- F, --userfile <userfile>**  
Read userdata from file
- g, --tag <tag>**  
Tags to set or empty string to remove all
- H, --hostname <hostname>**  
Host portion of the FQDN
- u, --userdata <userdata>**  
User defined metadata string
- public-speed <public\_speed>**  
Public port speed. -1 is best speed available  
**Options** 0|10|100|1000|10000|-1
- private-speed <private\_speed>**  
Private port speed. -1 is best speed available  
**Options** 0|10|100|1000|10000|-1

## Arguments

- IDENTIFIER**  
Required argument

When setting port speed, use “-1” to indicate best possible configuration. Using 10/100/1000/10000 on a server with a redundant interface may result the interface entering a degraded state. See [setPublicNetworkInterfaceSpeed](#) for more information.

### 6.15.9 hw list

List hardware servers.

```
hw list [OPTIONS]
```

#### Options

- c, --cpu** <cpu>  
Filter by number of CPU cores
- D, --domain** <domain>  
Filter by domain
- d, --datacenter** <datacenter>  
Filter by datacenter
- H, --hostname** <hostname>  
Filter by hostname
- m, --memory** <memory>  
Filter by memory in gigabytes
- n, --network** <network>  
Filter by network port speed in Mbps
- tag** <tag>  
Filter by tags (multiple occurrence permitted)
- sortby** <sortby>  
Column to sort by [default: hostname]
- columns** <columns>  
Columns to display. [options: guid, primary\_ip, backend\_ip, datacenter, action, created\_by, tags] [default: id,hostname,primary\_ip,backend\_ip,datacenter,action]
- l, --limit** <limit>  
How many results to get in one api call, default is 100 [default: 100]

### 6.15.10 hw power-cycle

Power cycle a server.

```
hw power-cycle [OPTIONS] IDENTIFIER
```

#### Arguments

**IDENTIFIER**  
Required argument

### 6.15.11 hw power-off

Power off an active server.

```
hw power-off [OPTIONS] IDENTIFIER
```

### Arguments

**IDENTIFIER**

Required argument

## 6.15.12 hw power-on

Power on a server.

```
hw power-on [OPTIONS] IDENTIFIER
```

### Arguments

**IDENTIFIER**

Required argument

## 6.15.13 hw reboot

Reboot an active server.

```
hw reboot [OPTIONS] IDENTIFIER
```

### Options

**--hard, --soft**

Perform a hard or soft reboot

### Arguments

**IDENTIFIER**

Required argument

## 6.15.14 hw reload

Reload operating system on a server.

```
hw reload [OPTIONS] IDENTIFIER
```

### Options

**-i, --postinstall <postinstall>**

Post-install script to download (Only HTTPS executes, HTTP leaves file in /root)

**-k, --key <key>**

SSH keys to add to the root user (multiple occurrence permitted)

## Arguments

### **IDENTIFIER**

Required argument

## 6.15.15 hw rescue

Reboot server into a rescue image.

```
hw rescue [OPTIONS] IDENTIFIER
```

## Arguments

### **IDENTIFIER**

Required argument

## 6.15.16 hw reflash-firmware

Reflash server firmware.

```
hw reflash-firmware [OPTIONS] IDENTIFIER
```

## Arguments

### **IDENTIFIER**

Required argument

Reflash here means the current version of the firmware running on your server will be re-flashed onto the selected hardware. This does require a reboot. See *slcli hw update-firmware* if you want the newest version.

## 6.15.17 hw update-firmware

Update server firmware.

```
hw update-firmware [OPTIONS] IDENTIFIER
```

## Arguments

### **IDENTIFIER**

Required argument

This function updates the firmware of a server. If already at the latest version, no software is installed.

## 6.15.18 hw toggle-ipmi

Toggle the IPMI interface on and off

```
hw toggle-ipmi [OPTIONS] IDENTIFIER
```

## Options

**--enable, --disable**  
Whether enable (DEFAULT) or disable the interface.

## Arguments

**IDENTIFIER**  
Required argument

### 6.15.19 hw ready

Check if a server is ready.

```
hw ready [OPTIONS] IDENTIFIER
```

## Options

**--wait <wait>**  
Seconds to wait [default: 0]

## Arguments

**IDENTIFIER**  
Required argument

### 6.15.20 hw dns-sync

Sync DNS records.

```
hw dns-sync [OPTIONS] IDENTIFIER
```

## Options

**-a, --a-record**  
Sync the A record for the host

**--aaaa-record**  
Sync the AAAA record for the host

**--ptr**  
Sync the PTR record for the host

**--ttl <ttl>**  
Sets the TTL for the A and/or PTR records [default: 7200]

## Arguments

**IDENTIFIER**  
Required argument

## 6.16 Disk Image Commands

### 6.16.1 image delete

Delete an image.

```
image delete [OPTIONS] IDENTIFIER
```

#### Arguments

**IDENTIFIER**

Required argument

### 6.16.2 image detail

Get details for an image.

```
image detail [OPTIONS] IDENTIFIER
```

#### Arguments

**IDENTIFIER**

Required argument

### 6.16.3 image edit

Edit details of an image.

```
image edit [OPTIONS] IDENTIFIER
```

#### Options

**--name** <name>  
Name of the image

**--note** <note>  
Additional note for the image

**--tag** <tag>  
Tags for the image

#### Arguments

**IDENTIFIER**

Required argument

## 6.16.4 image list

List images.

```
image list [OPTIONS]
```

### Options

- name** <name>  
Filter on image name
- public, --private**  
Display only public or private images

## 6.16.5 image import

Import an image.

The URI for an object storage object (.vhd/.iso file) of the format: swift://<objectStorageAccount>@<cluster>/<container>/<objectPath> or cos://<regionName>/<bucketName>/<objectPath> if using IBM Cloud Object Storage

```
image import [OPTIONS] NAME URI
```

### Options

- note** <note>  
The note to be applied to the imported template
- os-code** <os\_code>  
The referenceCode of the operating system software description for the imported VHD, ISO, or RAW image
- ibm-api-key** <ibm\_api\_key>  
The IBM Cloud API Key with access to IBM Cloud Object Storage instance and IBM KeyProtect instance. For help creating this key see <https://console.ibm.com/docs/services/cloud-object-storage/iam/users-serviceids.html#serviceidapikey>
- root-key-crn** <root\_key\_crn>  
CRN of the root key in your KMS instance
- wrapped-dek** <wrapped\_dek>  
Wrapped Data Encryption Key provided by IBM KeyProtect. For more info see <https://console.ibm.com/docs/services/key-protect/wrap-keys.html#wrap-keys>
- cloud-init**  
Specifies if image is cloud-init
- byol**  
Specifies if image is bring your own license
- is-encrypted**  
Specifies if image is encrypted

## Arguments

### NAME

Required argument

### URI

Required argument

## 6.16.6 image export

Export an image to object storage.

The URI for an object storage object (.vhd/.iso file) of the format: swift://<objectStorageAccount>@<cluster>/<container>/<objectPath> or cos://<regionName>/<bucketName>/<objectPath> if using IBM Cloud Object Storage

```
image export [OPTIONS] IDENTIFIER URI
```

## Options

**--ibm-api-key** <ibm\_api\_key>

The IBM Cloud API Key with access to IBM Cloud Object Storage instance. For help creating this key see <https://console.ibm.com/docs/services/cloud-object-storage/iam/users-serviceids.html#serviceidapikeys>

## Arguments

### IDENTIFIER

Required argument

### URI

Required argument

## 6.17 Interacting with IPSEC Tunnels

The IPSEC *Command-line Interface* commands can be used to configure an existing IPSEC tunnel context. Subnets in the SoftLayer private network can be associated to the tunnel context along with user-defined remote subnets. Address translation entries may also be defined to provide NAT functionality from static subnet IP addresses associated with the tunnel context to user-defined remote subnet IP addresses.

---

**Note:** Most CLI actions that affect an IPSEC tunnel context do not result in configuration changes to SoftLayer network devices. A separate *configure* command is available to issue a device configuration request.

---

To see more information about the IPSEC tunnel context module and API interaction, see *IPSEC Module* documentation.

### 6.17.1 ipsec list

A list of all IPSEC tunnel contexts associated with the current user's account can be retrieved via the `ipsec list` command. This provides a brief overview of all tunnel contexts and can be used to retrieve an individual context's identifier, which all other CLI commands require.



```

$ slcli ipsec list
:.....:
↪:.....:
: id : name : friendly name : internal peer IP address : remote peer IP address_
↪:      created      :
:.....:
↪:.....:
: 445 : ipsec038 : ipsec tunnel : 173.192.250.79 : 158.85.80.22 _
↪: 2012-03-05T14:07:34-06:00 :
:.....:
↪:.....:

```

## 6.17.2 ipsec detail

More detailed information can be retrieved for an individual context using the `ipsec detail` command. Using the detail command, information about associated internal subnets, remote subnets, static subnets, service subnets and address translations may also be retrieved using multiple instances of the `-i|--include` option.

```

$ slcli ipsec detail 445 -i at -i is -i rs -i sr -i ss
Context Details:
:.....:
: name : value :
:.....:
: id : 445 :
: name : ipsec038 :
: friendly name : ipsec tunnel :
: internal peer IP address : 173.192.250.79 :
: remote peer IP address : 158.85.80.22 :
: advanced configuration flag : 0 :
: preshared key : secret :
: phase 1 authentication : MD5 :
: phase 1 diffie hellman group : 0 :
: phase 1 encryption : DES :
: phase 1 key life : 240 :
: phase 2 authentication : MD5 :
: phase 2 diffie hellman group : 1 :
: phase 2 encryption : DES :
: phase 2 key life : 240 :
: phase 2 perfect forward secrecy : 1 :
: created : 2012-03-05T14:07:34-06:00 :
: modified : 2017-05-17T12:01:33-06:00 :
:.....:
Address Translations:
:.....:
↪:.....:
: id : static IP address : static IP address id : remote IP address : remote IP_
↪address id : note :
:.....:
↪:.....:
: 15920 : 10.1.249.86 : 9791681 : 158.85.80.22 : 98828_
↪ : windows server :
: 15918 : 10.1.249.84 : 9791679 : 158.85.80.20 : 98824_
↪ : unix server :
:.....:
↪:.....:
Internal Subnets:

```

(continues on next page)

(continued from previous page)

```

:.....:.....:.....:.....:
:  id   : network identifier : cidr : note :
:.....:.....:.....:.....:
: 180767 : 10.28.67.128 : 26 : :
:.....:.....:.....:.....:
Remote Subnets:
:.....:.....:.....:.....:
:  id   : network identifier : cidr : note :
:.....:.....:.....:.....:
: 7852 : 158.85.80.20 : 30 : :
:.....:.....:.....:.....:
Static Subnets:
:.....:.....:.....:.....:
:  id   : network identifier : cidr : note :
:.....:.....:.....:.....:
: 231807 : 10.1.249.84 : 30 : :
:.....:.....:.....:.....:
Service Subnets:
:.....:.....:.....:.....:
:  id   : network identifier : cidr : note :
:.....:.....:.....:.....:
: 162079 : 10.0.80.0 : 25 : :
:.....:.....:.....:.....:

```

### 6.17.3 ipsec update

Most values listed in the tunnel context detail printout can be modified using the `ipsec update` command. The following is given when executing with the `-h|--help` option and highlights all properties that may be modified.

```

$ slcli ipsec update -h
Usage: slcli ipsec update [OPTIONS] CONTEXT_ID

Update tunnel context properties.

Updates are made atomically, so either all are accepted or none are.

Key life values must be in the range 120-172800.

Phase 2 perfect forward secrecy must be in the range 0-1.

A separate configuration request should be made to realize changes on
network devices.

Options:
  --friendly-name TEXT           Friendly name value
  --remote-peer TEXT            Remote peer IP address value
  --preshared-key TEXT          Preshared key value
  --p1-auth, --phase1-auth [MD5|SHA1|SHA256]
                                Phase 1 authentication value
  --p1-crypto, --phase1-crypto [DES|3DES|AES128|AES192|AES256]
                                Phase 1 encryption value
  --p1-dh, --phase1-dh [0|1|2|5] Phase 1 diffie hellman group value
  --p1-key-ttl, --phase1-key-ttl INTEGER RANGE
                                Phase 1 key life value
  --p2-auth, --phase2-auth [MD5|SHA1|SHA256]

```

(continues on next page)

(continued from previous page)

```

Phase 2 authentication value
--p2-crypto, --phase2-crypto [DES|3DES|AES128|AES192|AES256]
Phase 2 encryption value
--p2-dh, --phase2-dh [0|1|2|5] Phase 2 diffie hellman group value
--p2-forward-secrecy, --phase2-forward-secrecy INTEGER RANGE
Phase 2 perfect forward secrecy value
--p2-key-ttl, --phase2-key-ttl INTEGER RANGE
Phase 2 key life value
-h, --help Show this message and exit.

```

### 6.17.4 ipsec configure

A request to configure SoftLayer network devices for a given tunnel context can be issued using the `ipsec configure` command.

**Note:** Once a configuration request is received, the IPSEC tunnel context will be placed into an unmodifiable state, and further changes against the tunnel context will be prevented. Once configuration changes have been made, the tunnel context may again be modified. The unmodifiable state of a tunnel context is indicated by an *advanced configuration flag* value of 1.

### 6.17.5 ipsec subnet-add

Internal, remote and service subnets can be associated to an IPSEC tunnel context using the `ipsec subnet-add` command. Additionally, remote subnets can be created using this same command, which will then be associated to the targeted tunnel context.

**Note:** The targeted subnet type must be specified. A subnet id must be provided when associating internal and service subnets. Either a subnet id or a network identifier must be provided when associating remote subnets. If a network identifier is provided when associating a remote subnet, that subnet will first be created and then associated to the tunnel context.

The following is an example of associating an internal subnet to a tunnel context.

```

$ slcli ipsec subnet-add 445 --subnet-id 180767 --subnet-type internal
Added internal subnet #180767

```

The following is an example of creating and associating a remote subnet to a tunnel context.

```

$ slcli ipsec subnet-add 445 --subnet-type remote --network 50.100.0.0/26
Created subnet 50.100.0.0/26 #21268
Added remote subnet #21268

```

### 6.17.6 ipsec subnet-remove

Internal, remote and service subnets can be disassociated from an IPSEC tunnel context via the `ipsec subnet-remove` command.

**Note:** The targeted subnet id and type must be specified. When disassociating remote subnets, that subnet record will also be deleted.

---

The following is an example of disassociating an internal subnet from a tunnel context.

```
$ slcli ipsec subnet-remove 445 --subnet-id 180767 --subnet-type internal
Removed internal subnet #180767
```

### 6.17.7 ipsec translation-add

Address translation entries can be added to a tunnel context to provide NAT functionality from a statically routed subnet associated with the tunnel context to a remote subnet. This action is performed with the `ipsec translation-add` command.

**Note:** Both static and remote IP address values must be specified. An optional note value may also be provided.

---

The following is an example of adding a new address translation entry.

```
$ slcli ipsec translation-add 445 --static-ip 10.1.249.87 --remote-ip 50.100.0.10 --
↳note 'email server'
Created translation from 10.1.249.87 to 50.100.0.10 #15922
```

### 6.17.8 ipsec translation-remove

Address translation entries can be removed using the `ipsec translation-remove` command.

The following is an example of removing an address translation entry.

```
$ slcli ipsec translation-remove 445 --translation-id 15922
Removed translation #15922
```

### 6.17.9 ipsec translation-update

Address translation entries may also be modified using the `ipsec translation-update` command.

The following is an example of updating an existing address translation entry.

```
$ slcli ipsec translation-update 445 --translation-id 15924 --static-ip 10.1.249.86 --
↳remote-ip 50.100.0.8 --note 'new email server'
Updated translation #15924
```

## 6.18 LoadBalancers

These commands were added in version 5.8.0

## 6.18.1 LBaaS Commands

- [LBaaS Product](#)
- [LBaaS Documentation](#)

### loadbal detail

Get Load Balancer as a Service details.

```
loadbal detail [OPTIONS] IDENTIFIER
```

### Arguments

#### IDENTIFIER

Required argument

### loadbal list

List active Load Balancer as a Service devices.

```
loadbal list [OPTIONS]
```

### loadbal health

Manage LBaaS health checks.

```
loadbal health [OPTIONS] IDENTIFIER
```

### Options

- uuid** <uuid>  
Health check UUID to modify. [required]
- i, --interval** <interval>  
Seconds between checks. [2-60]
- r, --retry** <retry>  
Number of times before marking as DOWN. [1-10]
- t, --timeout** <timeout>  
Seconds to wait for a connection. [1-59]
- u, --url** <url>  
Url path for HTTP/HTTPS checks.

### Arguments

#### IDENTIFIER

Required argument

### loadbal member-add

Add a new LBaaS members.

```
loadbal member-add [OPTIONS] IDENTIFIER
```

#### Options

- private, --public**  
Private or public IP of the new member. [required]
- m, --member** <member>  
Member IP address. [required]
- w, --weight** <weight>  
Weight of this member.

#### Arguments

**IDENTIFIER**  
Required argument

### loadbal member-remove

Remove a LBaaS member.

Member UUID can be found from *slcli lb detail*.

```
loadbal member-remove [OPTIONS] IDENTIFIER
```

#### Options

- m, --member** <member>  
Member UUID [required]

#### Arguments

**IDENTIFIER**  
Required argument

### loadbal pool-add

Adds a listener to the identifier LB

```
loadbal pool-add [OPTIONS] IDENTIFIER
```

## Options

- P, --frontProtocol** <frontprotocol>  
Protocol type to use for incoming connections [default: HTTP]  
**Options** HTTP|HTTPS|TCP
- p, --backProtocol** <backprotocol>  
Protocol type to use when connecting to backend servers. Defaults to whatever `--frontProtocol` is.  
**Options** HTTP|HTTPS|TCP
- f, --frontPort** <frontport>  
Internet side port [required]
- b, --backPort** <backport>  
Private side port [required]
- m, --method** <method>  
Balancing Method [default: ROUNDROBIN]  
**Options** ROUNDROBIN|LEASTCONNECTION|WEIGHTED\_RR
- c, --connections** <connections>  
Maximum number of connections to allow.
- s, --sticky**  
Make sessions sticky based on `source_ip`.
- x, --sslCert** <sslcert>  
SSL certificate ID. See `slcli ssl list`

## Arguments

- IDENTIFIER**  
Required argument

## loadbal pool-edit

Updates a listener's configuration.

LISTENER should be a UUID, and can be found from `slcli lb detail <IDENTIFIER>`

```
loadbal pool-edit [OPTIONS] IDENTIFIER LISTENER
```

## Options

- P, --frontProtocol** <frontprotocol>  
Protocol type to use for incoming connections  
**Options** HTTP|HTTPS|TCP
- p, --backProtocol** <backprotocol>  
Protocol type to use when connecting to backend servers. Defaults to whatever `--frontProtocol` is.  
**Options** HTTP|HTTPS|TCP
- f, --frontPort** <frontport>  
Internet side port

- b, --backPort** <backport>  
Private side port
- m, --method** <method>  
Balancing Method  
**Options** ROUNDROBIN|LEASTCONNECTION|WEIGHTED\_RR
- c, --connections** <connections>  
Maximum number of connections to allow.
- s, --sticky**  
Make sessions sticky based on source\_ip.
- x, --sslCert** <sslcert>  
SSL certificate ID. See *slcli ssl list*

## Arguments

- IDENTIFIER**  
Required argument
- LISTENER**  
Required argument

## loadbal pool-delete

Removes the listener from identified LBaaS instance  
LISTENER should be a UUID, and can be found from *slcli lb detail <IDENTIFIER>*

```
loadbal pool-delete [OPTIONS] IDENTIFIER LISTENER
```

## Arguments

- IDENTIFIER**  
Required argument
- LISTENER**  
Required argument

## loadbal l7pool-add

Adds a new l7 pool  
-S is in colon delimited format to make grouping IP:port:weight a bit easier.

```
loadbal l7pool-add [OPTIONS] IDENTIFIER
```

## Options

- n, --name** <name>  
Name for this L7 pool. [required]



- m, --method** <method>  
Balancing Method. [default: ROUNDROBIN]  
**Options** ROUNDROBIN|LEASTCONNECTION|WEIGHTED\_RR
- P, --protocol** <protocol>  
Protocol type to use for incoming connections [default: HTTP]  
**Options** HTTP|HTTPS
- S, --server** <server>  
Backend servers that are part of this pool. Format is colon delimited. BACKEND\_IP:PORT:WEIGHT. eg. 10.0.0.1:80:50 (multiple occurrence permitted) [required]
- healthPath** <healthpath>  
Health check path. [default: /]
- healthInterval** <healthinterval>  
Health check interval between checks. [default: 5]
- healthRetry** <healthretry>  
Health check number of times before marking as DOWN. [default: 2]
- healthTimeout** <healthtimeout>  
Health check timeout. [default: 2]
- s, --sticky**  
Make sessions sticky based on source\_ip.

## Arguments

**IDENTIFIER**  
Required argument

### loadbal l7pool-del

Deletes the identified pool

Identifier is L7Pool Id. NOT the UUID

```
loadbal l7pool-del [OPTIONS] IDENTIFIER
```

## Arguments

**IDENTIFIER**  
Required argument

### loadbal order

Creates a LB. Protocols supported are TCP, HTTP, and HTTPS.

```
loadbal order [OPTIONS]
```

## Options

- n, --name** <name>  
Label for this loadbalancer. [required]
- d, --datacenter** <datacenter>  
Datacenter shortname (dal13). [required]
- l, --label** <label>  
A descriptive label for this loadbalancer.
- f, --frontend** <frontend>  
PROTOCOL:PORT string for incoming internet connections. [default: HTTP:80; required]
- b, --backend** <backend>  
PROTOCOL:PORT string for connecting to backend servers. [default: HTTP:80; required]
- m, --method** <method>  
Balancing Method. [default: ROUNDROBIN]  
**Options** ROUNDROBIN|LEASTCONNECTION|WEIGHTED\_RR
- s, --subnet** <subnet>  
Private subnet Id to order the LB on. See *slcli lb order-options* [required]
- public**  
Use a Public to Public loadbalancer. [default: False]
- verify**  
Only verify an order, dont actually create one. [default: False]

## loadbal order-options

Prints options for order a LBaaS

```
loadbal order-options [OPTIONS]
```

## Options

- d, --datacenter** <datacenter>  
Show only selected datacenter, use shortname (dal13) format.

## loadbal cancel

Cancels a LBaaS instance

```
loadbal cancel [OPTIONS] IDENTIFIER
```

## Arguments

- IDENTIFIER**  
Required argument

## 6.18.2 NetScaler Commands

### loadbal ns-detail

Get Netscaler details.

```
loadbal ns-detail [OPTIONS] IDENTIFIER
```

#### Arguments

##### **IDENTIFIER**

Required argument

### loadbal ns-list

List active Netscaler devices.

```
loadbal ns-list [OPTIONS]
```

## 6.19 Object Storage Commands

### 6.19.1 object-storage accounts

List object storage accounts.

```
object-storage accounts [OPTIONS]
```

### 6.19.2 object-storage endpoints

List object storage endpoints.

```
object-storage endpoints [OPTIONS]
```

### 6.19.3 object-storage credential list

Retrieve credentials used for generating an AWS signature. Max of 2.

```
object-storage credential list [OPTIONS] IDENTIFIER
```

#### Arguments

##### **IDENTIFIER**

Required argument

### 6.19.4 object-storage credential limit

Credential limits for this IBM Cloud Object Storage account.

```
object-storage credential limit [OPTIONS] IDENTIFIER
```

#### Arguments

##### **IDENTIFIER**

Required argument

### 6.19.5 object-storage credential delete

Delete the credential of an Object Storage Account.

```
object-storage credential delete [OPTIONS] IDENTIFIER
```

#### Options

**-c, --credential\_id** <credential\_id>

This is the credential id associated with the volume

#### Arguments

##### **IDENTIFIER**

Required argument

### 6.19.6 object-storage credential create

Create credentials for an IBM Cloud Object Storage Account

```
object-storage credential create [OPTIONS] IDENTIFIER
```

#### Arguments

##### **IDENTIFIER**

Required argument

## 6.20 Ordering

The Order *Command-line Interface* commands can be used to build an order for any product in the SoftLayer catalog.

The basic flow for ordering goes something like this...

1. package-list
2. category-list <package key name>
3. item-list <package key name>

4. place <package key name> <item key names> <location>

### 6.20.1 order package-list

List packages that can be ordered via the placeOrder API.

```
# List out all packages for ordering
slcli order package-list

# List out all packages with "server" in the name
slcli order package-list --keyword server

# Select only specifict package types
slcli order package-list --package_type BARE_METAL_CPU
```

```
order package-list [OPTIONS]
```

#### Options

**--keyword** <keyword>

A word (or string) used to filter package names.

**--package\_type** <package\_type>

The keyname for the type of package. BARE\_METAL\_CPU for example

#### Note:

- CLOUD\_SERVER: These are Virtual Servers
- BARE\_METAL\_INSTANCE: Hourly Bare Metal
- BARE\_METAL\_SERVER: Other monthly server types
- *#\_PROC\_#\_DRIVES*: Packages in this format will contain only this CPU model and Drive bays
- ADDITIONAL\_PRODUCTS: Additional IPs, Vlans, SSL certs and other things are in here
- NETWORK\_GATEWAY\_APPLIANCE: Vyattas

Bluemix services listed here may still need to be ordered through the Bluemix CLI/Portal

### 6.20.2 order package-locations

List Datacenters a package can be ordered in.

Use the location Key Name to place orders

```
order package-locations [OPTIONS] PACKAGE_KEYNAME
```

#### Arguments

**PACKAGE\_KEYNAME**

Required argument

### 6.20.3 order category-list

List the categories of a package.

```
# List the categories of Bare Metal servers
slcli order category-list BARE_METAL_SERVER

# List the required categories for Bare Metal servers
slcli order category-list BARE_METAL_SERVER --required
```

```
order category-list [OPTIONS] PACKAGE_KEYNAME
```

#### Options

##### **--required**

List only the required categories for the package

#### Arguments

##### **PACKAGE\_KEYNAME**

Required argument

Shows all the available categories for a certain package, useful in finding the required categories. Categories that are required will need to have a corresponding item included with any orders

These are all the required categories for BARE\_METAL\_SERVER

```
$ slcli order category-list BARE_METAL_SERVER
:.....:.....:.....:
:      name      :      categoryCode      : isRequired :
:.....:.....:.....:
:      Server     :      server             :      Y      :
:      Operating System :      os                 :      Y      :
:      RAM        :      ram                 :      Y      :
:      Disk Controller :      disk_controller    :      Y      :
:      First Hard Drive :      disk0              :      Y      :
:      Public Bandwidth :      bandwidth         :      Y      :
:      Uplink Port Speeds :      port_speed        :      Y      :
:      Remote Management :      remote_management  :      Y      :
:      Primary IP Addresses :      pri_ip_addresses  :      Y      :
:      VPN Management - Private Network :      vpn_management    :      Y      :
:.....:.....:.....:
```

### 6.20.4 order item-list

List package items used for ordering.

The item keyNames listed can be used with *slcli order place* to specify the items that are being ordered in the package.

---

**Note:** Items with a numbered category, like disk0 or gpu0, can be included multiple times in an order to match how many of the item you want to order.

---

```
# List all items in the VSI package
slcli order item-list CLOUD_SERVER

# List Ubuntu OSes from the os category of the Bare Metal package
slcli order item-list BARE_METAL_SERVER --category os --keyword ubuntu
```

```
order item-list [OPTIONS] PACKAGE_KEYNAME
```

## Options

**--keyword** <keyword>  
A word (or string) used to filter item names.

**--category** <category>  
Category code to filter items by

## Arguments

**PACKAGE\_KEYNAME**  
Required argument

Shows all the prices for a given package. Collect all the items you want included on your server. Don't forget to include the required category items. If forgotten, `order place` will tell you about it.

### 6.20.5 order preset-list

List package presets.

---

**Note:** Presets are set CPU / RAM / Disk allotments. You still need to specify required items. Some packages do not have presets.

---

```
# List the presets for Bare Metal servers
slcli order preset-list BARE_METAL_SERVER

# List the Bare Metal server presets that include a GPU
slcli order preset-list BARE_METAL_SERVER --keyword gpu
```

```
order preset-list [OPTIONS] PACKAGE_KEYNAME
```

## Options

**--keyword** <keyword>  
A word (or string) used to filter preset names.

## Arguments

**PACKAGE\_KEYNAME**  
Required argument

## 6.20.6 order place

Place or verify an order.

This CLI command is used for placing/verifying an order of the specified package in the given location (denoted by a datacenter's long name). Orders made via the CLI can then be converted to be made programmatically by calling `SoftLayer.OrderingManager.place_order()` with the same keynames.

Packages for ordering can be retrieved from `slcli order package-list`. Presets for ordering can be retrieved from `slcli order preset-list` (not all packages have presets)

Items can be retrieved from `slcli order item-list`. In order to find required items for the order, use `slcli order category-list`, and then provide the `--category` option for each category code in `slcli order item-list`.

Example:

```
# Order an hourly VSI with 4 CPU, 16 GB RAM, 100 GB SAN disk,
# Ubuntu 16.04, and 1 Gbps public & private uplink in dal13
slcli order place --billing hourly CLOUD_SERVER DALLAS13 \
  GUEST_CORES_4 \
  RAM_16_GB \
  REBOOT_REMOTE_CONSOLE \
  1_GBPS_PUBLIC_PRIVATE_NETWORK_UPLINKS \
  BANDWIDTH_0_GB_2 \
  1_IP_ADDRESS \
  GUEST_DISK_100_GB_SAN \
  OS_UBUNTU_16_04_LTS_XENIAL_XERUS_MINIMAL_64_BIT_FOR_VSI \
  MONITORING_HOST_PING \
  NOTIFICATION_EMAIL_AND_TICKET \
  AUTOMATED_NOTIFICATION \
  UNLIMITED_SSL_VPN_USERS_1_PPTP_VPN_USER_PER_ACCOUNT \
  NESSUS_VULNERABILITY_ASSESSMENT_REPORTING \
  --extras '{"virtualGuests": [{"hostname": "test", "domain": "softlayer.com"}]}' \
  --complex-type SoftLayer_Container_Product_Order_Virtual_Guest
```

```
order place [OPTIONS] PACKAGE_KEYNAME LOCATION [ORDER_ITEMS]...
```

### Options

**--preset** <preset>

The order preset (if required by the package)

**--verify**

Flag denoting whether or not to only verify the order, not place it

**--quantity** <quantity>

The quantity of the item being ordered

**--billing** <billing>

Billing rate [default: hourly]

**Options** hourly|monthly

**--complex-type** <complex\_type>

The complex type of the order. Starts with 'SoftLayer\_Container\_Product\_Order'.

**--extras** <extras>

JSON string denoting extra data that needs to be sent with the order



## Arguments

### PACKAGE\_KEYNAME

Required argument

### LOCATION

Required argument

### ORDER\_ITEMS

Optional argument(s)

Now that you have the package you want, the prices needed, and found a location, it is time to place an order.

## 6.20.7 order place <preset>

```
$ slcli --really order place --preset D2620V4_64GB_2X1TB_SATA_RAID_1 BARE_METAL_
↪SERVER TORONTO \
  OS_UBUNTU_16_04_LTS_XENIAL_XERUS_64_BIT \
  BANDWIDTH_0_GB_2 \
  1_GBPS_PRIVATE_NETWORK_UPLINK \
  REBOOT_KVM_OVER_IP 1_IP_ADDRESS \
  UNLIMITED_SSL_VPN_USERS_1_PPTP_VPN_USER_PER_ACCOUNT \
  --extras '{"hardware": [{"hostname": "testOrder", "domain": "cgallo.com"}]}' \
  --complex-type SoftLayer_Container_Product_Order_Hardware_Server
```

## 6.20.8 order place <Virtual Server>

```
$ slcli order place --billing hourly CLOUD_SERVER DALLAS13 \
  GUEST_CORES_4 \
  RAM_16_GB \
  REBOOT_REMOTE_CONSOLE \
  1_GBPS_PUBLIC_PRIVATE_NETWORK_UPLINKS \
  BANDWIDTH_0_GB_2 \
  1_IP_ADDRESS \
  GUEST_DISK_100_GB_SAN \
  OS_UBUNTU_16_04_LTS_XENIAL_XERUS_MINIMAL_64_BIT_FOR_VSI \
  MONITORING_HOST_PING \
  NOTIFICATION_EMAIL_AND_TICKET \
  AUTOMATED_NOTIFICATION \
  UNLIMITED_SSL_VPN_USERS_1_PPTP_VPN_USER_PER_ACCOUNT \
  NESSUS_VULNERABILITY_ASSESSMENT_REPORTING \
  --extras '{"virtualGuests": [{"hostname": "test", "domain": "softlayer.com"}]}' \
  --complex-type SoftLayer_Container_Product_Order_Virtual_Guest
```

## 6.21 Quotes

### 6.21.1 order quote

View and Order a quote

```
order quote [OPTIONS] QUOTE
```

## Options

- verify**  
If specified, will only show what the quote will order, will NOT place an order [default: False]
- quantity** <quantity>  
The quantity of the item being ordered if different from quoted value
- complex-type** <complex\_type>  
The complex type of the order. Starts with 'SoftLayer\_Container\_Product\_Order'. [default: SoftLayer\_Container\_Product\_Order\_Hardware\_Server]
- u, --userdata** <userdata>  
User defined metadata string
- F, --userfile** <userfile>  
Read userdata from file
- i, --postinstall** <postinstall>  
Post-install script to download
- k, --key** <key>  
SSH keys to add to the root user (multiple occurrence permitted)
- fqdn** <fqdn>  
<hostname>.<domain.name.tld> formatted name to use. Specify one fqdn per server (multiple occurrence permitted) [required]
- image** <image>  
Image ID. See: 'slcli image list' for reference

## Arguments

- QUOTE**  
Required argument

### 6.21.2 order quote-list

List all active quotes on an account

```
order quote-list [OPTIONS]
```

### 6.21.3 order quote-detail

View a quote

```
order quote-detail [OPTIONS] QUOTE
```

## Arguments

- QUOTE**  
Required argument

## 6.21.4 order place-quote

Place a quote.

This CLI command is used for creating a quote of the specified package in the given location (denoted by a data-center's long name). Orders made via the CLI can then be converted to be made programmatically by calling `SoftLayer.OrderingManager.place_quote()` with the same keynames.

Packages for ordering can be retrieved from `slcli order package-list`. Presets for ordering can be retrieved from `slcli order preset-list` (not all packages have presets)

Items can be retrieved from `slcli order item-list`. In order to find required items for the order, use `slcli order category-list`, and then provide the `--category` option for each category code in `slcli order item-list`.

Example:

```
# Place quote a VSI with 4 CPU, 16 GB RAM, 100 GB SAN disk,
# Ubuntu 16.04, and 1 Gbps public & private uplink in dal13
slcli order place-quote --name "foobar" --send-email CLOUD_SERVER DALLAS13 \
    GUEST_CORES_4 \
    RAM_16_GB \
    REBOOT_REMOTE_CONSOLE \
    1_GBPS_PUBLIC_PRIVATE_NETWORK_UPLINKS \
    BANDWIDTH_0_GB_2 \
    1_IP_ADDRESS \
    GUEST_DISK_100_GB_SAN \
    OS_UBUNTU_16_04_LTS_XENIAL_XERUS_MINIMAL_64_BIT_FOR_VSI \
    MONITORING_HOST_PING \
    NOTIFICATION_EMAIL_AND_TICKET \
    AUTOMATED_NOTIFICATION \
    UNLIMITED_SSL_VPN_USERS_1_PPTP_VPN_USER_PER_ACCOUNT \
    NESSUS_VULNERABILITY_ASSESSMENT_REPORTING \
    --extras '{"virtualGuests": [{"hostname": "test", "domain": "softlayer.com"}]}' \
    --complex-type SoftLayer_Container_Product_Order_Virtual_Guest
```

```
order place-quote [OPTIONS] PACKAGE_KEYNAME LOCATION [ORDER_ITEMS]...
```

### Options

**--preset** <preset>

The order preset (if required by the package)

**--name** <name>

A custom name to be assigned to the quote (optional)

**--send-email**

The quote will be sent to the email address associated with your user.

**--complex-type** <complex\_type>

The complex type of the order. Starts with 'SoftLayer\_Container\_Product\_Order'.

**--extras** <extras>

JSON string denoting extra data that needs to be sent with the order

### Arguments

**PACKAGE\_KEYNAME**

Required argument

**LOCATION**

Required argument

**ORDER\_ITEMS**

Optional argument(s)

## 6.22 Reports

There are a few report type commands in the SLCLI.

### 6.22.1 summary

Account summary.

```
summary [OPTIONS]
```

#### Options

**--sortby** <sortby>

Column to sort by

**Options** datacenter|hardware|virtual\_servers|vlans|subnets|public\_ips

A list of datacenters, and how many servers, VSI, vlans, subnets and public\_ips are in each.

### 6.22.2 report bandwidth

Bandwidth report for every pool/server.

This reports on the total data transfered for each virtual sever, hardware server and bandwidth pool.

```
report bandwidth [OPTIONS]
```

#### Options

**--start** <start>

datetime in the format 'YYYY-MM-DD' or 'YYYY-MM-DD HH:MM:SS'

**--end** <end>

datetime in the format 'YYYY-MM-DD' or 'YYYY-MM-DD HH:MM:SS'

**--sortby** <sortby>

Column to sort by [default: hostname]

## 6.23 Reverse Whois Commands

### 6.23.1 rwhois edit

Edit the RWhois data on the account.

```
rwhois edit [OPTIONS]
```

## Options

- abuse** <abuse>  
Set the abuse email address
- address1** <address1>  
Update the address 1 field
- address2** <address2>  
Update the address 2 field
- city** <city>  
Set the city name
- company** <company>  
Set the company name
- country** <country>  
Set the two-letter country code
- firstname** <firstname>  
Update the first name field
- lastname** <lastname>  
Update the last name field
- postal** <postal>  
Set the postal code field
- public, --private**  
Flags the address as a public or private residence.
- state** <state>  
Set the two-letter state code

### 6.23.2 rwhois show

Display the RWhois information for your account.

```
rwhois show [OPTIONS]
```

## 6.24 Security Groups

### 6.24.1 securitygroup list

List security groups.

```
securitygroup list [OPTIONS]
```

## Options

**--sortby** <sortby>  
Column to sort by

**Options** id|name|description

**-l, --limit** <limit>  
How many results to get in one api call, default is 100 [default: 100]

## 6.24.2 securitygroup detail

Get details about a security group.

```
securitygroup detail [OPTIONS] IDENTIFIER
```

## Arguments

**IDENTIFIER**  
Required argument

## 6.24.3 securitygroup create

Create a security group.

```
securitygroup create [OPTIONS]
```

## Options

**-n, --name** <name>  
The name of the security group

**-d, --description** <description>  
The description of the security group

## 6.24.4 securitygroup edit

Edit details of a security group.

```
securitygroup edit [OPTIONS] GROUP_ID
```

## Options

**-n, --name** <name>  
The name of the security group

**-d, --description** <description>  
The description of the security group

## Arguments

**GROUP\_ID**

Required argument

### 6.24.5 securitygroup delete

Deletes the given security group

```
securitygroup delete [OPTIONS] SECURITYGROUP_ID
```

## Arguments

**SECURITYGROUP\_ID**

Required argument

### 6.24.6 securitygroup rule-list

List security group rules.

```
securitygroup rule-list [OPTIONS] SECURITYGROUP_ID
```

## Options

**--sortby** <sortby>

Column to sort by

**Options** id|remoteIp|remoteGroupId|direction|etherType|portRangeMin|portRangeMax|protocol|createDate|modifyDate

## Arguments

**SECURITYGROUP\_ID**

Required argument

### 6.24.7 securitygroup rule-add

Add a security group rule to a security group.

Examples:

# Add an SSH rule (TCP port 22) to a security group

slcli sg rule-add 384727

-direction ingress

-protocol tcp

-port-min 22

-port-max 22

# Add a ping rule (ICMP type 8 code 0) to a security group

```
slcli sg rule-add 384727
    -direction ingress
    -protocol icmp
    -port-min 8
    -port-max 0
```

```
securitygroup rule-add [OPTIONS] SECURITYGROUP_ID
```

### Options

- r, --remote-ip** <remote\_ip>  
The remote IP/CIDR to enforce
- s, --remote-group** <remote\_group>  
The ID of the remote security group to enforce
- d, --direction** <direction>  
The direction of traffic to enforce (ingress, egress)
- e, --ethertype** <ethertype>  
The ethertype (IPv4 or IPv6) to enforce
- M, --port-max** <port\_max>  
The upper port bound to enforce. When the protocol is ICMP, this specifies the ICMP code to permit
- m, --port-min** <port\_min>  
The lower port bound to enforce. When the protocol is ICMP, this specifies the ICMP type to permit
- p, --protocol** <protocol>  
The protocol (icmp, tcp, udp) to enforce

### Arguments

**SECURITYGROUP\_ID**  
Required argument

## 6.24.8 securitygroup rule-edit

Edit a security group rule in a security group.

```
securitygroup rule-edit [OPTIONS] SECURITYGROUP_ID RULE_ID
```

### Options

- r, --remote-ip** <remote\_ip>  
The remote IP/CIDR to enforce
- s, --remote-group** <remote\_group>  
The ID of the remote security group to enforce
- d, --direction** <direction>  
The direction of traffic to enforce



- e, --ethertype** <ethertype>  
The ethertype (IPv4 or IPv6) to enforce
- M, --port-max** <port\_max>  
The upper port bound to enforce
- m, --port-min** <port\_min>  
The lower port bound to enforce
- p, --protocol** <protocol>  
The protocol (icmp, tcp, udp) to enforce

### Arguments

**SECURITYGROUP\_ID**  
Required argument

**RULE\_ID**  
Required argument

## 6.24.9 securitygroup rule-remove

Remove a rule from a security group.

```
securitygroup rule-remove [OPTIONS] SECURITYGROUP_ID RULE_ID
```

### Arguments

**SECURITYGROUP\_ID**  
Required argument

**RULE\_ID**  
Required argument

## 6.24.10 securitygroup interface-list

List interfaces associated with security groups.

```
securitygroup interface-list [OPTIONS] SECURITYGROUP_ID
```

### Options

**--sortby** <sortby>  
Column to sort by

**Options** networkComponentId|virtualServerId|hostname|interface|ipAddress

### Arguments

**SECURITYGROUP\_ID**  
Required argument

### 6.24.11 securitygroup interface-add

Attach an interface to a security group.

```
securitygroup interface-add [OPTIONS] SECURITYGROUP_ID
```

#### Options

- n, --network-component** <network\_component>  
The network component to associate with the security group
- s, --server** <server>  
The server ID to associate with the security group
- i, --interface** <interface>  
The interface of the server to associate (public/private)

#### Arguments

**SECURITYGROUP\_ID**  
Required argument

### 6.24.12 securitygroup interface-remove

Detach an interface from a security group.

```
securitygroup interface-remove [OPTIONS] SECURITYGROUP_ID
```

#### Options

- n, --network-component** <network\_component>  
The network component to remove from with the security group
- s, --server** <server>  
The server ID to remove from the security group
- i, --interface** <interface>  
The interface of the server to remove (public/private)

#### Arguments

**SECURITYGROUP\_ID**  
Required argument

### 6.24.13 securitygroup event-log

Search for event logs by request id

```
securitygroup event-log [OPTIONS] REQUEST_ID
```

## Arguments

**REQUEST\_ID**

Required argument

## 6.25 SSH Keys

### 6.25.1 sshkey add

Add a new SSH key.

```
sshkey add [OPTIONS] LABEL
```

#### Options

**-f, --in-file** <in\_file>  
The id\_rsa.pub file to import for this key

**-k, --key** <key>  
The actual SSH key

**--note** <note>  
Extra note that will be associated with key

#### Arguments

**LABEL**

Required argument

### 6.25.2 sshkey remove

Permanently removes an SSH key.

```
sshkey remove [OPTIONS] IDENTIFIER
```

#### Arguments

**IDENTIFIER**

Required argument

### 6.25.3 sshkey edit

Edits an SSH key.

```
sshkey edit [OPTIONS] IDENTIFIER
```

## Options

**-k, --label** <label>  
The new label for the key

**--note** <note>  
New notes for the key

## Arguments

**IDENTIFIER**  
Required argument

### 6.25.4 sshkey list

List SSH keys.

```
sshkey list [OPTIONS]
```

## Options

**--sortby** <sortby>  
Column to sort by

**Options** idllabelfingerprintnotes

### 6.25.5 sshkey print

Prints out an SSH key to the screen.

```
sshkey print [OPTIONS] IDENTIFIER
```

## Options

**-f, --out-file** <out\_file>  
The public SSH key will be written to this file

## Arguments

**IDENTIFIER**  
Required argument

## 6.26 SSL Certificates

### 6.26.1 ssl add

Add and upload SSL certificate details.

```
ssl add [OPTIONS]
```

### Options

**--crt** <cert>  
Certificate file

**--csr** <csr>  
Certificate Signing Request file

**--icc** <icc>  
Intermediate Certificate file

**--key** <key>  
Private Key file

**--notes** <notes>  
Additional notes

### 6.26.2 ssl download

Download SSL certificate and key file.

```
ssl download [OPTIONS] IDENTIFIER
```

### Arguments

**IDENTIFIER**  
Required argument

### 6.26.3 ssl edit

Edit SSL certificate.

```
ssl edit [OPTIONS] IDENTIFIER
```

### Options

**--crt** <cert>  
Certificate file

**--csr** <csr>  
Certificate Signing Request file

**--icc** <icc>  
Intermediate Certificate file

**--key** <key>  
Private Key file

**--notes** <notes>  
Additional notes

## Arguments

### **IDENTIFIER**

Required argument

## 6.26.4 ssl list

List SSL certificates.

```
ssl list [OPTIONS]
```

## Options

**--status** <status>

Show certificates with this status [default: all]

**Options** allvalidexpired

**--sortby** <sortby>

Column to sort by

**Options** idlcommon\_namadays\_until\_expirenotes

## 6.26.5 ssl remove

Remove SSL certificate.

```
ssl remove [OPTIONS] IDENTIFIER
```

## Arguments

### **IDENTIFIER**

Required argument

## 6.27 Subnets

### 6.27.1 subnet cancel

Cancel a subnet.

```
subnet cancel [OPTIONS] IDENTIFIER
```

## Arguments

### **IDENTIFIER**

Required argument

## 6.27.2 subnet create

Add a new subnet to your account. Valid quantities vary by type.

### IPv4

static - 1, 2, 4, 8, 16, 32, 64, 128, 256

public - 4, 8, 16, 32, 64, 128, 256

private - 4, 8, 16, 32, 64, 128, 256

### IPv6

static - 64

public - 64

### endpoint-id

static - Network\_Subnet\_IpAddress identifier.

public - Network\_Vlan identifier

private - Network\_Vlan identifier

```
subnet create [OPTIONS] [static|public|private] QUANTITY ENDPOINT_ID
```

## Options

**--ipv6, --v6**

Order IPv6 Addresses

**--test**

Do not order the subnet; just get a quote

## Arguments

### NETWORK

Required argument

### QUANTITY

Required argument

### ENDPOINT\_ID

Required argument

## 6.27.3 subnet detail

Get subnet details.

```
subnet detail [OPTIONS] IDENTIFIER
```

## Options

- no-vs**  
Hide virtual server listing
- no-hardware**  
Hide hardware listing

## Arguments

- IDENTIFIER**  
Required argument

### 6.27.4 subnet list

List subnets.

```
subnet list [OPTIONS]
```

## Options

- sortby** <sortby>  
Column to sort by  
**Options** id|identifier|type|network\_space|datacenter|vlan\_id|IPs|hardware|vs
- d, --datacenter** <datacenter>  
Filter by datacenter shortname (sng01, dal05, ...)
- identifier** <identifier>  
Filter by network identifier
- t, --subnet-type** <subnet\_type>  
Filter by subnet type
- network-space** <network\_space>  
Filter by network space
- ipv4, --v4**  
Display only IPv4 subnets
- ipv6, --v6**  
Display only IPv6 subnets

### 6.27.5 subnet lookup

Find an IP address and display its subnet and device info.

```
subnet lookup [OPTIONS] IP_ADDRESS
```

## Arguments

- IP\_ADDRESS**  
Required argument



## 6.28 Support Tickets

### 6.28.1 ticket create

Create a support ticket.

```
ticket create [OPTIONS]
```

#### Options

**--title** <title>

The title of the ticket [required]

**--subject-id** <subject\_id>

The subject id to use for the ticket, issue 'slcli ticket subjects' to get the list [required]

**--body** <body>

The ticket body

**--hardware** <hardware\_identifier>

The identifier for hardware to attach

**--virtual** <virtual\_identifier>

The identifier for a virtual server to attach

**--priority** <priority>

Ticket priority, from 1 (Critical) to 4 (Minimal Impact). Only settable with Advanced and Premium support. See <https://www.ibm.com/cloud/support>

**Options** 1|2|3|4

### 6.28.2 ticket detail

Get details for a ticket.

```
ticket detail [OPTIONS] IDENTIFIER
```

#### Options

**--count** <count>

Number of updates [default: 10]

#### Arguments

**IDENTIFIER**

Required argument

### 6.28.3 ticket list

List tickets.

```
ticket list [OPTIONS]
```

### Options

**--open, --closed**

Display only open or closed tickets

## 6.28.4 ticket update

Adds an update to an existing ticket.

```
ticket update [OPTIONS] IDENTIFIER
```

### Options

**--body** <body>

The entry that will be appended to the ticket

### Arguments

**IDENTIFIER**

Required argument

## 6.28.5 ticket upload

Adds an attachment to an existing ticket.

```
ticket upload [OPTIONS] IDENTIFIER
```

### Options

**--path** <path>

The path of the attachment to be uploaded

**--name** <name>

The name of the attachment shown in the ticket

### Arguments

**IDENTIFIER**

Required argument

## 6.28.6 ticket subjects

List Subject IDs for ticket creation.

```
ticket subjects [OPTIONS]
```

### 6.28.7 ticket summary

Summary info about tickets.

```
ticket summary [OPTIONS]
```

### 6.28.8 ticket attach

Attach devices to a ticket.

```
ticket attach [OPTIONS] IDENTIFIER
```

#### Options

**--hardware** <hardware\_identifier>

The identifier for hardware to attach

**--virtual** <virtual\_identifier>

The identifier for a virtual server to attach

#### Arguments

**IDENTIFIER**

Required argument

### 6.28.9 ticket detach

Detach devices from a ticket.

```
ticket detach [OPTIONS] IDENTIFIER
```

#### Options

**--hardware** <hardware\_identifier>

The identifier for hardware to detach

**--virtual** <virtual\_identifier>

The identifier for a virtual server to detach

#### Arguments

**IDENTIFIER**

Required argument

## 6.29 Users

Version 5.6.0 introduces the ability to interact with user accounts from the cli.

### 6.29.1 user list

List Users.

```
user list [OPTIONS]
```

#### Options

**--columns** <columns>

Columns to display. [options: id, username, email, displayName, status, hardwareCount, virtualGuestCount]  
[default: id,username,email,displayName]

### 6.29.2 user detail

User details.

```
user detail [OPTIONS] IDENTIFIER
```

#### Options

**-k, --keys**

Show the users API key.

**-p, --permissions**

Display permissions assigned to this user. Master users will show no permissions

**-h, --hardware**

Display hardware this user has access to.

**-v, --virtual**

Display virtual guests this user has access to.

**-l, --logins**

Show login history of this user for the last 30 days

**-e, --events**

Show event log for this user.

#### Arguments

**IDENTIFIER**

Required argument

### 6.29.3 user permissions

User Permissions. TODO change to list all permissions, and which users have them

```
user permissions [OPTIONS] IDENTIFIER
```

#### Arguments

##### IDENTIFIER

Required argument

### 6.29.4 user edit-permissions

Enable or Disable specific permissions.

```
user edit-permissions [OPTIONS] IDENTIFIER
```

#### Options

##### --enable, --disable

Enable (DEFAULT) or Disable selected permissions

##### -p, --permission <permission>

Permission keyName to set, multiple instances allowed. Use keyword ALL to select ALL permissions

##### -u, --from-user <from\_user>

Set permissions to match this user's permissions. Will add then remove the appropriate permissions

#### Arguments

##### IDENTIFIER

Required argument

### 6.29.5 user edit-details

Edit a Users details

JSON strings should be enclosed in " and each item should be enclosed in ""

Example:

```
slcli user edit-details testUser -t '{"firstName": "Test", "lastName": "Testerson"}
```

```
user edit-details [OPTIONS] USER
```

#### Options

##### -t, --template <template>

A json string describing [https://softlayer.github.io/reference/datatypes/SoftLayer\\_User\\_Customer/](https://softlayer.github.io/reference/datatypes/SoftLayer_User_Customer/) [required]

## Arguments

### USER

Required argument

## 6.29.6 user create

Creates a user Users.

Remember to set the permissions and access for this new user.

Example:

```
slcli user create my@email.com -e my@email.com -p generate -a
-t '{"firstName": "Test", "lastName": "Testerson"}
```

```
user create [OPTIONS] USERNAME
```

## Options

**-e, --email** <email>

Email address for this user. Required for creation. [required]

**-p, --password** <password>

Password to set for this user. If no password is provided, user will be sent an email to generate one, which expires in 24 hours. '-p generate' will create a password for you (Requires Python 3.6+). Passwords require 8+ characters, upper and lowercase, a number and a symbol.

**-u, --from-user** <from\_user>

Base user to use as a template for creating this user. Will default to the user running this command. Information provided in -template supersedes this template.

**-t, --template** <template>

A json string describing [https://softlayer.github.io/reference/datatypes/SoftLayer\\_User\\_Customer/](https://softlayer.github.io/reference/datatypes/SoftLayer_User_Customer/)

**-a, --api-key**

Create an API key for this user.

## Arguments

### USERNAME

Required argument

## 6.29.7 user delete

Sets a user's status to CANCEL\_PENDING, which will immediately disable the account, and will eventually be fully removed from the account by an automated internal process.

Example: slcli user delete userId

```
user delete [OPTIONS] IDENTIFIER
```

## Arguments

**IDENTIFIER**

Required argument

## 6.30 VLANs

### 6.30.1 vlan detail

Get details about a VLAN.

```
vlan detail [OPTIONS] IDENTIFIER
```

## Options

**--no-vs**

Hide virtual server listing

**--no-hardware**

Hide hardware listing

## Arguments

**IDENTIFIER**

Required argument

### 6.30.2 vlan list

List VLANs.

```
vlan list [OPTIONS]
```

## Options

**--sortby** <sortby>

Column to sort by

**Options** id|number|name|firewall|datacenter|hardware|virtual\_servers|public\_ips**-d, --datacenter** <datacenter>

Filter by datacenter shortname (sng01, dal05, ...)

**-n, --number** <number>

Filter by VLAN number

**--name** <name>

Filter by VLAN name

**-l, --limit** <limit>

How many results to get in one api call, default is 100 [default: 100]

## 6.31 Working with Virtual Servers

Using the SoftLayer portal to order virtual servers is fine, but for a number of reasons it's often more convenient to use the command line. For this, you can use SoftLayer's command-line client to make administrative tasks quicker and easier. This page gives an intro to working with SoftLayer virtual servers using SoftLayer's command-line client.

---

**Note:** The following assumes that the client is already *configured with valid SoftLayer credentials*.

---

First, let's list the current virtual servers with `slcli vs list`.

```
$ slcli vs list
:.....:
↪.....:
: id : datacenter :      host      : cores : memory : primary_ip : ↵
↪backend_ip : active_transaction : owner  :
:.....:
↪.....:
:.....:
↪.....:
```

We don't have any virtual servers yet! Let's fix that. Before we can create a virtual server (VS), we need to know what options are available to us: RAM, CPU, operating systems, disk sizes, disk types, datacenters, and so on. Luckily, there's a simple command to show all options: `slcli vs create-options`.

*Some values were ommitted for brevity*

```
$ slcli vs create-options
:.....:
↪.....:
:          name : value                               ↵
↪          :
:.....:
↪.....:
:          datacenter : ams01                               ↵
↪          :
:          : ams03                               ↵
↪          :
:          : wdc07                               ↵
↪          :
:          flavors (balanced) : B1_1X2X25                           ↵
↪          :
:          : B1_1X2X25                           ↵
↪          :
:          : B1_1X2X100                           ↵
↪          :
:          cpus (standard) : 1,2,4,8,12,16,32,56                 ↵
↪          :
:          cpus (dedicated) : 1,2,4,8,16,32,56                   ↵
↪          :
:          cpus (dedicated host) : 1,2,4,8,12,16,32,56               ↵
↪          :
:          memory : 1024,2048,4096,6144,8192,12288,16384,32768,49152,
↪65536,131072,247808                 :
:          memory (dedicated host) : 1024,2048,4096,6144,8192,12288,16384,32768,49152,
↪65536,131072,247808                 :
:          os (CENTOS) : CENTOS_5_64                         ↵
↪          :
```

(continues on next page)



(continued from previous page)

```

:                                : CENTOS_LATEST_64
↪                                :
:      os (CLOUDLINUX) : CLOUDLINUX_5_64
↪                                :
:                                : CLOUDLINUX_6_64
↪                                :
:                                : CLOUDLINUX_LATEST
↪                                :
:                                : CLOUDLINUX_LATEST_64
↪                                :
:      os (COREOS) : COREOS_CURRENT_64
↪                                :
:                                : COREOS_LATEST
↪                                :
:                                : COREOS_LATEST_64
↪                                :
:      os (DEBIAN) : DEBIAN_6_64
↪                                :
:                                : DEBIAN_LATEST_64
↪                                :
:      os (OTHERUNIXLINUX) : OTHERUNIXLINUX_1_64
↪                                :
:                                : OTHERUNIXLINUX_LATEST
↪                                :
:                                : OTHERUNIXLINUX_LATEST_64
↪                                :
:      os (REDHAT) : REDHAT_5_64
↪                                :
:                                : REDHAT_6_64
↪                                :
:                                : REDHAT_7_64
↪                                :
:                                : REDHAT_LATEST
↪                                :
:                                : REDHAT_LATEST_64
↪                                :
:      san disk(0) : 25,100
↪                                :
:      san disk(2) : 10,20,25,30,40,50,75,100,125,150,175,200,250,300,
↪350,400,500,750,1000,1500,2000 :
:      local disk(0) : 25,100
↪                                :
:      local disk(2) : 25,100,150,200,300
↪                                :
: local (dedicated host) disk(0) : 25,100
↪                                :
:      nic (dedicated host) : 100,1000
↪                                :
:.....:.....
↪.....:.....

```

Here's the command to create a 2-core virtual server with 1GiB memory, running Ubuntu 14.04 LTS, and that is billed on an hourly basis in the San Jose 1 datacenter using the command `slcli vs create`.

```

$ slcli vs create --hostname=example --domain=softlayer.com -f B1_1X2X25 -o DEBIAN_
↪LATEST_64 --datacenter=ams01 --billing=hourly

```

(continues on next page)

(continued from previous page)

```

This action will incur charges on your account. Continue? [y/N]: y
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
↪.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
: ID : FQDN : guid
↪ : Order Date :
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
↪.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
: 70112999 : testtesttest.test.com : 1abc7afb-9618-4835-89c9-586f3711d8ea : 2019-
↪01-30T17:16:58-06:00 :
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
↪.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
: OrderId: 12345678
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
: Cost : Description
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
: 0.0 : Debian GNU/Linux 9.x Stretch/Stable - Minimal Install (64 bit) :
: 0.0 : 25 GB (SAN)
: 0.0 : Reboot / Remote Console
: 0.0 : 100 Mbps Public & Private Network Uplinks
: 0.0 : 0 GB Bandwidth Allotment
: 0.0 : 1 IP Address
: 0.0 : Host Ping and TCP Service Monitoring
: 0.0 : Email and Ticket
: 0.0 : Automated Reboot from Monitoring
: 0.0 : Unlimited SSL VPN Users & 1 PPTP VPN User per account
: 0.0 : Nessus Vulnerability Assessment & Reporting
: 0.0 : 2 GB
: 0.0 : 1 x 2.0 GHz or higher Core
: 0.000 : Total hourly cost
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....

```

After the last command, the virtual server is now being built. It should instantly appear in your virtual server list now.

```

$ slcli vs list
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
↪.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
: id : datacenter : host : cores : memory : primary_ip :
↪backend_ip : active_transaction :
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
↪.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
: 1234567 : ams01 : example.softlayer.com : 2 : 1G : 108.168.200.11 : 10.
↪54.80.200 : Assign Host :
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....
↪.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....

```

Cool. You may ask, “It’s creating... but how do I know when it’s done?” Well, here’s how:

```

$ slcli vs ready 'example' --wait=600
READY

```

When the previous command returns, you’ll know that the virtual server has finished the provisioning process and is ready to use. This is *very* useful for chaining commands together.

Now that you have your virtual server, let’s get access to it. To do that, use the `slcli vs detail` command. From the example below, you can see that the username is ‘root’ and password is ‘ABCDEFGH’.

**Warning:** Be careful when using the `--passwords` flag. This will print the virtual server's password on the screen. Make sure no one is looking over your shoulder. It's also advisable to change your root password soon after creating your virtual server, or to create a user with sudo access and disable SSH-based login directly to the root account.

```
$ slcli vs detail example --passwords
:.....:.....:
:      Name : Value      :
:.....:.....:
:      id   : 1234567      :
:  hostname : example.softlayer.com :
:      state : Active              :
:      state : Running             :
:  datacenter : ams01                :
:      cores : 2                    :
:      memory : 1G                      :
:  public_ip : 108.168.200.11       :
:  private_ip : 10.54.80.200         :
:      os    : Debian                    :
: private_only : False                  :
: private_cpu : False                  :
:      created : 2013-06-13T08:29:44-06:00 :
:      modified : 2013-06-13T08:31:57-06:00 :
:      users  : root ABCDEFGH      :
:.....:.....:
```

### 6.31.1 vs bandwidth

Bandwidth data over date range. Bandwidth is listed in GB

Using just a date might get you times off by 1 hour, use T00:01 to get just the specific days data Timezones can also be included with the YYYY-MM-DDTHH:mm:ss.000000-HH:mm format.

Due to some rounding and date alignment details, results here might be slightly different than results in the control portal.

Example:

```
slcli hw bandwidth 1234 -s 2019-05-01T00:01 -e 2019-05-02T00:00:01.000000-12:00
```

```
vs bandwidth [OPTIONS] IDENTIFIER
```

#### Options

- s, --start\_date** <start\_date>  
Start Date YYYY-MM-DD, YYYY-MM-DDTHH:mm:ss, [required]
- e, --end\_date** <end\_date>  
End Date YYYY-MM-DD, YYYY-MM-DDTHH:mm:ss [required]
- p, --summary\_period** <summary\_period>  
300, 600, 1800, 3600, 43200 or 86400 seconds [default: 3600]
- q, --quite\_summary**  
Only show the summary table [default: False]

## Arguments

### **IDENTIFIER**

Required argument

If no timezone is specified, IMS local time (CST) will be assumed, which might not match your user's selected timezone.

## 6.31.2 vs cancel

Cancel virtual servers.

```
vs cancel [OPTIONS] IDENTIFIER
```

## Arguments

### **IDENTIFIER**

Required argument

## 6.31.3 vs capture

Capture one or all disks from a virtual server to a SoftLayer image.

```
vs capture [OPTIONS] IDENTIFIER
```

## Options

**-n, --name** <name>

Name of the image [required]

**--all** <all>

Capture all disks belonging to the VS

**--note** <note>

Add a note to be associated with the image

## Arguments

### **IDENTIFIER**

Required argument

## 6.31.4 vs create

Order/create virtual servers.

```
vs create [OPTIONS]
```

## Options

- H, --hostname** <hostname>  
Host portion of the FQDN [required]
- D, --domain** <domain>  
Domain portion of the FQDN [required]
- c, --cpu** <cpu>  
Number of CPU cores (not available with flavors)
- m, --memory** <memory>  
Memory in mebibytes (not available with flavors)
- f, --flavor** <flavor>  
Public Virtual Server flavor key name
- d, --datacenter** <datacenter>  
Datacenter shortname [required]
- o, --os** <os>  
OS install code. Tip: you can specify <OS>\_LATEST
- image** <image>  
Image ID. See: 'slcli image list' for reference
- boot-mode** <boot\_mode>  
Specify the mode to boot the OS in. Supported modes are HVM and PV.
- billing** <billing>  
Billing rate [default: hourly]  
**Options** hourly|monthly
- dedicated, --public**  
Create a Dedicated Virtual Server
- host-id** <host\_id>  
Host Id to provision a Dedicated Host Virtual Server onto
- san**  
Use SAN storage instead of local disk.
- test**  
Do not actually create the virtual server
- export** <export>  
Exports options to a template file
- i, --postinstall** <postinstall>  
Post-install script to download
- k, --key** <key>  
SSH keys to add to the root user (multiple occurrence permitted)
- disk** <disk>  
Disk sizes (multiple occurrence permitted)
- private**  
Forces the VS to only have access the private network
- like** <like>  
Use the configuration from an existing VS

- n, --network** <network>  
Network port speed in Mbps
- g, --tag** <tag>  
Tags to add to the instance (multiple occurrence permitted)
- t, --template** <template>  
A template file that defaults the command-line options
- u, --userdata** <userdata>  
User defined metadata string
- F, --userfile** <userfile>  
Read userdata from file
- vlan-public** <vlan\_public>  
The ID of the public VLAN on which you want the virtual server placed
- vlan-private** <vlan\_private>  
The ID of the private VLAN on which you want the virtual server placed
- subnet-public** <subnet\_public>  
The ID of the public SUBNET on which you want the virtual server placed
- subnet-private** <subnet\_private>  
The ID of the private SUBNET on which you want the virtual server placed
- S, --public-security-group** <public\_security\_group>  
Security group ID to associate with the public interface (multiple occurrence permitted)
- s, --private-security-group** <private\_security\_group>  
Security group ID to associate with the private interface (multiple occurrence permitted)
- wait** <wait>  
Wait until VS is finished provisioning for up to X seconds before returning
- placementgroup** <placementgroup>  
Placement Group name or Id to order this guest on. See: slcli vs placementgroup list
- ipv6**  
Adds an IPv6 address to this guest
- transient**  
Create a transient virtual server

### 6.31.5 vs create-options

Virtual server order options.

```
vs create-options [OPTIONS]
```

### 6.31.6 vs dns-sync

Sync DNS records.

```
vs dns-sync [OPTIONS] IDENTIFIER
```

## Options

- a, --a-record**  
Sync the A record for the host
- aaaa-record**  
Sync the AAAA record for the host
- ptr**  
Sync the PTR record for the host
- ttl <ttl>**  
Sets the TTL for the A and/or PTR records [default: 7200]

## Arguments

- IDENTIFIER**  
Required argument

### 6.31.7 vs edit

Edit a virtual server's details.

```
vs edit [OPTIONS] IDENTIFIER
```

## Options

- D, --domain <domain>**  
Domain portion of the FQDN
- H, --hostname <hostname>**  
Host portion of the FQDN. example: server
- g, --tag <tag>**  
Tags to set or empty string to remove all
- u, --userdata <userdata>**  
User defined metadata string
- F, --userfile <userfile>**  
Read userdata from file
- public-speed <public\_speed>**  
Public port speed.  
**Options** 0|10|100|1000|10000
- private-speed <private\_speed>**  
Private port speed.  
**Options** 0|10|100|1000|10000

## Arguments

- IDENTIFIER**  
Required argument

### 6.31.8 vs list

List virtual servers.

```
vs list [OPTIONS]
```

#### Options

- c, --cpu** <cpu>  
Number of CPU cores
- D, --domain** <domain>  
Domain portion of the FQDN
- d, --datacenter** <datacenter>  
Datacenter shortname
- H, --hostname** <hostname>  
Host portion of the FQDN
- m, --memory** <memory>  
Memory in mebibytes
- n, --network** <network>  
Network port speed in Mbps
- hourly**  
Show only hourly instances
- monthly**  
Show only monthly instances
- transient** <transient>  
Filter by transient instances
- tag** <tag>  
Filter by tags (multiple occurrence permitted)
- sortby** <sortby>  
Column to sort by [default: hostname]
- columns** <columns>  
Columns to display. [options: guid, primary\_ip, backend\_ip, datacenter, action, power\_state, created\_by, tags]  
[default: id,hostname,primary\_ip,backend\_ip,datacenter,action]
- l, --limit** <limit>  
How many results to get in one api call, default is 100 [default: 100]

### 6.31.9 vs pause

Pauses an active virtual server.

```
vs pause [OPTIONS] IDENTIFIER
```



## Arguments

**IDENTIFIER**

Required argument

### 6.31.10 vs power-on

Power on a virtual server.

```
vs power-on [OPTIONS] IDENTIFIER
```

## Arguments

**IDENTIFIER**

Required argument

### 6.31.11 vs power-off

Power off an active virtual server.

```
vs power-off [OPTIONS] IDENTIFIER
```

## Options

**--hard, --soft**

Perform a hard shutdown

## Arguments

**IDENTIFIER**

Required argument

### 6.31.12 vs resume

Resumes a paused virtual server.

```
vs resume [OPTIONS] IDENTIFIER
```

## Arguments

**IDENTIFIER**

Required argument

### 6.31.13 vs rescue

Reboot into a rescue image.

```
vs rescue [OPTIONS] IDENTIFIER
```

#### Arguments

**IDENTIFIER**

Required argument

### 6.31.14 vs reboot

Reboot an active virtual server.

```
vs reboot [OPTIONS] IDENTIFIER
```

#### Options

**--hard, --soft**

Perform a hard or soft reboot

#### Arguments

**IDENTIFIER**

Required argument

### 6.31.15 vs ready

Check if a virtual server is ready.

```
vs ready [OPTIONS] IDENTIFIER
```

#### Options

**--wait** <wait>

Seconds to wait [default: 0]

#### Arguments

**IDENTIFIER**

Required argument

### 6.31.16 vs upgrade

Upgrade a virtual server.

```
vs upgrade [OPTIONS] IDENTIFIER
```

#### Options

- cpu** <cpu>  
Number of CPU cores
- private**  
CPU core will be on a dedicated host server.
- memory** <memory>  
Memory in megabytes
- network** <network>  
Network port speed in Mbps
- flavor** <flavor>  
Flavor keyName Do not use `-memory`, `-cpu` or `-private`, if you are using flavors

#### Arguments

**IDENTIFIER**  
Required argument

### 6.31.17 vs usage

Usage information of a virtual server.

```
vs usage [OPTIONS] IDENTIFIER
```

#### Options

- s, --start\_date** <start\_date>  
Start Date e.g. 2019-3-4 (yyyy-MM-dd) [required]
- e, --end\_date** <end\_date>  
End Date e.g. 2019-4-2 (yyyy-MM-dd) [required]
- t, --valid\_type** <valid\_type>  
Metric\_Data\_Type keyName e.g. CPU0, CPU1, MEMORY\_USAGE, etc. [required]
- p, --summary\_period** <summary\_period>  
300, 600, 1800, 3600, 43200 or 86400 seconds

#### Arguments

**IDENTIFIER**  
Required argument

## 6.31.18 Reserved Capacity

### Working with Reserved Capacity

There are two main concepts for Reserved Capacity. The [Reserved Capacity Group](#) and the [Reserved Capacity Instance](#). The Reserved Capacity Group, is a set block of capacity set aside for you at the time of the order. It will contain a set number of Instances which are all the same size. Instances can be ordered like normal VSIs, with the exception that you need to include the reservedCapacityGroupId, and it must be the same size as the group you are ordering the instance in.

- [About Reserved Capacity](#)
- [Reserved Capacity FAQ](#)

The SLCLI supports some basic Reserved Capacity Features.

#### vs capacity create

This command will create a Reserved Capacity Group.

**Warning: These groups can not be canceled until their contract expires in 1 or 3 years!**

```
$ slcli vs capacity create --name test-capacity -d dal13 -b 1411193 -c B1_1X2_1_YEAR_
↪TERM -q 10
```

#### vs cacpacity create\_options

This command will print out the Flavors that can be used to create a Reserved Capacity Group, as well as the backend routers available, as those are needed when creating a new group.

#### vs capacity create\_guest

This command will create a virtual server (Reserved Capacity Instance) inside of your Reserved Capacity Group. This command works very similar to the *slcli vs create* command.

```
$ slcli vs capacity create-guest --capacity-id 1234 --primary-disk 25 -H ABCD -D test.
↪com -o UBUNTU_LATEST_64 --ipv6 -k test-key --test
```

#### vs capacity detail

This command will print out some basic information about the specified Reserved Capacity Group.

#### vs capacity list

This command will list out all Reserved Capacity Groups. a # symbol represents a filled instance, and a - symbol represents an empty instance

```

$ slcli vs capacity list
:.....
:
:                               Reserved Capacity
:
:                               :
:.....:.....:.....:.....:.....:.....:.....
:.....:.....:.....:.....:.....:.....:.....
: ID :      Name      : Capacity :      Flavor      : Location :
:-----:-----:-----:-----:-----:
: 1234 : test-capacity   : #####----- : B1.1x2 (1 Year Term) : bcr02a.dal13 :
: 2018-09-24T16:33:09-06:00 :
:.....:.....:.....:.....:.....:.....:.....
:.....:.....:.....:.....:.....:.....:.....

```

## 6.31.19 Placement Groups

### Working with Placement Groups

A **Placement Group** is a way to control which physical servers your virtual servers get provisioned onto.

To create a `Virtual_PlacementGroup` object, you will need to know the following:

- `backendRouterId`, from `getAvailableRouters`
- `ruleId`, from `getAllObjects`
- `name`, can be any string, but must be unique on your account

Once a placement group is created, you can create new virtual servers in that group. Existing VSIs cannot be moved into a placement group. When ordering a VSI in a placement group, make sure to set the `placementGroupId` for each guest in your order.

use the `--placementgroup` option with `vs create` to specify creating a VSI in a specific group.

```

$ slcli vs create -H testGroup001 -D test.com -f B1_1X2X25 -d mex01 -o DEBIAN_
↳LATEST --placementgroup testGroup

```

Placement groups can only be deleted once all the virtual guests in the group have been reclaimed.

### vs placementgroup create

This command will create a placement group.

```

$ slcli vs placementgroup create --name testGroup -b bcr02a.dal06 -r SPREAD

```

### Options

- **--name TEXT** Name for this new placement group. [required]
- **-b, --backend\_router TEXT** backendRouter, can be either the hostname or id. [required]
- **-r, --rule TEXT** The keyName or Id of the rule to govern this placement group. [required]

## vs placementgroup create-options

This command will print out the available routers and rule sets for use in creating a placement group.

```

$ slcli vs placementgroup create-options
:.....:
:           Available Routers           :
:.....:.....:.....:
: Datacenter : Hostname : Backend Router Id :
:.....:.....:.....:
: Washington 1 : bcr01.wdc01 : 16358 :
: Tokyo 5 : bcr01a.tok05 : 1587015 :
:.....:.....:.....:
: Rules :
:.....:.....:
: Id : KeyName :
:.....:.....:
: 1 : SPREAD :
:.....:.....:

```

## vs placementgroup delete

This command will remove a placement group. The placement group needs to be empty for this command to succeed.

### Options

**--purge** Delete all guests in this placement group. The group itself can be deleted once all VMs are fully reclaimed

```
$ slcli vs placementgroup delete testGroup
```

You can use the flag `-purge` to auto-cancel all VSIs in a placement group. You will still need to wait for them to be reclaimed before proceeding to delete the group itself.

```

$ slcli vs placementgroup delete testGroup --purge
You are about to delete the following guests!
issues10691547768562.test.com, issues10691547768572.test.com,
↪issues10691547768552.test.com, issues10691548718280.test.com
This action will cancel all guests! Continue? [y/N]: y
Deleting issues10691547768562.test.com...
Deleting issues10691547768572.test.com...
Deleting issues10691547768552.test.com...
Deleting issues10691548718280.test.com...

```

## vs placementgroup list

This command will list all placement groups on your account.

```

$ slcli vs placementgroup list
:.....:
↪.....:

```

(continues on next page)







## 7.1 Contribution Guide

This page explains how to get started contributing code to the SoftLayer API Python Bindings project.

### 7.1.1 Code Organization

- **docs** - Where The source to this documentation lives.
- **SoftLayer** - All the source lives under here.
  - **API** - Primary API client.
  - **CLI** - Code for the command-line interface.
  - **managers** - API Managers. Abstractions to help use the API.

### 7.1.2 Setting Up A Dev Environment

Before working with the SoftLayer Python API client source, we strongly recommend that you know how to use Python's virtual environment, [virtualenv](#). Virtualenv allows you to create isolated Python environments that are individually tailored to particular development projects. Each environment can have its own set of libraries and even its own Python interpreter. This keeps them fully isolated, reducing the possibility of library conflicts between different projects.

After you have virtualenv, you should set up a virtual environment and activate it whenever you are working on softlayer-python. The commands needed to setup an environment and activate it might look something like this:

```
virtualenv --no-site-packages softlayer_env
source softlayer_env/bin/activate
```

Please refer to the virtualenv documentation for more information about creating, and working with virtual environments.

Once you have an appropriate environment, you will then download the SoftLayer API Python Bindings source code by following the *installation instructions*. Change into `softlayer-python` source directory and run the following to install the pre-requisites that you'll need in order to run the test suites:

```
pip install -r tools/test-requirements.txt
```

### 7.1.3 Testing

The project has a mix of functional and unit tests. Before submitting changes to be integrated into the project, you should validate your code using `tox`. Simply issue the `tox` command from the root of the source tree:

```
tox
```

In addition to testing different versions of Python, `tox` checks for common mistakes in the code using `Flake8` and `pylint`. You should eliminate the linting errors that are reported before submitting your code. You can run only the linting checks by using this command:

```
tox -eanalysis
```

The project's configuration instructs `tox` to test against many different versions of Python. A `tox` test will use as many of those as it can find on your local computer. Rather than installing all those versions, we recommend that you point the `Travis` continuous integration tool at your GitHub fork. `Travis` will run the test against the full suite of Python versions every time you push new code.

Using `tox` to run tests in multiple environments can be very time consuming. If you wish to quickly run the tests in your own environment, you may do so using `py.test`. The command to do that is:

```
py.test tests
```

### Fixtures

Testing of this project relies quite heavily on fixtures to simulate API calls. When running the unit tests, we use the `FixtureTransport` class, which instead of making actual API calls, loads data from `/fixtures/SoftLayer_Service_Name.py` and tries to find a variable that matches the method you are calling.

When adding new Fixtures you should try to sanitize the data of any account identifying results, such as account ids, username, and that sort of thing. It is ok to leave the id in place for things like datacenter ids, price ids.

To Overwrite a fixture, you can use a mock object to do so. Like either of these two methods:

```
# From tests/CLI/modules/vs_capacity_tests.py
from SoftLayer.fixtures import SoftLayer_Product_Package

def test_create_test(self):
    item_mock = self.set_mock('SoftLayer_Product_Package', 'getItems')
    item_mock.return_value = SoftLayer_Product_Package.getItems_RESERVED_CAPACITY

def test_detail_pending(self):
    capacity_mock = self.set_mock('SoftLayer_Virtual_ReservedCapacityGroup',
    ↪ 'getObject')
    get_object = {
        'name': 'test-capacity',
        'instances': []
    }
    capacity_mock.return_value = get_object
```

## 7.1.4 Documentation

The project is documented in [reStructuredText](#) and built using [Sphinx](#). If you have [fabric](#) installed, you simply need to run the following to build the docs:

```
fab make_html
```

The documentation will be built in `docs/_build/html`. If you don't have [fabric](#), use the following commands.

```
cd docs
make html
sphinx-build -b html ./ ./html
```

The primary docs are built at [Read the Docs](#).

## 7.1.5 Style

This project tries to follow [PEP 8](#) and most of the style suggestions that [pyflakes](#) recommends. Run [Flake8](#) regularly. [Flake8](#), with project-specific exceptions, can be run by using [tox](#):

```
tox -e analysis
```

[Autopep8](#) can fix a lot of the simple [flake8](#) errors about whitespace and indentation.

```
autopep8 -r -a -v -i --max-line-length 119
```

## 7.1.6 Contributing

Contributing to the Python API bindings follows the [fork-pull-request model](#) on [GitHub](#). The project uses [GitHub's issue tracker](#) and [pull requests](#) to manage source control, bug fixes and new feature development regarding the API bindings and the CLI. In order to contribute, we require that you sign a contributor agreement:

- Sign our contributor agreement (CLA) You can find the [CLA](#) [here](#).
- If you're contributing on behalf of your employer we'll need a signed copy of our corporate contributor agreement (CCLA) as well. You can find the [CCLA](#) [here](#).

## 7.1.7 Developer Resources

## 7.2 Command-Line Interface Developer Guide

The SoftLayer CLI can be used to manage many different SoftLayer services directly from the command line.

The command line parsing is currently based on [click](#), which is a command parsing library along with some additions to dynamically load modules from a routes-like file and from [entry points](#).

### 7.2.1 First Example

For the first example, we can create `slcli table-example` by creating the following file at `SoftLayer/CLI/table_example.py`:

```

"""A formatting table example."""
from SoftLayer.CLI import environment
from SoftLayer.CLI import formatting

import click

@click.command()
@environment.pass_env
def cli(env):
    """This returns an table that highlights how tables are output"""
    # create a table with two columns: col1, col2
    table = formatting.Table(['col1', 'col2'])

    # align the data facing each other
    # valid values are r, c, l for right, center, left
    # note, these are suggestions based on the format chosen by the user
    table.align['col1'] = 'r'
    table.align['col2'] = 'l'

    # add rows
    table.add_row(['test', 'test'])
    table.add_row(['test2', 'test2'])

    env.fout(table)

```

Then we need to register it so that *slcli table-example* will know to route to this new module. We do that by adding `ALL_ROUTES` in `SoftLayer/CLI/routes.py` to include the following:

```

...
('table-example', 'SoftLayer.CLI.table_example:cli'),
...

```

Which gives us

```

$ slcli table-example
:.....:.....:
:  col1 : col2  :
:.....:.....:
:  test : test  :
: test2 : test2 :
:.....:.....:

$ slcli --format=raw table-example
test  test
test2 test2

```

Formatting of the data represented in the table is actually controlled upstream from the `CLIRunnable`'s making supporting more data formats in the future easier.

## 7.2.2 Arguments

A command usually isn't very useful without context or arguments of some kind. With click, you have a large array of argument and option types at your disposal. Additionally, with the SoftLayer CLI, we have global options and context which is stored in `SoftLayer.CLI.environment.Environment` and is attainable through a decorator located at `SoftLayer.CLI.environment.pass_env`. An example of options and the environment is shown below. It also shows how

output should be done using *env.out* instead of printing. This is used for testing and to have a consistent way to print things onto the screen.

```

from SoftLayer.CLI import environment

import click

@click.command()
@click.option("--number",
              required=True,
              type=click.INT,
              help="print different output")
@click.option("--choice",
              type=click.Choice(['this', 'that']),
              help="print different output")
@click.option("--test", help="print different output")
@environment.pass_env
def cli(env, number, choice, test):
    """Argument parsing example"""

    if test:
        env.out("Just testing, move along...")
    else:
        env.out("This is fo'realz!")

    if choice == 'this':
        env.out("Selected this")
    elif choice == 'that':
        env.out("Selected that")

    env.out("This is a number: %d" % number)

```

Refer to the click library documentation for more options.

### 7.2.3 Accessing the API

A *SoftLayer* client is stood up for every command and is available through *SoftLayer.CLI.environment.Environment.client*. The example below shows how to make a simple API call to the *SoftLayer\_Account::getObject*.

```

from SoftLayer.CLI import environment

import click

@click.command()
@environment.pass_env
def cli(env):
    """Using the SoftLayer API client"""

    account = env.client['Account'].getObject()
    return account['companyName']

```

## 7.2.4 Aborting execution

When a confirmation fails, you probably want to stop execution and give a non-zero exit code. To do that, raise a *SoftLayer.CLI.exceptions.CLIAbort* exception with the message for the user as the first parameter. This will prevent any further execution and properly return the right error code.

```
raise CLIAbort("Aborting. Failed confirmation")
```

## 7.2.5 Documenting Commands

All commands should be documented, luckily there is a sphinx module that makes this pretty easy.

If you were adding a summary command to *slcli account* you would find the documentation in *docs/cli/account.rst* and you would just need to add this for your command

```
.. click:: SoftLayer.CLI.account.summary:cli
   :prog: account summary
   :show-nested:
```

The following REGEX can take the route entry and turn it into a document entry.

```
s/^\(['([a-z]*)':([a-z-]*)', '([a-zA-Z\.:_]*')\'),$/.. click:: $3\n   :prog: $1 $2\n   ↪:show-nested:\n/
```

Find:

```
^\(['([a-z]*)':([a-z-]*)', '([a-zA-Z\.:_]*')\'),$
```

REPLACE:

```
.. click:: $3
   :prog: $1 $2
   :show-nested:
```

I tried to get sphinx-click to auto document the ENTIRE slcli, but the results were all on one page, and required a few changes to sphinx-click itself to work. This is due to the fact that most commands in SLCLI use the function name “cli”, and some hacks would have to be put in place to use the path name instead.

## 7.2.6 Architecture

*SLCLI* is the base command, and it starts at *SoftLayerCLICore.py*. Commands are loaded from the *SoftLayerCLIroutes.py* file. How Click figures this out is defined by the *CommandLoader* class in *core.py*, which is an example of a *MultiCommand*.

There are a few examples of commands that are three levels deep, that use a bit more graceful command loader.

- *SoftLayerCLIVirtcapacity\_\_init\_\_.py*
- *SoftLayerCLIVirtplacementgroup\_\_init\_\_.py*
- *SoftLayerCLIobject\_storagecredential\_\_init\_\_.py*

These commands are not directly listed in the routes file, because the autoloader doesn’t have the ability to parse multiple commands like that. For now it was easier to make the rare third level commands have their own special loader than re-write the base command loader to be able to look deeper into the project for commands.

## CHAPTER 8

---

### External Links

---

- [SoftLayer API Documentation](#)
- [Source on GitHub](#)
- [Issues](#)
- [Pull Requests](#)
- [PyPI](#)





**S**

SoftLayer, 71  
SoftLayer.managers.account, 8  
SoftLayer.managers.autoscale, 9  
SoftLayer.managers.block, 10  
SoftLayer.managers.cdn, 16  
SoftLayer.managers.dedicated\_host, 18  
SoftLayer.managers.dns, 20  
SoftLayer.managers.event\_log, 23  
SoftLayer.managers.file, 24  
SoftLayer.managers.firewall, 29  
SoftLayer.managers.hardware, 30  
SoftLayer.managers.image, 35  
SoftLayer.managers.ipsec, 37  
SoftLayer.managers.load\_balancer, 40  
SoftLayer.managers.metadata, 43  
SoftLayer.managers.network, 44  
SoftLayer.managers.ordering, 48  
SoftLayer.managers.sshkey, 54  
SoftLayer.managers.ssl, 55  
SoftLayer.managers.ticket, 56  
SoftLayer.managers.vs, 58  
SoftLayer.managers.vs\_capacity, 66  
SoftLayer.managers.vs\_placement, 68



## Symbols

- aaaa-record
  - hw-dns-sync command line option, 131
  - vs-dns-sync command line option, 181
- abuse <abuse>
  - rwhois-edit command line option, 155
- ack
  - account-event-detail command line option, 82
- ack-all
  - account-events command line option, 81
- address1 <address1>
  - rwhois-edit command line option, 155
- address2 <address2>
  - rwhois-edit command line option, 155
- all
  - account-invoices command line option, 82
- all <all>
  - vs-capture command line option, 178
- amount <amount>
  - autoscale-scale command line option, 83
- billing <billing>
  - block-volume-duplicate command line option, 94
  - block-volume-order command line option, 96
  - dedicatedhost-create command line option, 104
  - file-volume-duplicate command line option, 118
  - file-volume-order command line option, 120
  - hw-create command line option, 126
  - order-place command line option, 150
  - vs-create command line option, 179
- body <body>
  - ticket-create command line option, 167
  - ticket-update command line option, 168
- boot-mode <boot\_mode>
  - vs-create command line option, 179
- by, -to
  - autoscale-scale command line option, 83
- by-id <by\_id>
  - dns-record-edit command line option, 107
- by-record <by\_record>
  - dns-record-edit command line option, 107
- byol
  - image-import command line option, 133
- capacity <capacity>
  - block-snapshot-order command line option, 91
  - file-snapshot-order command line option, 116
- category <category>
  - order-item-list command line option, 149
- city <city>
  - rwhois-edit command line option, 155
- closed
  - account-invoices command line option, 82
- cloud-init
  - image-import command line option, 133
- columns <columns>
  - block-access-list command line option, 86
  - block-replica-locations command line option, 89
  - block-replica-partners command

- line option, 88
- block-snapshot-list command line option, 91
- block-volume-list command line option, 94
- dedicatedhost-list command line option, 103
- dedicatedhost-list-guests command line option, 106
- file-access-list command line option, 111
- file-replica-locations command line option, 114
- file-replica-partners command line option, 113
- file-snapshot-list command line option, 116
- file-volume-list command line option, 119
- hw-list command line option, 128
- user-list command line option, 170
- vs-list command line option, 182
- comment <comment>
  - hw-cancel command line option, 125
- company <company>
  - rwhois-edit command line option, 155
- complex-type <complex\_type>
  - order-place command line option, 150
  - order-place-quote command line option, 153
  - order-quote command line option, 152
- count <count>
  - ticket-detail command line option, 167
- country <country>
  - rwhois-edit command line option, 155
- cpu <cpu>
  - autoscale-edit command line option, 85
  - vs-upgrade command line option, 185
- crt <crt>
  - ssl-add command line option, 163
  - ssl-edit command line option, 163
- csr <csr>
  - ssl-add command line option, 163
  - ssl-edit command line option, 163
- data <data>
  - dns-record-edit command line option, 108
  - dns-record-list command line option, 108
- day-of-week <day\_of\_week>
  - block-snapshot-enable command line option, 90
  - file-snapshot-enable command line option, 116
- dedicated, -public
  - vs-create command line option, 179
- details
  - account-invoice-detail command line option, 82
- disk <disk>
  - vs-create command line option, 179
- dry-run
  - dns-import command line option, 106
- enable, -disable
  - hw-toggle-ipmi command line option, 131
  - user-edit-permissions command line option, 171
- end <end>
  - report-bandwidth command line option, 154
- export <export>
  - dedicatedhost-create command line option, 104
  - hw-create command line option, 126
  - vs-create command line option, 179
- extras <extras>
  - order-place command line option, 150
  - order-place-quote command line option, 153
- firewall-type <firewall\_type>
  - firewall-add command line option, 121
- firstname <firstname>
  - rwhois-edit command line option, 155
- flavor <flavor>
  - vs-upgrade command line option, 185
- fqdn <fqdn>
  - order-quote command line option, 152
- guests
  - dedicatedhost-detail command line option, 104
- ha, -high-availability
  - firewall-add command line option, 121
- hard, -soft
  - hw-reboot command line option, 129
  - vs-power-off command line option, 183
  - vs-reboot command line option, 184
- hardware <hardware\_identifier>
  - ticket-attach command line option, 169
  - ticket-create command line option, 167

ticket-detach command line option, 169  
 -healthInterval <healthinterval>  
   loadbal-l7pool-add command line option, 143  
 -healthPath <healthpath>  
   loadbal-l7pool-add command line option, 143  
 -healthRetry <healthretry>  
   loadbal-l7pool-add command line option, 143  
 -healthTimeout <healthtimeout>  
   loadbal-l7pool-add command line option, 143  
 -history <history>  
   cdn-detail command line option, 98  
 -host-id <host\_id>  
   vs-create command line option, 179  
 -hour <hour>  
   block-snapshot-enable command line option, 90  
   file-snapshot-enable command line option, 115  
 -hourly  
   vs-list command line option, 182  
 -ibm-api-key <ibm\_api\_key>  
   image-export command line option, 134  
   image-import command line option, 133  
 -icc <icc>  
   ssl-add command line option, 163  
   ssl-edit command line option, 163  
 -id <\_id>  
   call-api command line option, 101  
 -identifier <identifier>  
   subnet-list command line option, 166  
 -image <image>  
   order-quote command line option, 152  
   vs-create command line option, 179  
 -immediate  
   block-snapshot-cancel command line option, 89  
   block-volume-cancel command line option, 92  
   file-snapshot-cancel command line option, 114  
   file-volume-cancel command line option, 117  
   hw-cancel command line option, 125  
 -iops <iops>  
   block-volume-order command line option, 95  
   file-volume-order command line option, 120  
 -ip-address <ip\_address>  
   block-access-authorize command line option, 85  
   block-access-revoke command line option, 86  
   file-access-authorize command line option, 111  
   file-access-revoke command line option, 112  
 -ip-version <ip\_version>  
   globalip-list command line option, 123  
 -ipv4, -v4  
   subnet-list command line option, 166  
 -ipv6  
   vs-create command line option, 180  
 -ipv6, -v6  
   subnet-create command line option, 165  
   subnet-list command line option, 166  
 -is-encrypted  
   image-import command line option, 133  
 -key <key>  
   ssl-add command line option, 163  
   ssl-edit command line option, 163  
 -keyword <keyword>  
   order-item-list command line option, 149  
   order-package-list command line option, 147  
   order-preset-list command line option, 149  
 -lastname <lastname>  
   rwhois-edit command line option, 155  
 -like <like>  
   vs-create command line option, 179  
 -limit <limit>  
   account-invoices command line option, 82  
   call-api command line option, 101  
 -location <location>  
   block-volume-order command line option, 95  
   file-volume-order command line option, 120  
 -mask <mask>  
   call-api command line option, 101  
 -max <maximum>  
   autoscale-edit command line option, 84  
 -memory <memory>

- autoscale-edit command line option, 85
- vs-upgrade command line option, 185
- metadata, -no-metadata
  - event-log-get command line option, 110
- min <minimum>
  - autoscale-edit command line option, 84
- minute <minute>
  - block-snapshot-enable command line option, 90
  - file-snapshot-enable command line option, 115
- monthly
  - vs-list command line option, 182
- name <name>
  - autoscale-edit command line option, 84
  - image-edit command line option, 132
  - image-list command line option, 133
  - order-place-quote command line option, 153
  - ticket-upload command line option, 168
  - vlan-list command line option, 173
- network <network>
  - vs-upgrade command line option, 185
- network-space <network\_space>
  - subnet-list command line option, 166
- no-hardware
  - subnet-detail command line option, 166
  - vlan-detail command line option, 173
- no-public
  - hw-create command line option, 126
- no-vs
  - subnet-detail command line option, 166
  - vlan-detail command line option, 173
- note <note>
  - image-edit command line option, 132
  - image-import command line option, 133
  - sshkey-add command line option, 161
  - sshkey-edit command line option, 162
  - vs-capture command line option, 178
- notes <notes>
  - ssl-add command line option, 163
  - ssl-edit command line option, 163
- offset <offset>
  - call-api command line option, 101
- open, -closed
  - ticket-list command line option, 168
- os-code <os\_code>
  - image-import command line option, 133
- os-type <os\_type>
  - block-replica-order command line option, 88
  - block-volume-order command line option, 95
- output-python, -no-output-python
  - call-api command line option, 101
- package\_type <package\_type>
  - order-package-list command line option, 147
- passwords
  - hw-detail command line option, 127
- path <path>
  - ticket-upload command line option, 168
- placementgroup <placementgroup>
  - vs-create command line option, 180
- port <port>
  - dns-record-add command line option, 107
- port-speed <port\_speed>
  - hw-create command line option, 126
- postal <postal>
  - rwhois-edit command line option, 155
- preset <preset>
  - order-place command line option, 150
  - order-place-quote command line option, 153
- price
  - dedicatedhost-detail command line option, 104
  - hw-detail command line option, 127
- priority <priority>
  - dns-record-add command line option, 107
  - ticket-create command line option, 167
- private
  - vs-create command line option, 179
  - vs-upgrade command line option, 185
- private, -public
  - loadbal-member-add command line option, 140
- private-speed <private\_speed>
  - hw-edit command line option, 127
  - vs-edit command line option, 181
- protocol <protocol>
  - dns-record-add command line option, 107
- ptr
  - hw-dns-sync command line option, 131

vs-dns-sync command line option, 181  
 -public  
   loadbal-order command line option, 144  
 -public, -private  
   image-list command line option, 133  
   rwhois-edit command line option, 155  
 -public-speed <public\_speed>  
   hw-edit command line option, 127  
   vs-edit command line option, 181  
 -quantity <quantity>  
   order-place command line option, 150  
   order-quote command line option, 152  
 -reason <reason>  
   block-snapshot-cancel command line option, 89  
   block-volume-cancel command line option, 92  
   file-snapshot-cancel command line option, 114  
   file-volume-cancel command line option, 117  
   hw-cancel command line option, 125  
 -record <record>  
   dns-record-list command line option, 108  
 -replicant-id <replicant\_id>  
   block-replica-failover command line option, 87  
   file-replica-failover command line option, 112  
 -required  
   order-category-list command line option, 148  
 -retention-count <retention\_count>  
   block-snapshot-enable command line option, 90  
   file-snapshot-enable command line option, 115  
 -root-key-crn <root\_key\_crn>  
   image-import command line option, 133  
 -san  
   vs-create command line option, 179  
 -schedule-type <schedule\_type>  
   block-snapshot-disable command line option, 90  
   block-snapshot-enable command line option, 90  
   file-snapshot-disable command line option, 115  
   file-snapshot-enable command line option, 115  
 -send-email  
   order-place-quote command line option, 153  
 -service <service>  
   dns-record-add command line option, 107  
 -service-offering <service\_offering>  
   block-volume-order command line option, 96  
   file-volume-order command line option, 120  
 -size <size>  
   block-volume-order command line option, 95  
   file-volume-order command line option, 120  
 -snapshot-size <snapshot\_size>  
   block-volume-order command line option, 96  
   file-volume-order command line option, 120  
 -sortby <sortby>  
   block-access-list command line option, 86  
   block-replica-locations command line option, 89  
   block-replica-partners command line option, 88  
   block-snapshot-list command line option, 91  
   block-volume-count command line option, 93  
   block-volume-limits command line option, 96  
   block-volume-list command line option, 94  
   cdn-list command line option, 98  
   dedicatedhost-list command line option, 103  
   dedicatedhost-list-guests command line option, 105  
   file-access-list command line option, 111  
   file-replica-locations command line option, 114  
   file-replica-partners command line option, 113  
   file-snapshot-list command line option, 116  
   file-volume-count command line option, 118  
   file-volume-limits command line option, 121  
   file-volume-list command line option, 119

- hw-list command line option, [128](#)
- report-bandwidth command line option, [154](#)
- securitygroup-interface-list command line option, [159](#)
- securitygroup-list command line option, [156](#)
- securitygroup-rule-list command line option, [157](#)
- sshkey-list command line option, [162](#)
- ssl-list command line option, [164](#)
- subnet-list command line option, [166](#)
- summary command line option, [154](#)
- vlan-list command line option, [173](#)
- vs-list command line option, [182](#)
- start <start>
  - report-bandwidth command line option, [154](#)
- state <state>
  - rwhois-edit command line option, [155](#)
- status <status>
  - ssl-list command line option, [164](#)
- storage-type <storage\_type>
  - block-volume-list command line option, [94](#)
  - block-volume-order command line option, [95](#)
  - file-volume-list command line option, [119](#)
  - file-volume-order command line option, [120](#)
- subject-id <subject\_id>
  - ticket-create command line option, [167](#)
- subnet-id <subnet\_id>
  - block-subnets-assign command line option, [97](#)
  - block-subnets-remove command line option, [97](#)
- subnet-private <subnet\_private>
  - vs-create command line option, [180](#)
- subnet-public <subnet\_public>
  - vs-create command line option, [180](#)
- tag <tag>
  - dedicatedhost-list command line option, [103](#)
  - dedicatedhost-list-guests command line option, [105](#)
  - hw-list command line option, [128](#)
  - image-edit command line option, [132](#)
  - vs-list command line option, [182](#)
- test
  - hw-create command line option, [126](#)
  - subnet-create command line option, [165](#)
  - vs-create command line option, [179](#)
- test <test>
  - globalip-create command line option, [123](#)
- tier <tier>
  - block-replica-order command line option, [88](#)
  - block-snapshot-order command line option, [91](#)
  - block-volume-order command line option, [95](#)
  - file-replica-order command line option, [113](#)
  - file-snapshot-order command line option, [116](#)
  - file-volume-order command line option, [120](#)
- title <title>
  - ticket-create command line option, [167](#)
- transient
  - vs-create command line option, [180](#)
- transient <transient>
  - vs-list command line option, [182](#)
- ttl <ttl>
  - dns-record-add command line option, [107](#)
  - dns-record-edit command line option, [108](#)
  - dns-record-list command line option, [108](#)
  - hw-dns-sync command line option, [131](#)
  - vs-dns-sync command line option, [181](#)
- type <type>
  - dns-record-list command line option, [108](#)
- up, -down
  - autoscale-scale command line option, [83](#)
- upgrade
  - block-snapshot-order command line option, [92](#)
  - file-snapshot-order command line option, [116](#)
- userdata <userdata>
  - autoscale-edit command line option, [85](#)
- uuid <uuid>
  - loadbal-health command line option, [139](#)
- verify
  - dedicatedhost-create command line



option, 104  
 loadbal-order command line option, 144  
 order-place command line option, 150  
 order-quote command line option, 152  
 -virtual <virtual\_identifier>  
 ticket-attach command line option, 169  
 ticket-create command line option, 167  
 ticket-detach command line option, 169  
 -vlan-private <vlan\_private>  
 vs-create command line option, 180  
 -vlan-public <vlan\_public>  
 vs-create command line option, 180  
 -wait <wait>  
 hw-create command line option, 126  
 hw-ready command line option, 131  
 vs-create command line option, 180  
 vs-ready command line option, 184  
 -weight <weight>  
 dns-record-add command line option, 107  
 -wrapped-dek <wrapped\_dek>  
 image-import command line option, 133  
 -zone <zone>  
 dns-record-add command line option, 107  
 -D, -date-max <date\_max>  
 event-log-get command line option, 110  
 -D, -disk <disk>  
 dedicatedhost-list command line option, 103  
 -D, -domain <domain>  
 dedicatedhost-create command line option, 103  
 dedicatedhost-list-guests command line option, 105  
 hw-create command line option, 125  
 hw-edit command line option, 127  
 hw-list command line option, 128  
 vs-create command line option, 179  
 vs-edit command line option, 181  
 vs-list command line option, 182  
 -F, -userfile <userfile>  
 autoscale-edit command line option, 85  
 hw-edit command line option, 127  
 order-quote command line option, 152  
 vs-create command line option, 180  
 vs-edit command line option, 181  
 -H, -header <header>  
 cdn-origin-add command line option, 99  
 -H, -hostname <hostname>  
 dedicatedhost-create command line option, 103  
 dedicatedhost-list-guests command line option, 105  
 hw-create command line option, 125  
 hw-edit command line option, 127  
 hw-list command line option, 128  
 vs-create command line option, 179  
 vs-edit command line option, 181  
 vs-list command line option, 182  
 -H, -name <name>  
 dedicatedhost-list command line option, 103  
 -M, -port-max <port\_max>  
 securitygroup-rule-add command line option, 158  
 securitygroup-rule-edit command line option, 159  
 -P, -frontProtocol <frontprotocol>  
 loadbal-pool-add command line option, 141  
 loadbal-pool-edit command line option, 141  
 -P, -protocol <protocol>  
 cdn-origin-add command line option, 99  
 loadbal-l7pool-add command line option, 143  
 -S, -public-security-group <public\_security\_group>  
 vs-create command line option, 180  
 -S, -server <server>  
 loadbal-l7pool-add command line option, 143  
 -a, -a-record  
 hw-dns-sync command line option, 131  
 vs-dns-sync command line option, 181  
 -a, -api-key  
 user-create command line option, 172  
 -b, -backPort <backport>  
 loadbal-pool-add command line option, 141  
 loadbal-pool-edit command line option, 141  
 -b, -backend <backend>  
 loadbal-order command line option, 144  
 -b, -bucket-name <bucket\_name>  
 cdn-origin-add command line option, 99

-c, -cache-query <cache\_query>  
cdn-origin-add command line option,  
99

-c, -connections <connections>  
loadbal-pool-add command line  
option, 141  
loadbal-pool-edit command line  
option, 142

-c, -cpu <cpu>  
dedicatedhost-list command line  
option, 103  
dedicatedhost-list-guests command  
line option, 105  
hw-list command line option, 128  
vs-create command line option, 179  
vs-list command line option, 182

-c, -credential\_id <credential\_id>  
object-storage-credential-delete  
command line option, 146

-c, -duplicate-size <duplicate\_size>  
block-volume-duplicate command  
line option, 93  
file-volume-duplicate command line  
option, 118

-c, -new-size <new\_size>  
block-volume-modify command line  
option, 95  
file-volume-modify command line  
option, 119

-d, -datacenter <datacenter>  
block-volume-count command line  
option, 93  
block-volume-list command line  
option, 94  
dedicatedhost-create command line  
option, 103  
dedicatedhost-create-options  
command line option, 104  
dedicatedhost-list command line  
option, 103  
file-volume-count command line  
option, 118  
file-volume-list command line  
option, 119  
hw-create command line option, 126  
hw-list command line option, 128  
loadbal-order command line option,  
144  
loadbal-order-options command line  
option, 144  
subnet-list command line option, 166  
vlan-list command line option, 173  
vs-create command line option, 179  
vs-list command line option, 182

-d, -date-min <date\_min>  
autoscale-logs command line option,  
84  
event-log-get command line option,  
110

-d, -description <description>  
securitygroup-create command line  
option, 156  
securitygroup-edit command line  
option, 156

-d, -direction <direction>  
securitygroup-rule-add command  
line option, 158  
securitygroup-rule-edit command  
line option, 158

-e, -email <email>  
user-create command line option, 172

-e, -end\_date <end\_date>  
hw-bandwidth command line option,  
124  
vs-bandwidth command line option,  
177  
vs-usage command line option, 185

-e, -ethertype <ethertype>  
securitygroup-rule-add command  
line option, 158  
securitygroup-rule-edit command  
line option, 158

-e, -events  
user-detail command line option, 170

-e, -extensions <extensions>  
cdn-origin-add command line option,  
99

-e, -extra <extra>  
hw-create command line option, 126

-e, -obj-event <obj\_event>  
event-log-get command line option,  
110

-f, -filter <\_filters>  
call-api command line option, 101

-f, -flavor <flavor>  
dedicatedhost-create command line  
option, 104  
dedicatedhost-create-options  
command line option, 104  
vs-create command line option, 179

-f, -frontPort <frontport>  
loadbal-pool-add command line  
option, 141  
loadbal-pool-edit command line  
option, 141

-f, -frontend <frontend>  
loadbal-order command line option,  
144

-f, -in-file <in\_file>  
     sshkey-add command line option, 161  
 -f, -out-file <out\_file>  
     sshkey-print command line option,  
     162  
 -g, -tag <tag>  
     hw-edit command line option, 127  
     vs-create command line option, 180  
     vs-edit command line option, 181  
 -g, -tags <tags>  
     autoscale-tag command line option,  
     84  
 -h, -hardware  
     user-detail command line option, 170  
 -h, -hardware-id <hardware\_id>  
     block-access-authorize command  
     line option, 85  
     block-access-revoke command line  
     option, 86  
     file-access-authorize command line  
     option, 111  
     file-access-revoke command line  
     option, 112  
 -i, -duplicate-iops <duplicate\_iops>  
     block-volume-duplicate command  
     line option, 93  
     file-volume-duplicate command line  
     option, 118  
 -i, -interface <interface>  
     securitygroup-interface-add  
     command line option, 160  
     securitygroup-interface-remove  
     command line option, 160  
 -i, -interval <interval>  
     loadbal-health command line option,  
     139  
 -i, -ip-address-id <ip\_address\_id>  
     block-access-authorize command  
     line option, 85  
     block-access-revoke command line  
     option, 86  
     file-access-authorize command line  
     option, 111  
     file-access-revoke command line  
     option, 112  
 -i, -new-iops <new\_iops>  
     block-volume-modify command line  
     option, 95  
     file-volume-modify command line  
     option, 119  
 -i, -obj-id <obj\_id>  
     event-log-get command line option,  
     110  
 -i, -postinstall <postinstall>  
     hw-create command line option, 126  
     hw-reload command line option, 129  
     order-quote command line option, 152  
     vs-create command line option, 179  
 -k, -key <key>  
     hw-create command line option, 126  
     hw-reload command line option, 129  
     order-quote command line option, 152  
     sshkey-add command line option, 161  
     vs-create command line option, 179  
 -k, -keys  
     user-detail command line option, 170  
 -k, -label <label>  
     sshkey-edit command line option, 162  
 -l, -label <label>  
     loadbal-order command line option,  
     144  
 -l, -limit <limit>  
     event-log-get command line option,  
     110  
     hw-list command line option, 128  
     securitygroup-list command line  
     option, 156  
     vlan-list command line option, 173  
     vs-list command line option, 182  
 -l, -location <location>  
     block-replica-order command line  
     option, 88  
     file-replica-order command line  
     option, 113  
 -l, -logins  
     user-detail command line option, 170  
 -m, -member <member>  
     loadbal-member-add command line  
     option, 140  
     loadbal-member-remove command line  
     option, 140  
 -m, -memory <memory>  
     dedicatedhost-list command line  
     option, 103  
     dedicatedhost-list-guests command  
     line option, 105  
     hw-list command line option, 128  
     vs-create command line option, 179  
     vs-list command line option, 182  
 -m, -method <method>  
     loadbal-l7pool-add command line  
     option, 142  
     loadbal-order command line option,  
     144  
     loadbal-pool-add command line  
     option, 141  
     loadbal-pool-edit command line  
     option, 142

-m, -port-min <port\_min>  
securitygroup-rule-add command line option, 158  
securitygroup-rule-edit command line option, 159

-n, -name <name>  
loadbal-l7pool-add command line option, 142  
loadbal-order command line option, 144  
securitygroup-create command line option, 156  
securitygroup-edit command line option, 156  
vs-capture command line option, 178

-n, -network <network>  
hw-list command line option, 128  
vs-create command line option, 179  
vs-list command line option, 182

-n, -network-component <network\_component>  
securitygroup-interface-add command line option, 160  
securitygroup-interface-remove command line option, 160

-n, -notes <notes>  
block-snapshot-create command line option, 89  
file-snapshot-create command line option, 114

-n, -number <number>  
vlan-list command line option, 173

-o, -optimize-for <optimize\_for>  
cdn-origin-add command line option, 99

-o, -origin-snapshot-id <origin\_snapshot\_id>  
block-volume-duplicate command line option, 93  
file-volume-duplicate command line option, 118

-o, -os <os>  
hw-create command line option, 125  
vs-create command line option, 179

-p, -backProtocol <backprotocol>  
loadbal-pool-add command line option, 141  
loadbal-pool-edit command line option, 141

-p, -password <password>  
block-access-password command line option, 87  
user-create command line option, 172

-p, -permission <permission>  
user-edit-permissions command line option, 171

-p, -permissions  
user-detail command line option, 170

-p, -port <port>  
cdn-origin-add command line option, 99

-p, -protocol <protocol>  
securitygroup-rule-add command line option, 158  
securitygroup-rule-edit command line option, 159

-p, -summary\_period <summary\_period>  
hw-bandwidth command line option, 124  
vs-bandwidth command line option, 177  
vs-usage command line option, 185

-q, -quite\_summary  
hw-bandwidth command line option, 124  
vs-bandwidth command line option, 177

-r, -remote-ip <remote\_ip>  
securitygroup-rule-add command line option, 158  
securitygroup-rule-edit command line option, 158

-r, -retry <retry>  
loadbal-health command line option, 139

-r, -router <router>  
dedicatedhost-create command line option, 103

-s, -duplicate-snapshot-size <duplicate\_snapshot\_size>  
block-volume-duplicate command line option, 94  
file-volume-duplicate command line option, 118

-s, -private-security-group <private\_security\_group>  
vs-create command line option, 180

-s, -remote-group <remote\_group>  
securitygroup-rule-add command line option, 158  
securitygroup-rule-edit command line option, 158

-s, -server <server>  
securitygroup-interface-add command line option, 160  
securitygroup-interface-remove command line option, 160

-s, -size <size>

hw-create command line option, 125  
 -s, -snapshot-id <snapshot\_id>  
   block-snapshot-restore command line option, 92  
   file-snapshot-restore command line option, 117  
 -s, -snapshot-schedule <snapshot\_schedule>  
   block-replica-order command line option, 88  
   file-replica-order command line option, 113  
 -s, -start\_date <start\_date>  
   hw-bandwidth command line option, 124  
   vs-bandwidth command line option, 177  
   vs-usage command line option, 185  
 -s, -sticky  
   loadbal-l7pool-add command line option, 143  
   loadbal-pool-add command line option, 141  
   loadbal-pool-edit command line option, 142  
 -s, -subnet <subnet>  
   loadbal-order command line option, 144  
 -s, -subnet-id <subnet\_id>  
   file-access-authorize command line option, 111  
   file-access-revoke command line option, 112  
 -t, -duplicate-tier <duplicate\_tier>  
   block-volume-duplicate command line option, 93  
   file-volume-duplicate command line option, 118  
 -t, -new-tier <new\_tier>  
   block-volume-modify command line option, 95  
   file-volume-modify command line option, 119  
 -t, -obj-type <obj\_type>  
   event-log-get command line option, 110  
 -t, -origin-type <origin\_type>  
   cdn-origin-add command line option, 99  
 -t, -subnet-type <subnet\_type>  
   subnet-list command line option, 166  
 -t, -template <template>  
   dedicatedhost-create command line option, 104  
   hw-create command line option, 126  
   user-create command line option, 172  
   user-edit-details command line option, 171  
   vs-create command line option, 180  
 -t, -timeout <timeout>  
   loadbal-health command line option, 139  
 -t, -valid\_type <valid\_type>  
   vs-usage command line option, 185  
 -u, -from-user <from\_user>  
   user-create command line option, 172  
   user-edit-permissions command line option, 171  
 -u, -url <url>  
   loadbal-health command line option, 139  
 -u, -userdata <userdata>  
   hw-edit command line option, 127  
   order-quote command line option, 152  
   vs-create command line option, 180  
   vs-edit command line option, 181  
 -u, -username <username>  
   block-volume-list command line option, 94  
   file-volume-list command line option, 119  
 -v, -virtual  
   user-detail command line option, 170  
 -v, -virtual-id <virtual\_id>  
   block-access-authorize command line option, 85  
   block-access-revoke command line option, 86  
   file-access-authorize command line option, 111  
   file-access-revoke command line option, 112  
 -v6, -ipv6  
   globalip-create command line option, 123  
 -w, -weight <weight>  
   loadbal-member-add command line option, 140  
 -x, -sslCert <sslcert>  
   loadbal-pool-add command line option, 141  
   loadbal-pool-edit command line option, 142  
 -z, -utc-offset <utc\_offset>  
   event-log-get command line option, 110

## A

- ACCESS\_ID
- block-access-password command line option, 87
  - block-subnets-assign command line option, 97
  - block-subnets-list command line option, 97
  - block-subnets-remove command line option, 98
- account-event-detail command line option
- ack, 82
  - IDENTIFIER, 82
- account-events command line option
- ack-all, 81
- account-invoice-detail command line option
- details, 82
  - IDENTIFIER, 82
- account-invoices command line option
- all, 82
  - closed, 82
  - limit <limit>, 82
- AccountManager (class in SoftLayer.managers.account), 8
- ack\_event() (SoftLayer.managers.account.AccountManager method), 8
- add\_certificate() (SoftLayer.managers.ssl.SSLManager method), 55
- add\_global\_ip() (SoftLayer.managers.network.NetworkManager method), 44
- add\_internal\_subnet() (SoftLayer.managers.ipsec.IPSECManger method), 37
- add\_key() (SoftLayer.managers.sshkey.SshKeyManager method), 54
- add\_lb\_l7\_pool() (SoftLayer.managers.load\_balancer.LoadBalancerManager method), 40
- add\_lb\_listener() (SoftLayer.managers.load\_balancer.LoadBalancerManager method), 41
- add\_lb\_member() (SoftLayer.managers.load\_balancer.LoadBalancerManager method), 41
- add\_origin() (SoftLayer.managers.cdn.CDNManager method), 16
- add\_remote\_subnet() (SoftLayer.managers.ipsec.IPSECManger method), 37
- add\_securitygroup\_rule() (SoftLayer.managers.network.NetworkManager method), 44
- add\_securitygroup\_rules() (SoftLayer.managers.network.NetworkManager method), 44
- add\_service\_subnet() (SoftLayer.managers.ipsec.IPSECManger method), 37
- add\_standard\_firewall() (SoftLayer.managers.firewall.FirewallManager method), 29
- add\_subnet() (SoftLayer.managers.network.NetworkManager method), 44
- add\_vlan\_firewall() (SoftLayer.managers.firewall.FirewallManager method), 29
- apply\_configuration() (SoftLayer.managers.ipsec.IPSECManger method), 38
- assign\_global\_ip() (SoftLayer.managers.network.NetworkManager method), 45
- assign\_subnets\_to\_acl() (SoftLayer.managers.block.BlockStorageManager method), 10
- attach\_hardware() (SoftLayer.managers.ticket.TicketManager method), 56
- attach\_securitygroup\_component() (SoftLayer.managers.network.NetworkManager method), 45
- attach\_securitygroup\_components() (SoftLayer.managers.network.NetworkManager method), 45
- attach\_virtual\_server() (SoftLayer.managers.ticket.TicketManager method), 56
- authenticate\_with\_password() (SoftLayer.BaseClient method), 71
- authorize\_host\_to\_volume() (SoftLayer.managers.block.BlockStorageManager method), 10
- authorize\_host\_to\_volume() (SoftLayer.managers.file.FileStorageManager method), 24
- autoscale-detail command line option
- IDENTIFIER, 83
- autoscale-edit command line option
- cpu <cpu>, 85
  - max <maximum>, 84
  - memory <memory>, 85
  - min <minimum>, 84

- name <name>, 84
  - userdata <userdata>, 85
  - F, -userfile <userfile>, 85
  - IDENTIFIER, 85
  - autoscale-logs command line option
    - d, -date-min <date\_min>, 84
    - IDENTIFIER, 84
  - autoscale-scale command line option
    - amount <amount>, 83
    - by, -to, 83
    - up, -down, 83
    - IDENTIFIER, 83
  - autoscale-tag command line option
    - g, -tags <tags>, 84
    - IDENTIFIER, 84
  - AutoScaleManager (class in *SoftLayer.managers.autoscale*), 9
- ## B
- BaseClient (class in *SoftLayer*), 71
  - BasicAuthentication (class in *SoftLayer*), 73
  - block-access-authorize command line option
    - ip-address <ip\_address>, 85
    - h, -hardware-id <hardware\_id>, 85
    - i, -ip-address-id <ip\_address\_id>, 85
    - v, -virtual-id <virtual\_id>, 85
    - VOLUME\_ID, 85
  - block-access-list command line option
    - columns <columns>, 86
    - sortby <sortby>, 86
    - VOLUME\_ID, 86
  - block-access-password command line option
    - p, -password <password>, 87
    - ACCESS\_ID, 87
  - block-access-revoke command line option
    - ip-address <ip\_address>, 86
    - h, -hardware-id <hardware\_id>, 86
    - i, -ip-address-id <ip\_address\_id>, 86
    - v, -virtual-id <virtual\_id>, 86
    - VOLUME\_ID, 86
  - block-replica-failback command line option
    - VOLUME\_ID, 87
  - block-replica-failover command line option
    - replicant-id <replicant\_id>, 87
    - VOLUME\_ID, 87
  - block-replica-locations command line option
    - columns <columns>, 89
    - orderby <orderby>, 89
    - VOLUME\_ID, 89
  - block-replica-order command line option
    - os-type <os\_type>, 88
    - tier <tier>, 88
    - l, -location <location>, 88
    - s, -snapshot-schedule <snapshot\_schedule>, 88
    - VOLUME\_ID, 88
  - block-replica-partners command line option
    - columns <columns>, 88
    - orderby <orderby>, 88
    - VOLUME\_ID, 88
  - block-snapshot-cancel command line option
    - immediate, 89
    - reason <reason>, 89
    - VOLUME\_ID, 89
  - block-snapshot-create command line option
    - n, -notes <notes>, 89
    - VOLUME\_ID, 89
  - block-snapshot-delete command line option
    - SNAPSHOT\_ID, 90
  - block-snapshot-disable command line option
    - schedule-type <schedule\_type>, 90
    - VOLUME\_ID, 90
  - block-snapshot-enable command line option
    - day-of-week <day\_of\_week>, 90
    - hour <hour>, 90
    - minute <minute>, 90
    - retention-count <retention\_count>, 90
    - schedule-type <schedule\_type>, 90
    - VOLUME\_ID, 91
  - block-snapshot-list command line option
    - columns <columns>, 91
    - orderby <orderby>, 91
    - VOLUME\_ID, 91
  - block-snapshot-order command line option
    - capacity <capacity>, 91
    - tier <tier>, 91
    - upgrade, 92
    - VOLUME\_ID, 92
  - block-snapshot-restore command line option



-s, -snapshot-id <snapshot\_id>, 92  
 VOLUME\_ID, 92  
 block-snapshot-schedule-list command line option  
 VOLUME\_ID, 91  
 block-subnets-assign command line option  
 -subnet-id <subnet\_id>, 97  
 ACCESS\_ID, 97  
 block-subnets-list command line option  
 ACCESS\_ID, 97  
 block-subnets-remove command line option  
 -subnet-id <subnet\_id>, 97  
 ACCESS\_ID, 98  
 block-volume-cancel command line option  
 -immediate, 92  
 -reason <reason>, 92  
 VOLUME\_ID, 92  
 block-volume-count command line option  
 -sortby <sortby>, 93  
 -d, -datacenter <datacenter>, 93  
 block-volume-detail command line option  
 VOLUME\_ID, 93  
 block-volume-duplicate command line option  
 -billing <billing>, 94  
 -c, -duplicate-size <duplicate\_size>, 93  
 -i, -duplicate-iops <duplicate\_iops>, 93  
 -o, -origin-snapshot-id <origin\_snapshot\_id>, 93  
 -s, -duplicate-snapshot-size <duplicate\_snapshot\_size>, 94  
 -t, -duplicate-tier <duplicate\_tier>, 93  
 ORIGIN\_VOLUME\_ID, 94  
 block-volume-limits command line option  
 -sortby <sortby>, 96  
 block-volume-list command line option  
 -columns <columns>, 94  
 -sortby <sortby>, 94  
 -storage-type <storage\_type>, 94  
 -d, -datacenter <datacenter>, 94  
 -u, -username <username>, 94  
 block-volume-modify command line option  
 -c, -new-size <new\_size>, 95  
 -i, -new-iops <new\_iops>, 95  
 -t, -new-tier <new\_tier>, 95

VOLUME\_ID, 95  
 block-volume-order command line option  
 -billing <billing>, 96  
 -iops <iops>, 95  
 -location <location>, 95  
 -os-type <os\_type>, 95  
 -service-offering <service\_offering>, 96  
 -size <size>, 95  
 -snapshot-size <snapshot\_size>, 96  
 -storage-type <storage\_type>, 95  
 -tier <tier>, 95  
 block-volume-set-lun-id command line option  
 LUN\_ID, 96  
 VOLUME\_ID, 96  
 BlockStorageManager (class in SoftLayer.managers.block), 10  
 build\_filter() (SoftLayer.managers.event\_log.EventLogManager static method), 23

## C

call() (SoftLayer.BaseClient method), 72  
 call-api command line option  
 -id <\_id>, 101  
 -limit <limit>, 101  
 -mask <mask>, 101  
 -offset <offset>, 101  
 -output-python, -no-output-python, 101  
 -f, -filter <\_filters>, 101  
 METHOD, 101  
 PARAMETERS, 101  
 SERVICE, 101  
 cancel\_block\_volume() (SoftLayer.managers.block.BlockStorageManager method), 11  
 cancel\_file\_volume() (SoftLayer.managers.file.FileStorageManager method), 24  
 cancel\_firewall() (SoftLayer.managers.firewall.FirewallManager method), 29  
 cancel\_global\_ip() (SoftLayer.managers.network.NetworkManager method), 45  
 cancel\_guests() (SoftLayer.managers.dedicated\_host.DedicatedHostManager method), 18  
 cancel\_hardware() (SoftLayer.managers.hardware.HardwareManager method), 31



cancel\_host() (SoftLayer.managers.dedicated\_host.DedicatedHostManager method), 18

cancel\_instance() (SoftLayer.managers.vs.VSManager method), 58

cancel\_lbaas() (SoftLayer.managers.load\_balancer.LoadBalancerManager method), 41

cancel\_snapshot\_space() (SoftLayer.managers.block.BlockStorageManager method), 11

cancel\_snapshot\_space() (SoftLayer.managers.file.FileStorageManager method), 24

cancel\_subnet() (SoftLayer.managers.network.NetworkManager method), 45

CapacityManager (class in SoftLayer.managers.vs\_capacity), 66

capture() (SoftLayer.managers.vs.VSManager method), 58

cdn-detail command line option  
-history <history>, 98  
UNIQUE\_ID, 98

cdn-list command line option  
-sortby <sortby>, 98

cdn-origin-add command line option  
-H, -header <header>, 99  
-P, -protocol <protocol>, 99  
-b, -bucket-name <bucket\_name>, 99  
-c, -cache-query <cache\_query>, 99  
-e, -extensions <extensions>, 99  
-o, -optimize-for <optimize\_for>, 99  
-p, -port <port>, 99  
-t, -origin-type <origin\_type>, 99  
ORIGIN, 99  
PATH, 99  
UNIQUE\_ID, 99

cdn-origin-list command line option  
UNIQUE\_ID, 99

cdn-origin-remove command line option  
ORIGIN\_PATH, 100  
UNIQUE\_ID, 100

cdn-purge command line option  
PATH, 100  
UNIQUE\_ID, 100

CDNManager (class in SoftLayer.managers.cdn), 16

change\_port\_speed() (SoftLayer.managers.hardware.HardwareManager method), 31

change\_port\_speed() (SoftLayer.managers.vs.VSManager method), 59

Client() (in module SoftLayer), 73

create() (SoftLayer.managers.vs\_capacity.CapacityManager method), 67

create() (SoftLayer.managers.vs\_placement.PlacementManager method), 68

create\_client\_from\_env() (in module SoftLayer), 72

create\_guest() (SoftLayer.managers.vs\_capacity.CapacityManager method), 67

create\_instance() (SoftLayer.managers.vs.VSManager method), 59

create\_instances() (SoftLayer.managers.vs.VSManager method), 61

create\_or\_update\_lun\_id() (SoftLayer.managers.block.BlockStorageManager method), 11

create\_record() (SoftLayer.managers.dns.DNSManager method), 20

create\_record\_mx() (SoftLayer.managers.dns.DNSManager method), 21

create\_record\_ptr() (SoftLayer.managers.dns.DNSManager method), 21

create\_record\_srv() (SoftLayer.managers.dns.DNSManager method), 21

create\_remote\_subnet() (SoftLayer.managers.ipsec.IPSECManger method), 38

create\_securitygroup() (SoftLayer.managers.network.NetworkManager method), 45

create\_snapshot() (SoftLayer.managers.block.BlockStorageManager method), 11

create\_snapshot() (SoftLayer.managers.file.FileStorageManager method), 25

create\_ticket() (SoftLayer.managers.ticket.TicketManager method), 57

create\_translation() (SoftLayer.managers.ipsec.IPSECManger method), 38

create\_zone() (SoftLayer.managers.dns.DNSManager method), 21

## D

## DATA

- dns-record-add command line option, 107
- deauthorize\_host\_to\_volume() (*SoftLayer.managers.block.BlockStorageManager method*), 11
- deauthorize\_host\_to\_volume() (*SoftLayer.managers.file.FileStorageManager method*), 25
- dedicatedhost-cancel command line option
  - IDENTIFIER, 105
- dedicatedhost-cancel-guests command line option
  - IDENTIFIER, 105
- dedicatedhost-create command line option
  - billing <billing>, 104
  - export <export>, 104
  - verify, 104
  - D, -domain <domain>, 103
  - H, -hostname <hostname>, 103
  - d, -datacenter <datacenter>, 103
  - f, -flavor <flavor>, 104
  - r, -router <router>, 103
  - t, -template <template>, 104
- dedicatedhost-create-options command line option
  - d, -datacenter <datacenter>, 104
  - f, -flavor <flavor>, 104
- dedicatedhost-detail command line option
  - guests, 104
  - price, 104
  - IDENTIFIER, 105
- dedicatedhost-list command line option
  - columns <columns>, 103
  - sortby <sortby>, 103
  - tag <tag>, 103
  - D, -disk <disk>, 103
  - H, -name <name>, 103
  - c, -cpu <cpu>, 103
  - d, -datacenter <datacenter>, 103
  - m, -memory <memory>, 103
- dedicatedhost-list-guests command line option
  - columns <columns>, 106
  - sortby <sortby>, 105
  - tag <tag>, 105
  - D, -domain <domain>, 105
  - H, -hostname <hostname>, 105
  - c, -cpu <cpu>, 105
  - m, -memory <memory>, 105
- IDENTIFIER, 106
- DedicatedHostManager (*class in SoftLayer.managers.dedicated\_host*), 18
- del\_lb\_l7\_pool() (*SoftLayer.managers.load\_balancer.LoadBalancerManager method*), 41
- delete() (*SoftLayer.managers.vs\_placement.PlacementManager method*), 68
- delete\_image() (*SoftLayer.managers.image.ImageManager method*), 35
- delete\_key() (*SoftLayer.managers.sshkey.SshKeyManager method*), 54
- delete\_lb\_member() (*SoftLayer.managers.load\_balancer.LoadBalancerManager method*), 41
- delete\_record() (*SoftLayer.managers.dns.DNSManager method*), 21
- delete\_remote\_subnet() (*SoftLayer.managers.ipsec.IPSECManger method*), 38
- delete\_securitygroup() (*SoftLayer.managers.network.NetworkManager method*), 45
- delete\_snapshot() (*SoftLayer.managers.block.BlockStorageManager method*), 12
- delete\_snapshot() (*SoftLayer.managers.file.FileStorageManager method*), 25
- delete\_zone() (*SoftLayer.managers.dns.DNSManager method*), 21
- detach\_hardware() (*SoftLayer.managers.ticket.TicketManager method*), 57
- detach\_securitygroup\_component() (*SoftLayer.managers.network.NetworkManager method*), 45
- detach\_securitygroup\_components() (*SoftLayer.managers.network.NetworkManager method*), 45
- detach\_virtual\_server() (*SoftLayer.managers.ticket.TicketManager method*), 57
- details() (*SoftLayer.managers.autoscale.AutoScaleManager method*), 9
- disable\_snapshots() (*SoftLayer.managers.block.BlockStorageManager method*), 12
- disable\_snapshots() (*SoftLayer.managers.file.FileStorageManager method*), 25

- method*), 25
- dns-import command line option
- dry-run, 106
  - ZONEFILE, 106
- dns-record-add command line option
- port <port>, 107
  - priority <priority>, 107
  - protocol <protocol>, 107
  - service <service>, 107
  - ttl <ttl>, 107
  - weight <weight>, 107
  - zone <zone>, 107
  - DATA, 107
  - RECORD, 107
  - RECORD\_TYPE, 107
- dns-record-edit command line option
- by-id <by\_id>, 107
  - by-record <by\_record>, 107
  - data <data>, 108
  - ttl <ttl>, 108
  - ZONE\_ID, 108
- dns-record-list command line option
- data <data>, 108
  - record <record>, 108
  - ttl <ttl>, 108
  - type <type>, 108
  - ZONE, 108
- dns-record-remove command line option
- RECORD\_ID, 108
- dns-zone-create command line option
- ZONE, 109
- dns-zone-delete command line option
- ZONE, 109
- dns-zone-print command line option
- ZONE, 109
- DNSManager (*class in SoftLayer.managers.dns*), 20
- dump\_zone() (*SoftLayer.managers.dns.DNSManager method*), 22
- ## E
- edit() (*SoftLayer.managers.autoscale.AutoScaleManager method*), 9
- edit() (*SoftLayer.managers.hardware.HardwareManager method*), 32
- edit() (*SoftLayer.managers.image.ImageManager method*), 35
- edit() (*SoftLayer.managers.vs.VSManager method*), 61
- edit\_certificate() (*SoftLayer.managers.ssl.SSLManager method*), 55
- edit\_dedicated\_fwl\_rules() (*SoftLayer.managers.firewall.FirewallManager method*), 29
- edit\_key() (*SoftLayer.managers.sshkey.SshKeyManager method*), 54
- edit\_record() (*SoftLayer.managers.dns.DNSManager method*), 22
- edit\_rwhois() (*SoftLayer.managers.network.NetworkManager method*), 46
- edit\_securitygroup() (*SoftLayer.managers.network.NetworkManager method*), 46
- edit\_securitygroup\_rule() (*SoftLayer.managers.network.NetworkManager method*), 46
- edit\_standard\_fwl\_rules() (*SoftLayer.managers.firewall.FirewallManager method*), 29
- edit\_zone() (*SoftLayer.managers.dns.DNSManager method*), 22
- enable\_snapshots() (*SoftLayer.managers.block.BlockStorageManager method*), 12
- enable\_snapshots() (*SoftLayer.managers.file.FileStorageManager method*), 25
- end\_date (*SoftLayer.managers.cdn.CDNManager attribute*), 16
- ENDPOINT\_ID
- subnet-create command line option, 165
- event-log-get command line option
- metadata, -no-metadata, 110
  - D, -date-max <date\_max>, 110
  - d, -date-min <date\_min>, 110
  - e, -obj-event <obj\_event>, 110
  - i, -obj-id <obj\_id>, 110
  - l, -limit <limit>, 110
  - t, -obj-type <obj\_type>, 110
  - z, -utc-offset <utc\_offset>, 110
- EventManager (*class in SoftLayer.managers.event\_log*), 23
- export\_image\_to\_uri() (*SoftLayer.managers.image.ImageManager method*), 35
- ## F
- failback\_from\_replicant() (*SoftLayer.managers.block.BlockStorageManager method*), 12
- failback\_from\_replicant() (*SoftLayer.managers.file.FileStorageManager method*), 26
- failover\_to\_replicant() (*SoftLayer.managers.block.BlockStorageManager method*), 12

*method*), 12

`failover_to_replicant()` (*SoftLayer.managers.file.FileStorageManager method*), 26

`file-access-authorize` command line option

- `-ip-address <ip_address>`, 111
- `-h, -hardware-id <hardware_id>`, 111
- `-i, -ip-address-id <ip_address_id>`, 111
- `-s, -subnet-id <subnet_id>`, 111
- `-v, -virtual-id <virtual_id>`, 111

VOLUME\_ID, 111

`file-access-list` command line option

- `-columns <columns>`, 111
- `-sortby <sortby>`, 111

VOLUME\_ID, 111

`file-access-revoke` command line option

- `-ip-address <ip_address>`, 112
- `-h, -hardware-id <hardware_id>`, 112
- `-i, -ip-address-id <ip_address_id>`, 112
- `-s, -subnet-id <subnet_id>`, 112
- `-v, -virtual-id <virtual_id>`, 112

VOLUME\_ID, 112

`file-replica-failback` command line option

VOLUME\_ID, 112

`file-replica-failover` command line option

- `-replicant-id <replicant_id>`, 112

VOLUME\_ID, 113

`file-replica-locations` command line option

- `-columns <columns>`, 114
- `-sortby <sortby>`, 114

VOLUME\_ID, 114

`file-replica-order` command line option

- `-tier <tier>`, 113
- `-l, -location <location>`, 113
- `-s, -snapshot-schedule <snapshot_schedule>`, 113

VOLUME\_ID, 113

`file-replica-partners` command line option

- `-columns <columns>`, 113
- `-sortby <sortby>`, 113

VOLUME\_ID, 113

`file-snapshot-cancel` command line option

- `-immediate`, 114
- `-reason <reason>`, 114

VOLUME\_ID, 114

`file-snapshot-create` command line option

- `-n, -notes <notes>`, 114

VOLUME\_ID, 115

`file-snapshot-delete` command line option

- `SNAPSHOT_ID`, 115

`file-snapshot-disable` command line option

- `-schedule-type <schedule_type>`, 115

VOLUME\_ID, 115

`file-snapshot-enable` command line option

- `-day-of-week <day_of_week>`, 116
- `-hour <hour>`, 115
- `-minute <minute>`, 115
- `-retention-count <retention_count>`, 115
- `-schedule-type <schedule_type>`, 115

VOLUME\_ID, 116

`file-snapshot-list` command line option

- `-columns <columns>`, 116
- `-sortby <sortby>`, 116

VOLUME\_ID, 116

`file-snapshot-order` command line option

- `-capacity <capacity>`, 116
- `-tier <tier>`, 116

`-upgrade`, 116

VOLUME\_ID, 117

`file-snapshot-restore` command line option

- `-s, -snapshot-id <snapshot_id>`, 117

VOLUME\_ID, 117

`file-snapshot-schedule-list` command line option

VOLUME\_ID, 121

`file-volume-cancel` command line option

- `-immediate`, 117
- `-reason <reason>`, 117

VOLUME\_ID, 117

`file-volume-count` command line option

- `-sortby <sortby>`, 118

- `-d, -datacenter <datacenter>`, 118

`file-volume-detail` command line option

VOLUME\_ID, 118

`file-volume-duplicate` command line option

- `-billing <billing>`, 118
- `-c, -duplicate-size <duplicate_size>`, 118
- `-i, -duplicate-iops <duplicate_iops>`, 118

- o, -origin-snapshot-id  
    <origin\_snapshot\_id>, 118
  - s, -duplicate-snapshot-size  
    <duplicate\_snapshot\_size>, 118
  - t, -duplicate-tier  
    <duplicate\_tier>, 118
  - ORIGIN\_VOLUME\_ID, 119
  - file-volume-limits command line option  
    -sortby <sortby>, 121
  - file-volume-list command line option  
    -columns <columns>, 119  
    -sortby <sortby>, 119  
    -storage-type <storage\_type>, 119  
    -d, -datacenter <datacenter>, 119  
    -u, -username <username>, 119
  - file-volume-modify command line option  
    -c, -new-size <new\_size>, 119  
    -i, -new-iops <new\_iops>, 119  
    -t, -new-tier <new\_tier>, 119
  - VOLUME\_ID, 120
  - file-volume-order command line option  
    -billing <billing>, 120  
    -iops <iops>, 120  
    -location <location>, 120  
    -service-offering  
        <service\_offering>, 120  
    -size <size>, 120  
    -snapshot-size <snapshot\_size>, 120  
    -storage-type <storage\_type>, 120  
    -tier <tier>, 120
  - FileStorageManager (class in Soft-  
    Layer.managers.file), 24
  - filter\_outlet\_packages() (Soft-  
    Layer.managers.ordering.OrderingManager  
    static method), 49
  - firewall-add command line option  
    -firewall-type <firewall\_type>, 121  
    -ha, -high-availability, 121  
    TARGET, 121
  - firewall-cancel command line option  
    IDENTIFIER, 122
  - firewall-detail command line option  
    IDENTIFIER, 122
  - firewall-edit command line option  
    IDENTIFIER, 122
  - FirewallManager (class in Soft-  
    Layer.managers.firewall), 29
- ## G
- generate\_order() (Soft-  
    Layer.managers.ordering.OrderingManager  
    method), 49
  - generate\_order\_template() (Soft-  
    Layer.managers.ordering.OrderingManager  
    method), 49
  - get() (SoftLayer.managers.metadata.MetadataManager  
    method), 43
  - get\_adc() (SoftLayer.managers.load\_balancer.LoadBalancerManager  
    method), 41
  - get\_adcs() (SoftLayer.managers.load\_balancer.LoadBalancerManager  
    method), 41
  - get\_all\_rules() (Soft-  
    Layer.managers.vs\_placement.PlacementManager  
    method), 68
  - get\_available\_routers() (Soft-  
    Layer.managers.vs\_capacity.CapacityManager  
    method), 67
  - get\_backend\_router\_id\_from\_hostname()  
    (SoftLayer.managers.vs\_placement.PlacementManager  
    method), 68
  - get\_bandwidth\_allocation() (Soft-  
    Layer.managers.hardware.HardwareManager  
    method), 32
  - get\_bandwidth\_allocation() (Soft-  
    Layer.managers.vs.VSManager method),  
    62
  - get\_bandwidth\_data() (Soft-  
    Layer.managers.hardware.HardwareManager  
    method), 32
  - get\_bandwidth\_data() (Soft-  
    Layer.managers.vs.VSManager method),  
    62
  - get\_billing\_items() (Soft-  
    Layer.managers.account.AccountManager  
    method), 8
  - get\_block\_volume\_access\_list() (Soft-  
    Layer.managers.block.BlockStorageManager  
    method), 12
  - get\_block\_volume\_details() (Soft-  
    Layer.managers.block.BlockStorageManager  
    method), 12
  - get\_block\_volume\_snapshot\_list() (Soft-  
    Layer.managers.block.BlockStorageManager  
    method), 13
  - get\_cancellation\_reasons() (Soft-  
    Layer.managers.hardware.HardwareManager  
    method), 32
  - get\_cdn() (SoftLayer.managers.cdn.CDNManager  
    method), 16
  - get\_certificate() (Soft-  
    Layer.managers.ssl.SSLManager method),  
    55
  - get\_create\_options() (Soft-  
    Layer.managers.dedicated\_host.DedicatedHostManager  
    method), 18
  - get\_create\_options() (Soft-  
    Layer.managers.hardware.HardwareManager  
    method), 32

<code>get_create_options()</code>	(SoftLayer.managers.vs.VSManager method), 62	<code>Layer.managers.ordering.OrderingManager method), 49</code>
<code>get_create_options()</code>	(SoftLayer.managers.vs_capacity.CapacityManager method), 67	<code>get_key()</code> (SoftLayer.managers.sshkey.SshKeyManager method), 54
<code>get_dedicated_fwl_rules()</code>	(SoftLayer.managers.firewall.FirewallManager method), 30	<code>get_lb()</code> (SoftLayer.managers.load_balancer.LoadBalancerManager method), 42
<code>get_dedicated_package()</code>	(SoftLayer.managers.firewall.FirewallManager method), 30	<code>get_lbaas()</code> (SoftLayer.managers.load_balancer.LoadBalancerManager method), 42
<code>get_event()</code>	(SoftLayer.managers.account.AccountManager method), 8	<code>get_lbaas_uuid_id()</code> (SoftLayer.managers.load_balancer.LoadBalancerManager method), 42
<code>get_event_log_types()</code>	(SoftLayer.managers.event_log.EventLogManager method), 23	<code>get_location_id()</code> (SoftLayer.managers.ordering.OrderingManager method), 50
<code>get_event_logs()</code>	(SoftLayer.managers.event_log.EventLogManager method), 23	<code>get_logs()</code> (SoftLayer.managers.autoscale.AutoScaleManager method), 9
<code>get_event_logs_by_request_id()</code>	(SoftLayer.managers.network.NetworkManager method), 46	<code>get_nas_credentials()</code> (SoftLayer.managers.network.NetworkManager method), 46
<code>get_file_volume_access_list()</code>	(SoftLayer.managers.file.FileStorageManager method), 26	<code>get_object()</code> (SoftLayer.managers.vs_capacity.CapacityManager method), 67
<code>get_file_volume_details()</code>	(SoftLayer.managers.file.FileStorageManager method), 26	<code>get_object()</code> (SoftLayer.managers.vs_placement.PlacementManager method), 69
<code>get_file_volume_snapshot_list()</code>	(SoftLayer.managers.file.FileStorageManager method), 26	<code>get_only_active_packages()</code> (SoftLayer.managers.ordering.OrderingManager static method), 50
<code>get_firewalls()</code>	(SoftLayer.managers.firewall.FirewallManager method), 30	<code>get_order_container()</code> (SoftLayer.managers.ordering.OrderingManager method), 50
<code>get_hardware()</code>	(SoftLayer.managers.hardware.HardwareManager method), 32	<code>get_origins()</code> (SoftLayer.managers.cdn.CDNManager method), 17
<code>get_host()</code>	(SoftLayer.managers.dedicated_host.DedicatedHostManager method), 18	<code>get_package_by_key()</code> (SoftLayer.managers.ordering.OrderingManager method), 50
<code>get_image()</code>	(SoftLayer.managers.image.ImageManager method), 36	<code>get_package_by_type()</code> (SoftLayer.managers.ordering.OrderingManager method), 50
<code>get_instance()</code>	(SoftLayer.managers.vs.VSManager method), 62	<code>get_package_id_by_type()</code> (SoftLayer.managers.ordering.OrderingManager method), 50
<code>get_invoices()</code>	(SoftLayer.managers.account.AccountManager method), 8	<code>get_packages_of_type()</code> (SoftLayer.managers.ordering.OrderingManager method), 50
<code>get_item_capacity()</code>	(SoftLayer.managers.ordering.OrderingManager method), 49	<code>get_policy()</code> (SoftLayer.managers.autoscale.AutoScaleManager method), 9
<code>get_item_price_id()</code>	(SoftLayer.managers.ordering.OrderingManager static method), 49	<code>get_preset_by_key()</code> (SoftLayer.managers.ordering.OrderingManager method), 50
<code>get_item_prices()</code>	(SoftLayer.managers.ordering.OrderingManager method), 49	<code>get_preset_prices()</code> (SoftLayer.managers.ordering.OrderingManager method), 50



<code>method</code> ), 50		<code>get_subnets_in_acl()</code> (Soft- <i>Layer.managers.block.BlockStorageManager</i> <i>method</i> ), 13
<code>get_price_id_list()</code> (Soft- <i>Layer.managers.ordering.OrderingManager</i> <i>method</i> ), 51		<code>get_summary()</code> (Soft- <i>Layer.managers.account.AccountManager</i> <i>method</i> ), 9
<code>get_quote_details()</code> (Soft- <i>Layer.managers.ordering.OrderingManager</i> <i>method</i> ), 51		<code>get_summary_data_usage()</code> (Soft- <i>Layer.managers.vs.VSManager</i> <i>method</i> ), 62
<code>get_quotes()</code> (Soft- <i>Layer.managers.ordering.OrderingManager</i> <i>method</i> ), 51		<code>get_ticket()</code> (Soft- <i>Layer.managers.ticket.TicketManager</i> <i>method</i> ), 57
<code>get_record()</code> (Soft- <i>Layer.managers.dns.DNSManager</i> <i>method</i> ), 22		<code>get_tracking_id()</code> (Soft- <i>Layer.managers.hardware.HardwareManager</i> <i>method</i> ), 33
<code>get_records()</code> (Soft- <i>Layer.managers.dns.DNSManager</i> <i>method</i> ), 22		<code>get_tracking_id()</code> (Soft- <i>Layer.managers.vs.VSManager</i> <i>method</i> ), 63
<code>get_replication_locations()</code> (Soft- <i>Layer.managers.block.BlockStorageManager</i> <i>method</i> ), 13		<code>get_translation()</code> (Soft- <i>Layer.managers.ipsec.IPSECManager</i> <i>method</i> ), 38
<code>get_replication_locations()</code> (Soft- <i>Layer.managers.file.FileStorageManager</i> <i>method</i> ), 26		<code>get_translations()</code> (Soft- <i>Layer.managers.ipsec.IPSECManager</i> <i>method</i> ), 38
<code>get_replication_partners()</code> (Soft- <i>Layer.managers.block.BlockStorageManager</i> <i>method</i> ), 13		<code>get_tunnel_context()</code> (Soft- <i>Layer.managers.ipsec.IPSECManager</i> <i>method</i> ), 38
<code>get_replication_partners()</code> (Soft- <i>Layer.managers.file.FileStorageManager</i> <i>method</i> ), 26		<code>get_tunnel_contexts()</code> (Soft- <i>Layer.managers.ipsec.IPSECManager</i> <i>method</i> ), 39
<code>get_request()</code> ( <i>SoftLayer.BasicAuthentication</i> <i>method</i> ), 73		<code>get_upcoming_events()</code> (Soft- <i>Layer.managers.account.AccountManager</i> <i>method</i> ), 9
<code>get_router_options()</code> (Soft- <i>Layer.managers.dedicated_host.DedicatedHostManager</i> <i>method</i> ), 19		<code>get_usage_metrics()</code> (Soft- <i>Layer.managers.cdn.CDNManager</i> <i>method</i> ), 17
<code>get_routers()</code> (Soft- <i>Layer.managers.vs_placement.PlacementManager</i> <i>method</i> ), 69		<code>get_virtual_guests()</code> (Soft- <i>Layer.managers.autoscale.AutoScaleManager</i> <i>method</i> ), 10
<code>get_rule_id_from_name()</code> (Soft- <i>Layer.managers.vs_placement.PlacementManager</i> <i>method</i> ), 69		<code>get_vlan()</code> ( <i>SoftLayer.managers.network.NetworkManager</i> <i>method</i> ), 47
<code>get_rwhois()</code> (Soft- <i>Layer.managers.network.NetworkManager</i> <i>method</i> ), 46		<code>get_zone()</code> ( <i>SoftLayer.managers.dns.DNSManager</i> <i>method</i> ), 22
<code>get_securitygroup()</code> (Soft- <i>Layer.managers.network.NetworkManager</i> <i>method</i> ), 46		globalip-assign command line option IDENTIFIER, 123 TARGET, 123
<code>get_standard_fwl_rules()</code> (Soft- <i>Layer.managers.firewall.FirewallManager</i> <i>method</i> ), 30		globalip-cancel command line option IDENTIFIER, 123
<code>get_standard_package()</code> (Soft- <i>Layer.managers.firewall.FirewallManager</i> <i>method</i> ), 30		globalip-create command line option -test <test>, 123 -v6, -ipv6, 123
<code>get_subnet()</code> (Soft- <i>Layer.managers.network.NetworkManager</i> <i>method</i> ), 47		globalip-list command line option -ip-version <ip_version>, 123
		globalip-unassign command line option

- IDENTIFIER, 124
- GROUP\_ID  
securitygroup-edit command line option, 157
- ## H
- HardwareManager (class in *SoftLayer.managers.hardware*), 30
- has\_firewall() (in *module SoftLayer.managers.firewall*), 30
- hw-bandwidth command line option  
-e, -end\_date <end\_date>, 124  
-p, -summary\_period <summary\_period>, 124  
-q, -quite\_summary, 124  
-s, -start\_date <start\_date>, 124  
IDENTIFIER, 124
- hw-cancel command line option  
-comment <comment>, 125  
-immediate, 125  
-reason <reason>, 125  
IDENTIFIER, 125
- hw-create command line option  
-billing <billing>, 126  
-export <export>, 126  
-no-public, 126  
-port-speed <port\_speed>, 126  
-test, 126  
-wait <wait>, 126  
-D, -domain <domain>, 125  
-H, -hostname <hostname>, 125  
-d, -datacenter <datacenter>, 126  
-e, -extra <extra>, 126  
-i, -postinstall <postinstall>, 126  
-k, -key <key>, 126  
-o, -os <os>, 125  
-s, -size <size>, 125  
-t, -template <template>, 126
- hw-credentials command line option  
IDENTIFIER, 126
- hw-detail command line option  
-passwords, 127  
-price, 127  
IDENTIFIER, 127
- hw-dns-sync command line option  
-aaaa-record, 131  
-ptr, 131  
-ttl <ttl>, 131  
-a, -a-record, 131  
IDENTIFIER, 131
- hw-edit command line option  
-private-speed <private\_speed>, 127  
-public-speed <public\_speed>, 127  
-D, -domain <domain>, 127  
-F, -userfile <userfile>, 127  
-H, -hostname <hostname>, 127  
-g, -tag <tag>, 127  
-u, -userdata <userdata>, 127  
IDENTIFIER, 127
- hw-list command line option  
-columns <columns>, 128  
-sortby <sortby>, 128  
-tag <tag>, 128  
-D, -domain <domain>, 128  
-H, -hostname <hostname>, 128  
-c, -cpu <cpu>, 128  
-d, -datacenter <datacenter>, 128  
-l, -limit <limit>, 128  
-m, -memory <memory>, 128  
-n, -network <network>, 128
- hw-power-cycle command line option  
IDENTIFIER, 128
- hw-power-off command line option  
IDENTIFIER, 129
- hw-power-on command line option  
IDENTIFIER, 129
- hw-ready command line option  
-wait <wait>, 131  
IDENTIFIER, 131
- hw-reboot command line option  
-hard, -soft, 129  
IDENTIFIER, 129
- hw-reflash-firmware command line option  
IDENTIFIER, 130
- hw-reload command line option  
-i, -postinstall <postinstall>, 129  
-k, -key <key>, 129  
IDENTIFIER, 130
- hw-rescue command line option  
IDENTIFIER, 130
- hw-toggle-ipmi command line option  
-enable, -disable, 131  
IDENTIFIER, 131
- hw-update-firmware command line option  
IDENTIFIER, 130
- ## I
- IDENTIFIER  
account-event-detail command line option, 82  
account-invoice-detail command line option, 82  
autoscale-detail command line option, 83  
autoscale-edit command line option, 85



- autoscale-logs command line option, 84
- autoscale-scale command line option, 83
- autoscale-tag command line option, 84
- dedicatedhost-cancel command line option, 105
- dedicatedhost-cancel-guests command line option, 105
- dedicatedhost-detail command line option, 105
- dedicatedhost-list-guests command line option, 106
- firewall-cancel command line option, 122
- firewall-detail command line option, 122
- firewall-edit command line option, 122
- globalip-assign command line option, 123
- globalip-cancel command line option, 123
- globalip-unassign command line option, 124
- hw-bandwidth command line option, 124
- hw-cancel command line option, 125
- hw-credentials command line option, 126
- hw-detail command line option, 127
- hw-dns-sync command line option, 131
- hw-edit command line option, 127
- hw-power-cycle command line option, 128
- hw-power-off command line option, 129
- hw-power-on command line option, 129
- hw-ready command line option, 131
- hw-reboot command line option, 129
- hw-reflash-firmware command line option, 130
- hw-reload command line option, 130
- hw-rescue command line option, 130
- hw-toggle-ipmi command line option, 131
- hw-update-firmware command line option, 130
- image-delete command line option, 132
- image-detail command line option, 132
- image-edit command line option, 132
- image-export command line option, 134
- loadbal-cancel command line option, 144
- loadbal-detail command line option, 139
- loadbal-health command line option, 139
- loadbal-l7pool-add command line option, 143
- loadbal-l7pool-del command line option, 143
- loadbal-member-add command line option, 140
- loadbal-member-remove command line option, 140
- loadbal-ns-detail command line option, 145
- loadbal-pool-add command line option, 141
- loadbal-pool-delete command line option, 142
- loadbal-pool-edit command line option, 142
- object-storage-credential-create command line option, 146
- object-storage-credential-delete command line option, 146
- object-storage-credential-limit command line option, 146
- object-storage-credential-list command line option, 145
- securitygroup-detail command line option, 156
- sshkey-edit command line option, 162
- sshkey-print command line option, 162
- sshkey-remove command line option, 161
- ssl-download command line option, 163
- ssl-edit command line option, 164
- ssl-remove command line option, 164
- subnet-cancel command line option, 164
- subnet-detail command line option, 166
- ticket-attach command line option, 169
- ticket-detach command line option, 169
- ticket-detail command line option, 167
- ticket-update command line option,

- 168
  - ticket-upload command line option, 168
  - user-delete command line option, 173
  - user-detail command line option, 170
  - user-edit-permissions command line option, 171
  - user-permissions command line option, 171
  - vlan-detail command line option, 173
  - vs-bandwidth command line option, 178
  - vs-cancel command line option, 178
  - vs-capture command line option, 178
  - vs-dns-sync command line option, 181
  - vs-edit command line option, 181
  - vs-pause command line option, 183
  - vs-power-off command line option, 183
  - vs-power-on command line option, 183
  - vs-ready command line option, 184
  - vs-reboot command line option, 184
  - vs-rescue command line option, 184
  - vs-resume command line option, 183
  - vs-upgrade command line option, 185
  - vs-usage command line option, 185
  - image-delete command line option IDENTIFIER, 132
  - image-detail command line option IDENTIFIER, 132
  - image-edit command line option
    - name <name>, 132
    - note <note>, 132
    - tag <tag>, 132
    - IDENTIFIER, 132
  - image-export command line option
    - ibm-api-key <ibm\_api\_key>, 134
    - IDENTIFIER, 134
    - URI, 134
  - image-import command line option
    - byol, 133
    - cloud-init, 133
    - ibm-api-key <ibm\_api\_key>, 133
    - is-encrypted, 133
    - note <note>, 133
    - os-code <os\_code>, 133
    - root-key-crn <root\_key\_crn>, 133
    - wrapped-dek <wrapped\_dek>, 133
    - NAME, 134
    - URI, 134
  - image-list command line option
    - name <name>, 133
    - public, -private, 133
  - ImageManager (class in *SoftLayer.managers.image*), 35
  - import\_image\_from\_uri() (*SoftLayer.managers.image.ImageManager* method), 36
  - IP\_ADDRESS
    - subnet-lookup command line option, 166
  - ip\_lookup() (*SoftLayer.managers.network.NetworkManager* method), 47
  - IPSECManger (class in *SoftLayer.managers.ipsec*), 37
  - iter\_call() (*SoftLayer.BaseClient* method), 72
- ## L
- LABEL
    - sshkey-add command line option, 161
  - lbaas\_order\_options() (*SoftLayer.managers.load\_balancer.LoadBalancerManager* method), 42
  - list() (*SoftLayer.managers.autoscale.AutoScaleManager* method), 10
  - list() (*SoftLayer.managers.vs\_capacity.CapacityManager* method), 68
  - list() (*SoftLayer.managers.vs\_placement.PlacementManager* method), 69
  - list\_block\_volume\_limit() (*SoftLayer.managers.block.BlockStorageManager* method), 13
  - list\_block\_volumes() (*SoftLayer.managers.block.BlockStorageManager* method), 13
  - list\_categories() (*SoftLayer.managers.ordering.OrderingManager* method), 51
  - list\_cdn() (*SoftLayer.managers.cdn.CDNManager* method), 17
  - list\_certs() (*SoftLayer.managers.ssl.SSLManager* method), 56
  - list\_file\_volume\_limit() (*SoftLayer.managers.file.FileStorageManager* method), 26
  - list\_file\_volumes() (*SoftLayer.managers.file.FileStorageManager* method), 27
  - list\_global\_ips() (*SoftLayer.managers.network.NetworkManager* method), 47
  - list\_guests() (*SoftLayer.managers.dedicated\_host.DedicatedHostManager* method), 19
  - list\_hardware() (*SoftLayer.managers.hardware.HardwareManager* method), 33

<code>list_instances()</code>	(SoftLayer.managers.dedicated_host.DedicatedHostManager method), 19	option, 142
<code>list_instances()</code>	(SoftLayer.managers.vs.VSManager method), 63	loadbal-cancel command line option IDENTIFIER, 144
<code>list_items()</code>	(SoftLayer.managers.ordering.OrderingManager method), 51	loadbal-detail command line option IDENTIFIER, 139
<code>list_keys()</code>	(SoftLayer.managers.sshkey.SshKeyManager method), 54	loadbal-health command line option -uuid <uuid>, 139
<code>list_packages()</code>	(SoftLayer.managers.ordering.OrderingManager method), 51	-i, -interval <interval>, 139
<code>list_presets()</code>	(SoftLayer.managers.ordering.OrderingManager method), 51	-r, -retry <retry>, 139
<code>list_private_images()</code>	(SoftLayer.managers.image.ImageManager method), 36	-t, -timeout <timeout>, 139
<code>list_public_images()</code>	(SoftLayer.managers.image.ImageManager method), 36	-u, -url <url>, 139
<code>list_securitygroup_rules()</code>	(SoftLayer.managers.network.NetworkManager method), 47	IDENTIFIER, 139
<code>list_securitygroups()</code>	(SoftLayer.managers.network.NetworkManager method), 47	loadbal-l7pool-add command line option -healthInterval <healthinterval>, 143
<code>list_subjects()</code>	(SoftLayer.managers.ticket.TicketManager method), 57	-healthPath <healthpath>, 143
<code>list_subnets()</code>	(SoftLayer.managers.network.NetworkManager method), 47	-healthRetry <healthretry>, 143
<code>list_tickets()</code>	(SoftLayer.managers.ticket.TicketManager method), 57	-healthTimeout <healthtimeout>, 143
<code>list_vlans()</code>	(SoftLayer.managers.network.NetworkManager method), 47	-P, -protocol <protocol>, 143
<code>list_volume_schedules()</code>	(SoftLayer.managers.block.BlockStorageManager method), 13	-S, -server <server>, 143
<code>list_volume_schedules()</code>	(SoftLayer.managers.file.FileStorageManager method), 27	-m, -method <method>, 142
<code>list_zones()</code>	(SoftLayer.managers.dns.DNSManager method), 22	-n, -name <name>, 142
LISTENER		-s, -sticky, 143
loadbal-pool-delete command line option, 142		IDENTIFIER, 143
loadbal-pool-edit command line		loadbal-l7pool-del command line option IDENTIFIER, 143
		loadbal-member-add command line option -private, -public, 140
		-m, -member <member>, 140
		-w, -weight <weight>, 140
		IDENTIFIER, 140
		loadbal-member-remove command line option
		-m, -member <member>, 140
		IDENTIFIER, 140
		loadbal-ns-detail command line option IDENTIFIER, 145
		loadbal-order command line option
		-public, 144
		-verify, 144
		-b, -backend <backend>, 144
		-d, -datacenter <datacenter>, 144
		-f, -frontend <frontend>, 144
		-l, -label <label>, 144
		-m, -method <method>, 144
		-n, -name <name>, 144
		-s, -subnet <subnet>, 144
		loadbal-order-options command line option
		-d, -datacenter <datacenter>, 144
		loadbal-pool-add command line option
		-P, -frontProtocol <frontprotocol>, 141
		-b, -backPort <backport>, 141

-c, -connections <connections>, 141  
 -f, -frontPort <frontport>, 141  
 -m, -method <method>, 141  
 -p, -backProtocol <backprotocol>, 141  
 -s, -sticky, 141  
 -x, -sslCert <sslcert>, 141  
 IDENTIFIER, 141  
 loadbal-pool-delete command line option  
 IDENTIFIER, 142  
 LISTENER, 142  
 loadbal-pool-edit command line option  
 -P, -frontProtocol <frontprotocol>, 141  
 -b, -backPort <backport>, 141  
 -c, -connections <connections>, 142  
 -f, -frontPort <frontport>, 141  
 -m, -method <method>, 142  
 -p, -backProtocol <backprotocol>, 141  
 -s, -sticky, 142  
 -x, -sslCert <sslcert>, 142  
 IDENTIFIER, 142  
 LISTENER, 142  
 LoadBalancerManager (class in *SoftLayer.managers.load\_balancer*), 40  
 LOCATION  
 order-place command line option, 151  
 order-place-quote command line option, 154  
 LUN\_ID  
 block-volume-set-lun-id command line option, 96  
**M**  
 metadata command line option  
 PROP, 102  
 METADATA\_ATTRIBUTES (SoftLayer.managers.metadata attribute), 44  
 MetadataManager (class in *SoftLayer.managers.metadata*), 43  
 METHOD  
 call-api command line option, 101  
**N**  
 NAME  
 image-import command line option, 134  
 NETWORK  
 subnet-create command line option, 165  
 NetworkManager (class in *SoftLayer.managers.network*), 44

**O**

object-storage-credential-create command line option  
 IDENTIFIER, 146  
 object-storage-credential-delete command line option  
 -c, -credential\_id <credential\_id>, 146  
 IDENTIFIER, 146  
 object-storage-credential-limit command line option  
 IDENTIFIER, 146  
 object-storage-credential-list command line option  
 IDENTIFIER, 145  
 order-category-list command line option  
 -required, 148  
 PACKAGE\_KEYNAME, 148  
 order-item-list command line option  
 -category <category>, 149  
 -keyword <keyword>, 149  
 PACKAGE\_KEYNAME, 149  
 order-package-list command line option  
 -keyword <keyword>, 147  
 -package\_type <package\_type>, 147  
 order-package-locations command line option  
 PACKAGE\_KEYNAME, 147  
 order-place command line option  
 -billing <billing>, 150  
 -complex-type <complex\_type>, 150  
 -extras <extras>, 150  
 -preset <preset>, 150  
 -quantity <quantity>, 150  
 -verify, 150  
 LOCATION, 151  
 ORDER\_ITEMS, 151  
 PACKAGE\_KEYNAME, 151  
 order-place-quote command line option  
 -complex-type <complex\_type>, 153  
 -extras <extras>, 153  
 -name <name>, 153  
 -preset <preset>, 153  
 -send-email, 153  
 LOCATION, 154  
 ORDER\_ITEMS, 154  
 PACKAGE\_KEYNAME, 153  
 order-preset-list command line option  
 -keyword <keyword>, 149  
 PACKAGE\_KEYNAME, 149  
 order-quote command line option  
 -complex-type <complex\_type>, 152  
 -fqdn <fqdn>, 152

- image <image>, 152
- quantity <quantity>, 152
- verify, 152
- F, -userfile <userfile>, 152
- i, -postinstall <postinstall>, 152
- k, -key <key>, 152
- u, -userdata <userdata>, 152
- QUOTE, 152
- order-quote-detail command line option  
QUOTE, 152
- order\_block\_volume() (Soft-  
Layer.managers.block.BlockStorageManager  
method), 14
- order\_duplicate\_volume() (Soft-  
Layer.managers.block.BlockStorageManager  
method), 14
- order\_duplicate\_volume() (Soft-  
Layer.managers.file.FileStorageManager  
method), 27
- order\_file\_volume() (Soft-  
Layer.managers.file.FileStorageManager  
method), 27
- order\_guest() (SoftLayer.managers.vs.VSManager  
method), 64
- ORDER\_ITEMS
  - order-place command line option, 151
  - order-place-quote command line  
option, 154
- order\_lbaas() (Soft-  
Layer.managers.load\_balancer.LoadBalancerManager  
method), 42
- order\_modified\_volume() (Soft-  
Layer.managers.block.BlockStorageManager  
method), 14
- order\_modified\_volume() (Soft-  
Layer.managers.file.FileStorageManager  
method), 28
- order\_quote() (Soft-  
Layer.managers.ordering.OrderingManager  
method), 51
- order\_replicant\_volume() (Soft-  
Layer.managers.block.BlockStorageManager  
method), 15
- order\_replicant\_volume() (Soft-  
Layer.managers.file.FileStorageManager  
method), 28
- order\_snapshot\_space() (Soft-  
Layer.managers.block.BlockStorageManager  
method), 15
- order\_snapshot\_space() (Soft-  
Layer.managers.file.FileStorageManager  
method), 28
- OrderingManager (class in Soft-  
Layer.managers.ordering), 49
- ORIGIN
  - cdn-origin-add command line option,  
99
- ORIGIN\_PATH
  - cdn-origin-remove command line  
option, 100
- ORIGIN\_VOLUME\_ID
  - block-volume-duplicate command  
line option, 94
  - file-volume-duplicate command line  
option, 119
- P**
- PACKAGE\_KEYNAME
  - order-category-list command line  
option, 148
  - order-item-list command line  
option, 149
  - order-package-locations command  
line option, 147
  - order-place command line option, 151
  - order-place-quote command line  
option, 153
  - order-preset-list command line  
option, 149
- package\_locations() (Soft-  
Layer.managers.ordering.OrderingManager  
method), 52
- PARAMETERS
  - call-api command line option, 101
- PATH
  - cdn-origin-add command line option,  
99
  - cdn-purge command line option, 100
- place\_order() (Soft-  
Layer.managers.dedicated\_host.DedicatedHostManager  
method), 20
- place\_order() (Soft-  
Layer.managers.hardware.HardwareManager  
method), 33
- place\_order() (Soft-  
Layer.managers.ordering.OrderingManager  
method), 52
- place\_quote() (Soft-  
Layer.managers.ordering.OrderingManager  
method), 52
- PlacementManager (class in Soft-  
Layer.managers.vs\_placement), 68
- private\_network() (Soft-  
Layer.managers.metadata.MetadataManager  
method), 43
- PROP
  - metadata command line option, 102

- `public_network()` (*SoftLayer.managers.metadata.MetadataManager method*), 43
- `purge_content()` (*SoftLayer.managers.cdn.CDNManager method*), 17
- Python Enhancement Proposals  
PEP 8, 193
- ## Q
- QUANTITY  
subnet-create command line option, 165
- QUOTE  
order-quote command line option, 152  
order-quote-detail command line option, 152
- ## R
- RECORD  
dns-record-add command line option, 107
- RECORD\_ID  
dns-record-remove command line option, 108
- RECORD\_TYPE  
dns-record-add command line option, 107
- `reflash_firmware()` (*SoftLayer.managers.hardware.HardwareManager method*), 34
- `reload()` (*SoftLayer.managers.hardware.HardwareManager method*), 34
- `reload_instance()` (*SoftLayer.managers.vs.VSManager method*), 64
- `remove_certificate()` (*SoftLayer.managers.ssl.SSLManager method*), 56
- `remove_internal_subnet()` (*SoftLayer.managers.ipsec.IPSECManager method*), 39
- `remove_lb_listener()` (*SoftLayer.managers.load\_balancer.LoadBalancerManager method*), 42
- `remove_origin()` (*SoftLayer.managers.cdn.CDNManager method*), 17
- `remove_remote_subnet()` (*SoftLayer.managers.ipsec.IPSECManager method*), 39
- `remove_securitygroup_rule()` (*SoftLayer.managers.network.NetworkManager method*), 48
- `remove_securitygroup_rules()` (*SoftLayer.managers.network.NetworkManager method*), 48
- `remove_service_subnet()` (*SoftLayer.managers.ipsec.IPSECManager method*), 39
- `remove_subnets_from_acl()` (*SoftLayer.managers.block.BlockStorageManager method*), 15
- `remove_translation()` (*SoftLayer.managers.ipsec.IPSECManager method*), 39
- report-bandwidth command line option  
-end <end>, 154  
-sortby <sortby>, 154  
-start <start>, 154
- REQUEST\_ID  
securitygroup-event-log command line option, 161
- `rescue()` (*SoftLayer.managers.hardware.HardwareManager method*), 34
- `rescue()` (*SoftLayer.managers.vs.VSManager method*), 64
- `resolve_global_ip_ids()` (*SoftLayer.managers.network.NetworkManager method*), 48
- `resolve_ids()` (*SoftLayer.managers.account.AccountManager method*), 9
- `resolve_ids()` (*SoftLayer.managers.block.BlockStorageManager method*), 15
- `resolve_ids()` (*SoftLayer.managers.cdn.CDNManager method*), 17
- `resolve_ids()` (*SoftLayer.managers.dedicated\_host.DedicatedHostManager method*), 20
- `resolve_ids()` (*SoftLayer.managers.dns.DNSManager method*), 22
- `resolve_ids()` (*SoftLayer.managers.file.FileStorageManager method*), 28
- `resolve_ids()` (*SoftLayer.managers.firewall.FirewallManager method*), 30
- `resolve_ids()` (*SoftLayer.managers.hardware.HardwareManager method*), 34
- `resolve_ids()` (*SoftLayer.managers.image.ImageManager method*), 37
- `resolve_ids()` (*Soft-*



<i>Layer.managers.ipsec.IPSECManager</i> method), 39	securitygroup-create command line option
resolve_ids() ( <i>Soft-</i> <i>Layer.managers.load_balancer.LoadBalancerManager</i> method), 42	-d, -description <description>, 156 -n, -name <name>, 156
resolve_ids() ( <i>Soft-</i> <i>Layer.managers.sshkey.SshKeyManager</i> method), 54	securitygroup-delete command line option SECURITYGROUP_ID, 157
resolve_ids() ( <i>Soft-</i> <i>Layer.managers.ticket.TicketManager</i> method), 57	securitygroup-detail command line option IDENTIFIER, 156
resolve_ids() ( <i>SoftLayer.managers.vs.VSManager</i> method), 65	securitygroup-edit command line option -d, -description <description>, 156 -n, -name <name>, 156
resolve_ids() ( <i>Soft-</i> <i>Layer.managers.vs_capacity.CapacityManager</i> method), 68	GROUP_ID, 157 securitygroup-event-log command line option
resolve_ids() ( <i>Soft-</i> <i>Layer.managers.vs_placement.PlacementManager</i> method), 69	REQUEST_ID, 161 securitygroup-interface-add command line option
resolve_subnet_ids() ( <i>Soft-</i> <i>Layer.managers.network.NetworkManager</i> method), 48	-i, -interface <interface>, 160 -n, -network-component <network_component>, 160
resolve_vlan_ids() ( <i>Soft-</i> <i>Layer.managers.network.NetworkManager</i> method), 48	-s, -server <server>, 160 SECURITYGROUP_ID, 160
restore_from_snapshot() ( <i>Soft-</i> <i>Layer.managers.block.BlockStorageManager</i> method), 15	securitygroup-interface-list command line option -sortby <sortby>, 159 SECURITYGROUP_ID, 159
restore_from_snapshot() ( <i>Soft-</i> <i>Layer.managers.file.FileStorageManager</i> method), 28	securitygroup-interface-remove command line option -i, -interface <interface>, 160 -n, -network-component <network_component>, 160
RULE_ID	-s, -server <server>, 160 SECURITYGROUP_ID, 160
securitygroup-rule-edit command line option, 159	SECURITYGROUP_ID, 160
securitygroup-rule-remove command line option, 159	securitygroup-list command line option -sortby <sortby>, 156 -l, -limit <limit>, 156
rwhois-edit command line option	securitygroup-rule-add command line option -M, -port-max <port_max>, 158 -d, -direction <direction>, 158 -e, -ethertype <ethertype>, 158 -m, -port-min <port_min>, 158 -p, -protocol <protocol>, 158 -r, -remote-ip <remote_ip>, 158 -s, -remote-group <remote_group>, 158 SECURITYGROUP_ID, 158
-abuse <abuse>, 155	securitygroup-rule-edit command line option
-address1 <address1>, 155	-M, -port-max <port_max>, 159
-address2 <address2>, 155	-d, -direction <direction>, 158
-city <city>, 155	-e, -ethertype <ethertype>, 158
-company <company>, 155	-m, -port-min <port_min>, 158
-country <country>, 155	-p, -protocol <protocol>, 158
-firstname <firstname>, 155	-r, -remote-ip <remote_ip>, 158
-lastname <lastname>, 155	-s, -remote-group <remote_group>, 158 SECURITYGROUP_ID, 158
-postal <postal>, 155	securitygroup-rule-edit command line option
-public, -private, 155	-M, -port-max <port_max>, 159
-state <state>, 155	-d, -direction <direction>, 158 -e, -ethertype <ethertype>, 158
<b>S</b>	
scale() ( <i>SoftLayer.managers.autoscale.AutoScaleManager</i> method), 10	
scale_to() ( <i>SoftLayer.managers.autoscale.AutoScaleManager</i> method), 10	

- m, --port-min <port\_min>, 159
- p, --protocol <protocol>, 159
- r, --remote-ip <remote\_ip>, 158
- s, --remote-group <remote\_group>, 158
- RULE\_ID, 159
- SECURITYGROUP\_ID, 159
- securitygroup-rule-list command line option
  - sortby <sortby>, 157
  - SECURITYGROUP\_ID, 157
- securitygroup-rule-remove command line option
  - RULE\_ID, 159
  - SECURITYGROUP\_ID, 159
- SECURITYGROUP\_ID
  - securitygroup-delete command line option, 157
  - securitygroup-interface-add command line option, 160
  - securitygroup-interface-list command line option, 159
  - securitygroup-interface-remove command line option, 160
  - securitygroup-rule-add command line option, 158
  - securitygroup-rule-edit command line option, 159
  - securitygroup-rule-list command line option, 157
  - securitygroup-rule-remove command line option, 159
- SERVICE
  - call-api command line option, 101
- set\_credential\_password() (SoftLayer.managers.block.BlockStorageManager method), 15
- set\_tags() (SoftLayer.managers.vs.VSManager method), 65
- SNAPSHOT\_ID
  - block-snapshot-delete command line option, 90
  - file-snapshot-delete command line option, 115
- SoftLayer (module), 71
- SoftLayer.managers.account (module), 8
- SoftLayer.managers.autoscale (module), 9
- SoftLayer.managers.block (module), 10
- SoftLayer.managers.cdn (module), 16
- SoftLayer.managers.dedicated\_host (module), 18
- SoftLayer.managers.dns (module), 20
- SoftLayer.managers.event\_log (module), 23
- SoftLayer.managers.file (module), 24
- SoftLayer.managers.firewall (module), 29
- SoftLayer.managers.hardware (module), 30
- SoftLayer.managers.image (module), 35
- SoftLayer.managers.ipsec (module), 37
- SoftLayer.managers.load\_balancer (module), 40
- SoftLayer.managers.metadata (module), 43
- SoftLayer.managers.network (module), 44
- SoftLayer.managers.ordering (module), 48
- SoftLayer.managers.sshkey (module), 54
- SoftLayer.managers.ssl (module), 55
- SoftLayer.managers.ticket (module), 56
- SoftLayer.managers.vs (module), 58
- SoftLayer.managers.vs\_capacity (module), 66
- SoftLayer.managers.vs\_placement (module), 68
- SoftLayerAPIError, 73
- SoftLayerError, 73
- SoftLayerListResult (class in SoftLayer), 73
- sshkey-add command line option
  - note <note>, 161
  - f, --in-file <in\_file>, 161
  - k, --key <key>, 161
  - LABEL, 161
- sshkey-edit command line option
  - note <note>, 162
  - k, --label <label>, 162
  - IDENTIFIER, 162
- sshkey-list command line option
  - sortby <sortby>, 162
- sshkey-print command line option
  - f, --out-file <out\_file>, 162
  - IDENTIFIER, 162
- sshkey-remove command line option
  - IDENTIFIER, 161
- SshKeyManager (class in SoftLayer.managers.sshkey), 54
- ssl-add command line option
  - crt <crt>, 163
  - csr <csr>, 163
  - icc <icc>, 163
  - key <key>, 163
  - notes <notes>, 163
- ssl-download command line option
  - IDENTIFIER, 163
- ssl-edit command line option
  - crt <crt>, 163
  - csr <csr>, 163
  - icc <icc>, 163
  - key <key>, 163
  - notes <notes>, 163
  - IDENTIFIER, 164
- ssl-list command line option



- sortby <sortby>, 164
  - status <status>, 164
  - ssl-remove command line option  
IDENTIFIER, 164
  - SSLManager (*class in SoftLayer.managers.ssl*), 55
  - start\_data (*SoftLayer.managers.cdn.CDNManager attribute*), 18
  - subnet-cancel command line option  
IDENTIFIER, 164
  - subnet-create command line option
    - ipv6, -v6, 165
    - test, 165
    - ENDPOINT\_ID, 165
    - NETWORK, 165
    - QUANTITY, 165
  - subnet-detail command line option
    - no-hardware, 166
    - no-vs, 166
    - IDENTIFIER, 166
  - subnet-list command line option
    - identifier <identifier>, 166
    - ipv4, -v4, 166
    - ipv6, -v6, 166
    - network-space <network\_space>, 166
    - sortby <sortby>, 166
    - d, -datacenter <datacenter>, 166
    - t, -subnet-type <subnet\_type>, 166
  - subnet-lookup command line option  
IP\_ADDRESS, 166
  - summary command line option
    - sortby <sortby>, 154
  - summary\_by\_datacenter() (*SoftLayer.managers.network.NetworkManager method*), 48
  - sync\_host\_record() (*SoftLayer.managers.dns.DNSManager method*), 22
  - sync\_ptr\_record() (*SoftLayer.managers.dns.DNSManager method*), 23
- T**
- TARGET
    - firewall-add command line option, 121
    - globalip-assign command line option, 123
  - ticket-attach command line option
    - hardware <hardware\_identifier>, 169
    - virtual <virtual\_identifier>, 169
    - IDENTIFIER, 169
  - ticket-create command line option
    - body <body>, 167
    - hardware <hardware\_identifier>, 167
    - priority <priority>, 167
    - subject-id <subject\_id>, 167
    - title <title>, 167
    - virtual <virtual\_identifier>, 167
  - ticket-detach command line option
    - hardware <hardware\_identifier>, 169
    - virtual <virtual\_identifier>, 169
    - IDENTIFIER, 169
  - ticket-detail command line option
    - count <count>, 167
    - IDENTIFIER, 167
  - ticket-list command line option
    - open, -closed, 168
  - ticket-update command line option
    - body <body>, 168
    - IDENTIFIER, 168
  - ticket-upload command line option
    - name <name>, 168
    - path <path>, 168
    - IDENTIFIER, 168
  - TicketManager (*class in SoftLayer.managers.ticket*), 56
  - total\_count (*SoftLayer.SoftLayerListResult attribute*), 73
- U**
- unassign\_global\_ip() (*SoftLayer.managers.network.NetworkManager method*), 48
  - UNIQUE\_ID
    - cdn-detail command line option, 98
    - cdn-origin-add command line option, 99
    - cdn-origin-list command line option, 99
    - cdn-origin-remove command line option, 100
    - cdn-purge command line option, 100
  - update\_firmware() (*SoftLayer.managers.hardware.HardwareManager method*), 34
  - update\_lb\_health\_monitors() (*SoftLayer.managers.load\_balancer.LoadBalancerManager method*), 42
  - update\_ticket() (*SoftLayer.managers.ticket.TicketManager method*), 57
  - update\_translation() (*SoftLayer.managers.ipsec.IPSECManger method*), 39
  - update\_tunnel\_context() (*SoftLayer.managers.ipsec.IPSECManger method*), 40

upgrade() (*SoftLayer.managers.vs.VSManager method*), 65  
 upload\_attachment() (*SoftLayer.managers.ticket.TicketManager method*), 58  
 URI  
   image-export command line option, 134  
   image-import command line option, 134  
 USER  
   user-edit-details command line option, 172  
 user-create command line option  
   -a, -api-key, 172  
   -e, -email <email>, 172  
   -p, -password <password>, 172  
   -t, -template <template>, 172  
   -u, -from-user <from\_user>, 172  
   USERNAME, 172  
 user-delete command line option  
   IDENTIFIER, 173  
 user-detail command line option  
   -e, -events, 170  
   -h, -hardware, 170  
   -k, -keys, 170  
   -l, -logins, 170  
   -p, -permissions, 170  
   -v, -virtual, 170  
   IDENTIFIER, 170  
 user-edit-details command line option  
   -t, -template <template>, 171  
   USER, 172  
 user-edit-permissions command line option  
   -enable, -disable, 171  
   -p, -permission <permission>, 171  
   -u, -from-user <from\_user>, 171  
   IDENTIFIER, 171  
 user-list command line option  
   -columns <columns>, 170  
 user-permissions command line option  
   IDENTIFIER, 171  
 USERNAME  
   user-create command line option, 172  
**V**  
 verify\_create\_instance() (*SoftLayer.managers.vs.VSManager method*), 65  
 verify\_order() (*SoftLayer.managers.dedicated\_host.DedicatedHostManager method*), 20  
 verify\_order() (*SoftLayer.managers.hardware.HardwareManager method*), 35  
 verify\_order() (*SoftLayer.managers.ordering.OrderingManager method*), 53  
 verify\_quote() (*SoftLayer.managers.ordering.OrderingManager method*), 53  
 vlan-detail command line option  
   -no-hardware, 173  
   -no-vs, 173  
   IDENTIFIER, 173  
 vlan-list command line option  
   -name <name>, 173  
   -sortby <sortby>, 173  
   -d, -datacenter <datacenter>, 173  
   -l, -limit <limit>, 173  
   -n, -number <number>, 173  
 VOLUME\_ID  
   block-access-authorize command line option, 85  
   block-access-list command line option, 86  
   block-access-revoke command line option, 86  
   block-replica-failback command line option, 87  
   block-replica-failover command line option, 87  
   block-replica-locations command line option, 89  
   block-replica-order command line option, 88  
   block-replica-partners command line option, 88  
   block-snapshot-cancel command line option, 89  
   block-snapshot-create command line option, 89  
   block-snapshot-disable command line option, 90  
   block-snapshot-enable command line option, 91  
   block-snapshot-list command line option, 91  
   block-snapshot-order command line option, 92  
   block-snapshot-restore command line option, 92  
   block-snapshot-schedule-list command line option, 91  
   block-volume-cancel command line option, 92

- block-volume-detail command line option, 93
- block-volume-modify command line option, 95
- block-volume-set-lun-id command line option, 96
- file-access-authorize command line option, 111
- file-access-list command line option, 111
- file-access-revoke command line option, 112
- file-replica-failback command line option, 112
- file-replica-failover command line option, 113
- file-replica-locations command line option, 114
- file-replica-order command line option, 113
- file-replica-partners command line option, 113
- file-snapshot-cancel command line option, 114
- file-snapshot-create command line option, 115
- file-snapshot-disable command line option, 115
- file-snapshot-enable command line option, 116
- file-snapshot-list command line option, 116
- file-snapshot-order command line option, 117
- file-snapshot-restore command line option, 117
- file-snapshot-schedule-list command line option, 121
- file-volume-cancel command line option, 117
- file-volume-detail command line option, 118
- file-volume-modify command line option, 120
- vs-bandwidth command line option
  - e, -end\_date <end\_date>, 177
  - p, -summary\_period <summary\_period>, 177
  - q, -quite\_summary, 177
  - s, -start\_date <start\_date>, 177
  - IDENTIFIER, 178
- vs-cancel command line option
  - IDENTIFIER, 178
- vs-capture command line option
  - all <all>, 178
  - note <note>, 178
  - n, -name <name>, 178
  - IDENTIFIER, 178
- vs-create command line option
  - billing <billing>, 179
  - boot-mode <boot\_mode>, 179
  - dedicated, -public, 179
  - disk <disk>, 179
  - export <export>, 179
  - host-id <host\_id>, 179
  - image <image>, 179
  - ipv6, 180
  - like <like>, 179
  - placementgroup <placementgroup>, 180
  - private, 179
  - san, 179
  - subnet-private <subnet\_private>, 180
  - subnet-public <subnet\_public>, 180
  - test, 179
  - transient, 180
  - vlan-private <vlan\_private>, 180
  - vlan-public <vlan\_public>, 180
  - wait <wait>, 180
  - D, -domain <domain>, 179
  - F, -userfile <userfile>, 180
  - H, -hostname <hostname>, 179
  - S, -public-security-group <public\_security\_group>, 180
  - c, -cpu <cpu>, 179
  - d, -datacenter <datacenter>, 179
  - f, -flavor <flavor>, 179
  - g, -tag <tag>, 180
  - i, -postinstall <postinstall>, 179
  - k, -key <key>, 179
  - m, -memory <memory>, 179
  - n, -network <network>, 179
  - o, -os <os>, 179
  - s, -private-security-group <private\_security\_group>, 180
  - t, -template <template>, 180
  - u, -userdata <userdata>, 180
- vs-dns-sync command line option
  - aaaa-record, 181
  - ptr, 181
  - ttl <ttl>, 181
  - a, -a-record, 181
  - IDENTIFIER, 181
- vs-edit command line option
  - private-speed <private\_speed>, 181
  - public-speed <public\_speed>, 181
  - D, -domain <domain>, 181

-F, -userfile <userfile>, 181  
 -H, -hostname <hostname>, 181  
 -g, -tag <tag>, 181  
 -u, -userdata <userdata>, 181  
 IDENTIFIER, 181  
 vs-list command line option  
   -columns <columns>, 182  
   -hourly, 182  
   -monthly, 182  
   -sortby <sortby>, 182  
   -tag <tag>, 182  
   -transient <transient>, 182  
   -D, -domain <domain>, 182  
   -H, -hostname <hostname>, 182  
   -c, -cpu <cpu>, 182  
   -d, -datacenter <datacenter>, 182  
   -l, -limit <limit>, 182  
   -m, -memory <memory>, 182  
   -n, -network <network>, 182  
 vs-pause command line option  
   IDENTIFIER, 183  
 vs-power-off command line option  
   -hard, -soft, 183  
   IDENTIFIER, 183  
 vs-power-on command line option  
   IDENTIFIER, 183  
 vs-ready command line option  
   -wait <wait>, 184  
   IDENTIFIER, 184  
 vs-reboot command line option  
   -hard, -soft, 184  
   IDENTIFIER, 184  
 vs-rescue command line option  
   IDENTIFIER, 184  
 vs-resume command line option  
   IDENTIFIER, 183  
 vs-upgrade command line option  
   -cpu <cpu>, 185  
   -flavor <flavor>, 185  
   -memory <memory>, 185  
   -network <network>, 185  
   -private, 185  
   IDENTIFIER, 185  
 vs-usage command line option  
   -e, -end\_date <end\_date>, 185  
   -p, -summary\_period  
     <summary\_period>, 185  
   -s, -start\_date <start\_date>, 185  
   -t, -valid\_type <valid\_type>, 185  
   IDENTIFIER, 185  
 VSManager (class in SoftLayer.managers.vs), 58

## W

wait\_for\_ready() (Soft-

Layer.managers.hardware.HardwareManager  
 method), 35  
 wait\_for\_ready() (Soft-  
 Layer.managers.vs.VSManager  
 method),  
 66  
 wait\_for\_transaction() (Soft-  
 Layer.managers.vs.VSManager  
 method),  
 66

## Z

ZONE  
   dns-record-list command line  
     option, 108  
   dns-zone-create command line  
     option, 109  
   dns-zone-delete command line  
     option, 109  
   dns-zone-print command line option,  
     109  
 ZONE\_ID  
   dns-record-edit command line  
     option, 108  
 ZONEFILE  
   dns-import command line option, 106