
SoftLayer API Python Client Documentation

Release latest

SoftLayer Technologies, Inc.

Jul 06, 2020

1	Installation	3
1.1	What's Included	3
1.2	Using Pip	3
1.3	Debian/Ubuntu	3
1.4	From Source	3
2	Configuration File	5
3	API Documentation	7
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	11
3.2.4	SoftLayer.cdn	17
3.2.5	SoftLayer.dedicatedhost	19
3.2.6	SoftLayer.dns	22
3.2.7	SoftLayer.event_log	24
3.2.8	SoftLayer.file	25
3.2.9	SoftLayer.firewall	31
3.2.10	SoftLayer.hardware	33
3.2.11	SoftLayer.image	38
3.2.12	SoftLayer.ipsec	40
3.2.13	SoftLayer.load_balancer	44
3.2.14	SoftLayer.metadata	46
3.2.15	SoftLayer.network	47
3.2.16	SoftLayer.ordering	53
3.2.17	SoftLayer.sshkey	58
3.2.18	SoftLayer.ssl	59
3.2.19	SoftLayer.ticket	60
3.2.20	SoftLayer.vs	62
3.2.21	SoftLayer.vs_capacity	71
3.2.22	SoftLayer.vs_placement	72
3.3	Making API Calls	73
3.4	Debugging	75
3.5	API Reference	75
3.5.1	SoftLayer Python API Client	75

4	Command-line Interface	79
4.1	Configuration Setup	79
4.2	Usage Examples	80
5	Debugging	83
6	Command Directory	85
6.1	Account Commands	85
6.1.1	account summary	85
6.1.2	account events	85
6.1.3	account event-detail	85
6.1.4	account invoices	86
6.1.5	account invoice-detail	86
6.1.6	account billing-items	86
6.1.7	account item-detail	87
6.1.8	account cancel-item	87
6.2	Autoscale Commands	87
6.2.1	autoscale list	87
6.2.2	autoscale detail	87
6.2.3	autoscale scale	88
6.2.4	autoscale logs	88
6.2.5	autoscale tag	89
6.2.6	autoscale edit	89
6.3	Block Commands	90
6.3.1	block access-authorize	90
6.3.2	block access-list	90
6.3.3	block access-revoke	91
6.3.4	block access-password	91
6.3.5	block replica-failback	91
6.3.6	block replica-failover	92
6.3.7	block replica-order	92
6.3.8	block replica-partners	93
6.3.9	block replica-locations	93
6.3.10	block snapshot-cancel	93
6.3.11	block snapshot-create	94
6.3.12	block snapshot-delete	94
6.3.13	block snapshot-disable	94
6.3.14	block snapshot-enable	95
6.3.15	block snapshot-schedule-list	95
6.3.16	block snapshot-list	96
6.3.17	block snapshot-order	96
6.3.18	block snapshot-restore	96
6.3.19	block volume-cancel	97
6.3.20	block volume-count	97
6.3.21	block volume-detail	97
6.3.22	block volume-duplicate	98
6.3.23	block volume-list	99
6.3.24	block volume-modify	99
6.3.25	block volume-order	100
6.3.26	block volume-set-lun-id	101
6.3.27	block volume-limits	101
6.3.28	block subnets-list	101
6.3.29	block subnets-assign	102
6.3.30	block subnets-remove	102

6.3.31	block volume-refresh	102
6.3.32	block volume-convert	103
6.4	Interacting with CDN	103
6.4.1	cdn detail	103
6.4.2	cdn list	103
6.4.3	cdn origin-add	104
6.4.4	cdn origin-list	105
6.4.5	cdn origin-remove	105
6.4.6	cdn purge	105
6.5	Call API	105
6.5.1	call-api	106
6.6	Shell	107
6.7	MetaData	107
6.7.1	metadata	107
6.8	Config	107
6.8.1	config setup	108
6.8.2	config show	108
6.8.3	setup	108
6.9	Dedicated Host Commands	108
6.9.1	dedicatedhost list	108
6.9.2	dedicatedhost create	109
6.9.3	dedicatedhost create-options	109
6.9.4	dedicatedhost detail	110
6.9.5	dedicatedhost cancel	110
6.9.6	dedicatedhost cancel-guests	110
6.9.7	dedicatedhost list-guests	110
6.10	DNS Management	111
6.10.1	dns import	111
6.10.2	dns record-add	111
6.10.3	dns record-edit	113
6.10.4	dns record-list	113
6.10.5	dns record-remove	114
6.10.6	dns zone-create	114
6.10.7	dns zone-delete	114
6.10.8	dns zone-list	114
6.10.9	dns zone-print	114
6.11	Event-Log Commands	115
6.11.1	event-log get	115
6.11.2	event-log types	115
6.12	File Commands	116
6.12.1	file access-authorize	116
6.12.2	file access-list	116
6.12.3	file access-revoke	117
6.12.4	file replica-failback	117
6.12.5	file replica-failover	118
6.12.6	file replica-order	118
6.12.7	file replica-partners	119
6.12.8	file replica-locations	119
6.12.9	file snapshot-cancel	119
6.12.10	file snapshot-create	120
6.12.11	file snapshot-delete	120
6.12.12	file snapshot-disable	120
6.12.13	file snapshot-enable	121
6.12.14	file snapshot-list	121

6.12.15	file snapshot-order	122
6.12.16	file snapshot-restore	122
6.12.17	file volume-cancel	123
6.12.18	file volume-count	123
6.12.19	file volume-detail	123
6.12.20	file volume-duplicate	123
6.12.21	file volume-list	124
6.12.22	file volume-modify	125
6.12.23	file volume-order	125
6.12.24	file volume-limits	126
6.12.25	file snapshot-schedule-list	126
6.12.26	file volume-refresh	127
6.12.27	file volume-convert	127
6.13	Firewall Management	127
6.13.1	firewall add	127
6.13.2	firewall cancel	128
6.13.3	firewall detail	128
6.13.4	firewall edit	128
6.13.5	firewall list	128
6.14	Global IP Addresses	129
6.14.1	globalip assign	129
6.14.2	globalip cancel	129
6.14.3	globalip create	129
6.14.4	globalip list	129
6.14.5	globalip unassign	130
6.15	Interacting with Hardware	130
6.15.1	hardware bandwidth	130
6.15.2	hardware cancel-reasons	131
6.15.3	hardware cancel	131
6.15.4	hardware create-options	131
6.15.5	hardware create	131
6.15.6	hardware credentials	132
6.15.7	hardware detail	133
6.15.8	hardware billing	133
6.15.9	hardware edit	133
6.15.10	hardware list	134
6.15.11	hardware power-cycle	135
6.15.12	hardware power-off	135
6.15.13	hardware power-on	135
6.15.14	hardware reboot	136
6.15.15	hardware reload	136
6.15.16	hardware rescue	136
6.15.17	hardware reflash-firmware	137
6.15.18	hardware update-firmware	137
6.15.19	hardware toggle-ipmi	137
6.15.20	hardware ready	138
6.15.21	hardware dns-sync	138
6.15.22	hardware storage	138
6.16	Disk Image Commands	139
6.16.1	image delete	139
6.16.2	image detail	139
6.16.3	image edit	139
6.16.4	image list	140
6.16.5	image import	140

6.16.6	image export	141
6.16.7	image datacenter	141
6.17	Interacting with IPSEC Tunnels	142
6.17.1	ipsec list	142
6.17.2	ipsec detail	142
6.17.3	ipsec update	144
6.17.4	ipsec configure	146
6.17.5	ipsec subnet-add	147
6.17.6	ipsec subnet-remove	148
6.17.7	ipsec translation-add	148
6.17.8	ipsec translation-remove	149
6.17.9	ipsec translation-update	150
6.18	LoadBalancers	150
6.18.1	LBaaS Commands	150
6.18.2	NetScaler Commands	156
6.19	NAS Commands	157
6.19.1	nas list	157
6.19.2	nas credentials	157
6.20	Object Storage Commands	157
6.20.1	object-storage accounts	157
6.20.2	object-storage endpoints	157
6.20.3	object-storage credential list	158
6.20.4	object-storage credential limit	158
6.20.5	object-storage credential delete	158
6.20.6	object-storage credential create	158
6.21	Ordering	159
6.21.1	order package-list	159
6.21.2	order package-locations	160
6.21.3	order category-list	160
6.21.4	order item-list	161
6.21.5	order preset-list	161
6.21.6	order place	162
6.21.7	order place <preset>	163
6.21.8	order place <Virtual Server>	163
6.22	Quotes	164
6.22.1	order quote	164
6.22.2	order quote-list	165
6.22.3	order quote-detail	165
6.22.4	order place-quote	165
6.23	Reports	166
6.23.1	summary	166
6.23.2	report bandwidth	167
6.24	Reverse Whois Commands	167
6.24.1	rwhois edit	167
6.24.2	rwhois show	168
6.25	Security Groups	168
6.25.1	securitygroup list	168
6.25.2	securitygroup detail	168
6.25.3	securitygroup create	168
6.25.4	securitygroup edit	169
6.25.5	securitygroup delete	169
6.25.6	securitygroup rule-list	169
6.25.7	securitygroup rule-add	170
6.25.8	securitygroup rule-edit	171

6.25.9	securitygroup rule-remove	171
6.25.10	securitygroup interface-list	172
6.25.11	securitygroup interface-add	172
6.25.12	securitygroup interface-remove	172
6.25.13	securitygroup event-log	173
6.26	SSH Keys	173
6.26.1	sshkey add	173
6.26.2	sshkey remove	174
6.26.3	sshkey edit	174
6.26.4	sshkey list	174
6.26.5	sshkey print	174
6.27	SSL Certificates	175
6.27.1	ssl add	175
6.27.2	ssl download	175
6.27.3	ssl edit	176
6.27.4	ssl list	176
6.27.5	ssl remove	176
6.28	Subnets	177
6.28.1	subnet cancel	177
6.28.2	subnet create	177
6.28.3	subnet detail	178
6.28.4	subnet edit	178
6.28.5	subnet list	179
6.28.6	subnet lookup	179
6.28.7	subnet edit-ip	179
6.29	Tag Commands	180
6.29.1	tags list	180
6.29.2	tags set	180
6.29.3	tags details	180
6.29.4	tags delete	181
6.29.5	tags taggable	181
6.29.6	tags cleanup	181
6.30	Support Tickets	182
6.30.1	ticket create	182
6.30.2	ticket detail	182
6.30.3	ticket list	182
6.30.4	ticket update	183
6.30.5	ticket upload	183
6.30.6	ticket subjects	183
6.30.7	ticket summary	184
6.30.8	ticket attach	184
6.30.9	ticket detach	184
6.31	Users	185
6.31.1	user list	185
6.31.2	user detail	185
6.31.3	user permissions	186
6.31.4	user edit-permissions	186
6.31.5	user edit-details	186
6.31.6	user create	187
6.31.7	user delete	187
6.31.8	user vpn-manual	188
6.31.9	user vpn-subnet	188
6.32	VLANs	188
6.32.1	vlan detail	188

6.32.2	vlan edit	189
6.32.3	vlan list	189
6.33	Working with Virtual Servers	190
6.33.1	virtual bandwidth	193
6.33.2	virtual cancel	194
6.33.3	virtual capture	194
6.33.4	virtual create	195
6.33.5	virtual create-options	196
6.33.6	virtual dns-sync	197
6.33.7	virtual edit	197
6.33.8	virtual list	198
6.33.9	virtual pause	199
6.33.10	virtual power-on	199
6.33.11	virtual power-off	199
6.33.12	virtual resume	199
6.33.13	virtual rescue	200
6.33.14	virtual reboot	200
6.33.15	virtual ready	200
6.33.16	virtual upgrade	201
6.33.17	virtual usage	201
6.33.18	virtual storage	202
6.33.19	virtual billing	202
6.33.20	virtual detail	202
6.33.21	virtual reload	202
6.33.22	virtual credentials	203
6.33.23	Reserved Capacity	203
6.33.24	Placement Groups	204
7	Contributing	209
7.1	Contribution Guide	209
7.1.1	Code Organization	209
7.1.2	Setting Up A Dev Environment	209
7.1.3	Testing	210
7.1.4	Documentation	211
7.1.5	Style	211
7.1.6	Contributing	211
7.1.7	Developer Resources	211
7.2	Command-Line Interface Developer Guide	211
7.2.1	First Example	211
7.2.2	Arguments	212
7.2.3	Accessing the API	213
7.2.4	Aborting execution	214
7.2.5	Documenting Commands	214
7.2.6	Architecture	214
8	External Links	215
	Python Module Index	217
	Index	219

[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.

static add_event_filter (*_filter*, *event_type*)

Add data to the object filter.

Param *_filter*: event filter.

Param string *event_type*: event type.

cancel_item (*identifier*, *reason*='No longer needed', *note*=None)

Cancels a specific billing item with a reason

Parameters

- **identifier** (*int*) – Billing_Item id
- **reason** (*string*) – A cancellation reason
- **note** (*string*) – Custom note to set when cancelling. Defaults to information about who canceled the item.

Returns bool

get_account_billing_items (*mask=None*)
Gets all the topLevelBillingItems currently active on the account

Parameters **mask** (*string*) – Object Mask

Returns Billing_Item

get_billing_item (*identifier, mask=None*)
Gets details about a billing item

:param int identifier Billing_Item id :param string mask: Object mask to use. :return: Billing_Item

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 (*event_type*)
Retrieves a list of Notification_Occurrence_Events that have not ended yet

Param String event_type: notification event type.

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>

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, subnet_ids=None*)
 Authorizes hosts to 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. Only use with File volumes.

Returns Returns an array of SoftLayer_Network_Storage_Allowed_Host objects which now have access to the given 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

cancel_volume (*volume_id*, *reason='No longer needed'*, *immediate=False*)

Cancel the given storage volume.

Parameters

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

convert_dep_dupe (*volume_id*)

Convert a dependent duplicate volume to an independent volume.

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

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, *subnet_ids=None*)

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. Only use with File volumes

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 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

get_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_volume_count_limits ()

Returns a list of block volume count limit.

Returns Returns a list of block volume count limit.

get_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_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.

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, dependent_duplicate=False*)

Places an order for a duplicate 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 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

refresh_dep_dupe (*volume_id, snapshot_id*)
“Refresh a dependent duplicate volume with a snapshot from its parent.

Parameters

- **volume_id** (*integer*) – The id of the volume
- **snapshot_id** (*integer*) – The id of the snapshot

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 successfully 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 `VSManager.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
- **ttl** (*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
- **ttl** (*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
- **ttl** (*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: `strftime(‘%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.

- **t1** (*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'*, *t1=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*, *t1=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

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*)
Authorizes hosts to 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. Only use with File volumes.

Returns Returns an array of SoftLayer_Network_Storage_Allowed_Host objects which now have access to the given 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_flag** (*boolean*) – Cancel immediately or on anniversary date

cancel_volume (*volume_id*, *reason='No longer needed'*, *immediate=False*)
Cancels the given storage volume.

Parameters

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

convert_dep_dupe (*volume_id*)
Convert a dependent duplicate volume to an independent volume.

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

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, *subnet_ids=None*)
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. Only use with File volumes

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 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_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

get_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_volume_count_limits ()

Returns a list of block volume count limit.

Returns Returns a list of block volume count limit.

get_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_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.

list_file_volume_limit ()

Returns a list of block volume count limit.

Returns Returns a list of block 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, *dependent_duplicate=False*)

Places an order for a duplicate 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 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 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

refresh_dep_dupe (*volume_id, snapshot_id*)

“Refresh a dependent duplicate volume with a snapshot from its parent.

Parameters

- **volume_id** (*integer*) – The id of the volume
- **snapshot_id** (*integer*) – The id of the snapshot

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, redundant=None*)

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_hard_drives (*instance_id*)

Returns the hardware server hard drives.

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

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_storage_credentials (*instance_id*)

Returns the hardware server storage credentials.

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

get_storage_details (*instance_id*, *nas_type*)

Returns the hardware server attached network storage.

Parameters

- **instance_id** (*int*) – Id of the hardware server
- **nas_type** – storage type.

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

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

add_locations (*image_id*, *location_names*)

Add available locations to an archive image template.

Parameters

- **image_id** (*int*) – The ID of the image
- **location_names** – Locations for the Image.

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.)

get_locations_list (*image_id*, *location_names*)

Converts a list of location names to a list of locations.

Parameters

- **image_id** (*int*) – The ID of the image.
- **location_names** (*list*) – A list of location names strings.

Returns A list of locations associated with the given location names in the image.

get_storage_locations (*image_id*)

Get available locations for public image storage.

Parameters **image_id** (*int*) – The ID of the image

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.)

remove_locations (*image_id, location_names*)

Remove available locations from an archive image template.

Parameters

- **image_id** (*int*) – The ID of the image
- **location_names** – Locations for the Image.

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_secrecy** (*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_l7_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://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/

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/ :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/ :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/ :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_by_name (*name, mask=None*)

Gets a LBaaS by name.

Parameters

- **name** – Name of the LBaaS instance
- **mask** –

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, UUID or Name, this function will return UUI and LB Id.

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/
- **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 Listener 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

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')
'dal05'
>>> meta.get('fqdn')
'test.example.com'
```

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

get (*name*, *param=None*)
Retreive 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
- **vlan** (*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 (`instance_id, name=None, note=None, tags=None`)

Edit a vlan.

Parameters

- **instance_id** (`integer`) – the instance ID to edit.
- **name** (`string`) – valid name.
- **note** (`string`) – note about this particular vlan.
- **tags** (`string`) – tags to set on the vlan as a comma separated list. Use the empty string to remove all tags.

Returns `bool` – True or an Exception

edit_note_subnet (`subnet_id, note`)

Edit the note for this subnet.

Parameters

- **subnet_id** (`int`) – The ID of the subnet.
- **note** (`string`) – The note.

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_ip_by_address (*ip_address*)

get the ip address object

Parameters **ip_address** (*string*) – the ip address to edit.

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 diction 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.

set_subnet_ipaddress_note (*identifier, note*)

Set the ip address note of the subnet

Parameters **identifier** (*integer*) – the ip address ID to edit. **note** (*string*) – the note to edit.

set_tags (*tags, vlan_id*)

Sets tags on a vlan with a retry decorator

Just calls `vlan.setTags`, but if it fails from an `APIError` will retry.

set_tags_subnet (*subnet_id, tags*)

Tag a subnet by passing in one or more tags separated by a comma.

Parameters

- **subnet_id** (*int*) – The ID of the subnet.
- **tags** (*string*) – Comma separated list of tags.

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 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_local_disks (*instance_id*)

Returns the virtual server local disks.

Parameters `instance_id (int)` – Id of the virtual server

`get_portable_storage (instance_id)`

Returns the virtual server portable storage.

Parameters `instance_id (int)` – Id of the virtual server

`get_storage_credentials (instance_id)`

Returns the virtual server storage credentials.

Parameters `instance_id (int)` – Id of the virtual server

`get_storage_details (instance_id, nas_type)`

Returns the virtual server attached network storage.

Parameters

- `instance_id (int)` – Id of the virtual server
- `nas_type` – storage type.

`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 Soft-Layer.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,
    '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://softlayer.github.io/reference/services/SoftLayer_Virtual_ReservedCapacityGroup/
- 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://softlayer.github.io/reference/services/SoftLayer_Account/getPlacementGroups/
- 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/

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

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

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',
```

(continues on next page)

(continued from previous page)

```

'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

6.1.6 account billing-items

Lists billing items with some other useful information.

Similar to <https://cloud.ibm.com/billing/billing-items>


```
account billing-items [OPTIONS]
```

6.1.7 account item-detail

Gets detailed information about a billing item.

```
account item-detail [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.1.8 account cancel-item

Cancels a billing item.

```
account cancel-item [OPTIONS] IDENTIFIER
```

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|WINDOWS_SIXEN

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
- dependent-duplicate** <dependent_duplicate>
Whether or not this duplicate will be a dependent duplicate of the origin volume. [default: False]

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 performance|endurance

--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_SIXEN

--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

6.3.31 block volume-refresh

“Refresh a dependent duplicate volume with a snapshot from its parent.

```
block volume-refresh [OPTIONS] VOLUME_ID SNAPSHOT_ID
```


Arguments

VOLUME_ID

Required argument

SNAPSHOT_ID

Required argument

6.3.32 block volume-convert

Convert a dependent duplicate volume to an independent volume.

```
block volume-convert [OPTIONS] VOLUME_ID
```

Arguments

VOLUME_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|video|file

-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

This function allows you to easily call any API. The format is

```
slcli call-api SoftLayer_Service method param1 param2 -id=1234 -mask="mask[id,name]"
```

Parameters should be in the order they are presented on `sldn.softlayer.com`. Any complex parameters (those that link to other datatypes) should be presented as JSON strings. They need to be enclosed in single quotes ('), and variables and strings enclosed in double quotes (").

For example: `{"hostname": "test", "ssh_keys": [{"id": 1234}]}`

6.5.1 call-api

Call arbitrary API endpoints with the given SERVICE and METHOD.

For parameters that require a datatype, use a JSON string for that parameter. 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'
slcli call-api Account getVirtualGuests \
    --json-filter '{"virtualGuests":{"hostname":{"operation":"^= test"}}}' --limit=10
slcli -v call-api SoftLayer_User_Customer addBulkPortalPermission --id=1234567 \
    '{"keyName": "NETWORK_MESSAGE_DELIVERY_MANAGE"}'
```

```
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 should use `-json-filter`. (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

--json-filter <json_filter>

A JSON string to be passed in as the object filter to the API call. Remember to use double quotes (") for variable names. Can NOT be used with `-filter`. Dont use whitespace outside of strings, or the slcli might have trouble parsing it.

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 its 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
- dependent-duplicate** <dependent_duplicate>
Whether or not this duplicate will be a dependent duplicate of the origin volume. [default: False]

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 performancelendurance
- 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 deprecated

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

6.12.26 file volume-refresh

“Refresh a dependent duplicate volume with a snapshot from its parent.

```
file volume-refresh [OPTIONS] VOLUME_ID SNAPSHOT_ID
```

Arguments

VOLUME_ID

Required argument

SNAPSHOT_ID

Required argument

6.12.27 file volume-convert

Convert a dependent duplicate volume to an independent volume.

```
file volume-convert [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 hardware 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
```

```
hardware 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 hardware cancel-reasons

Display a list of cancellation reasons.

```
hardware cancel-reasons [OPTIONS]
```

6.15.3 hardware cancel

Cancel a dedicated server.

```
hardware 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 hardware create-options

Server order options for a given chassis.

```
hardware create-options [OPTIONS]
```

6.15.5 hardware create

Order/create a dedicated server.

```
hardware 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 hardware credentials

List server credentials.

```
hardware credentials [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.15.7 hardware detail

Get details for a hardware device.

```
hardware detail [OPTIONS] IDENTIFIER
```

Options

--passwords

Show passwords (check over your shoulder!)

--price

Show associated prices

Arguments

IDENTIFIER

Required argument

6.15.8 hardware billing

Get billing for a hardware device.

```
hardware billing [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.15.9 hardware edit

Edit hardware details.

```
hardware 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
- redundant**
The desired state of redundancy for the interface(s)
- degraded**
The desired state of degraded for the interface(s)

Arguments

IDENTIFIER

Required argument

Note : Using multiple ‘ : ‘ can cause an error.

```
$ slcli hw edit 123456 --tag "cloud:service:db2whoc, cloud:svcplan:flex, cloud:svcenv:prod, cloud:bmixenv:fra"
```

```
TransportError(0): ('Connection aborted.',
```

```
RemoteDisconnected('Remote end closed connection without response',))
```

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.

6.15.10 hardware list

List hardware servers.

```
hardware 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]

6.15.11 hardware power-cycle

Power cycle a server.

```
hardware power-cycle [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER
Required argument

6.15.12 hardware power-off

Power off an active server.

```
hardware power-off [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER
Required argument

6.15.13 hardware power-on

Power on a server.

```
hardware power-on [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.15.14 hardware reboot

Reboot an active server.

```
hardware reboot [OPTIONS] IDENTIFIER
```

Options

--hard, --soft

Perform a hard or soft reboot

Arguments

IDENTIFIER

Required argument

6.15.15 hardware reload

Reload operating system on a server.

```
hardware 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

6.15.16 hardware rescue

Reboot server into a rescue image.

```
hardware rescue [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.15.17 hardware reflash-firmware

Reflash server firmware.

```
hardware 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.

6.15.18 hardware update-firmware

Update server firmware.

```
hardware 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.

6.15.19 hardware toggle-ipmi

Toggle the IPMI interface on and off

```
hardware toggle-ipmi [OPTIONS] IDENTIFIER
```

Options

--enable, --disable

Whether enable (DEFAULT) or disable the interface.

Arguments

IDENTIFIER

Required argument

6.15.20 hardware ready

Check if a server is ready.

```
hardware ready [OPTIONS] IDENTIFIER
```

Options

--wait <wait>
Seconds to wait [default: 0]

Arguments

IDENTIFIER
Required argument

6.15.21 hardware dns-sync

Sync DNS records.

```
hardware 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

6.15.22 hardware storage

Get storage details for a hardware server.

```
hardware storage [OPTIONS] IDENTIFIER
```

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.bluemix.net/docs/services/cloud-object-storage/iam/users-serviceids.html#serviceidapikeys>

--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.bluemix.net/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

6.16.7 image datacenter

Add/Remove datacenter of an image.

```
image datacenter [OPTIONS] IDENTIFIER LOCATIONS...
```

Options

--add, --remove
To add or remove Datacenter

Arguments

IDENTIFIER

Required argument

LOCATIONS

Required argument(s)

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

ipsec list

List IPsec VPN tunnel contexts

```
ipsec list [OPTIONS]
```

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

ipsec detail

List IPSEC VPN tunnel context details.

Additional resources can be joined using multiple instances of the include option, for which the following choices are available.

at: address translations
 is: internal subnets
 rs: remote subnets
 sr: statically routed subnets
 ss: service subnets

```
ipsec detail [OPTIONS] CONTEXT_ID
```

Options

-i, --include <include>
 Include additional resources

Options at|is|rs|sr|ss

Arguments

CONTEXT_ID

Required argument

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:
```

(continues on next page)

(continued from previous page)

```
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:
↪.....:.....:
:  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:
:.....:.....:.....:.....:.....:.....:.....:.....:.....:.....:
:  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

ipsec update

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.

```
ipsec update [OPTIONS] CONTEXT_ID
```

Options

- friendly-name** <friendly_name>
Friendly name value
- remote-peer** <remote_peer>
Remote peer IP address value
- preshared-key** <preshared_key>
Preshared key value
- phase1-auth, --p1-auth** <phase1_auth>
Phase 1 authentication value
Options MD5|SHA1|SHA256
- phase1-crypto, --p1-crypto** <phase1_crypto>
Phase 1 encryption value
Options DES|3DES|AES128|AES192|AES256
- phase1-dh, --p1-dh** <phase1_dh>
Phase 1 diffie hellman group value
Options 0|1|2|5
- phase1-key-ttl, --p1-key-ttl** <phase1_key_ttl>
Phase 1 key life value
- phase2-auth, --p2-auth** <phase2_auth>
Phase 2 authentication value
Options MD5|SHA1|SHA256
- phase2-crypto, --p2-crypto** <phase2_crypto>
Phase 2 encryption value
Options DES|3DES|AES128|AES192|AES256
- phase2-dh, --p2-dh** <phase2_dh>
Phase 2 diffie hellman group value
Options 0|1|2|5
- phase2-forward-secret, --p2-forward-secret** <phase2_forward_secret>
Phase 2 perfect forward secrecy value
- phase2-key-ttl, --p2-key-ttl** <phase2_key_ttl>
Phase 2 key life value

Arguments

CONTEXT_ID

Required argument

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.
```

(continues on next page)

(continued from previous page)

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]
                                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

ipsec configure

Request configuration of a tunnel context.

This action will update the advancedConfigurationFlag on the context instance and further modifications against the context will be prevented until all changes can be propagated to network devices.

```
ipsec configure [OPTIONS] CONTEXT_ID
```

Arguments

CONTEXT_ID

Required argument

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

ipsec subnet-add

Add a subnet to an IPSEC tunnel context.

A subnet id may be specified to link to the existing tunnel context.

Otherwise, a network identifier in CIDR notation should be specified, indicating that a subnet resource should first be created before associating it with the tunnel context. Note that this is only supported for remote subnets, which are also deleted upon failure to attach to a context.

A separate configuration request should be made to realize changes on network devices.

```
ipsec subnet-add [OPTIONS] CONTEXT_ID
```

Options

-s, --subnet-id <subnet_id>
Subnet identifier to add

-t, --subnet-type, --type <subnet_type>
Subnet type to add [required]

Options internalremotelservice

-n, --network-identifier, --network <network_identifier>
Subnet network identifier to create

Arguments

CONTEXT_ID

Required argument

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

ipsec subnet-remove

Remove a subnet from an IPSEC tunnel context.

The subnet id to remove must be specified.

Remote subnets are deleted upon removal from a tunnel context.

A separate configuration request should be made to realize changes on network devices.

```
ipsec subnet-remove [OPTIONS] CONTEXT_ID
```

Options

-s, --subnet-id <subnet_id>
Subnet identifier to remove [required]

-t, --subnet-type, --type <subnet_type>
Subnet type to add [required]

Options internalremotelservice

Arguments

CONTEXT_ID
Required argument

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

ipsec translation-add

Add an address translation to an IPSEC tunnel context.

A separate configuration request should be made to realize changes on network devices.

```
ipsec translation-add [OPTIONS] CONTEXT_ID
```

Options

- s, --static-ip** <static_ip>
Static IP address value [required]
- r, --remote-ip** <remote_ip>
Remote IP address value [required]
- n, --note** <note>
Note value

Arguments

CONTEXT_ID
Required argument

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

ipsec translation-remove

Remove a translation entry from an IPSEC tunnel context.

A separate configuration request should be made to realize changes on network devices.

```
ipsec translation-remove [OPTIONS] CONTEXT_ID
```

Options

- t, --translation-id** <translation_id>
Translation identifier to remove [required]

Arguments

CONTEXT_ID
Required argument

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

ipsec translation-update

Update an address translation for an IPSEC tunnel context.

A separate configuration request should be made to realize changes on network devices.

```
ipsec translation-update [OPTIONS] CONTEXT_ID
```

Options

```
-t, --translation-id <translation_id>
    Translation identifier to update [required]
-s, --static-ip <static_ip>
    Static IP address value
-r, --remote-ip <remote_ip>
    Remote IP address value
-n, --note <note>
    Note value
```

Arguments

```
CONTEXT_ID
    Required argument
```

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-del

Remove a LBaaS member.

Member UUID can be found from *slcli lb detail*.

```
loadbal member-del [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-del

Removes the listener from identified LBaaS instance

LISTENER should be a UUID, and can be found from *slcli lb detail <IDENTIFIER>*

```
loadbal pool-del [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 NAS Commands

6.19.1 nas list

List NAS accounts.

```
nas list [OPTIONS]
```

6.19.2 nas credentials

List NAS account credentials.

```
nas credentials [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.20 Object Storage Commands

6.20.1 object-storage accounts

List object storage accounts.

```
object-storage accounts [OPTIONS]
```

6.20.2 object-storage endpoints

List object storage endpoints.

```
object-storage endpoints [OPTIONS]
```

6.20.3 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

6.20.4 object-storage credential limit

Credential limits for this IBM Cloud Object Storage account.

```
object-storage credential limit [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.20.5 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

6.20.6 object-storage credential create

Create credentials for an IBM Cloud Object Storage Account

```
object-storage credential create [OPTIONS] IDENTIFIER
```


Arguments

IDENTIFIER

Required argument

6.21 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>

6.21.1 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

6.21.2 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

6.21.3 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      :
```

(continues on next page)

(continued from previous page)

:	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	:
:	:	:	:

6.21.4 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.

6.21.5 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

6.21.6 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.

6.21.7 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
```

6.21.8 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
```

6.22 Quotes

6.22.1 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

6.22.2 order quote-list

List all active quotes on an account

```
order quote-list [OPTIONS]
```

6.22.3 order quote-detail

View a quote

```
order quote-detail [OPTIONS] QUOTE
```

Arguments

QUOTE

Required argument

6.22.4 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 \
```

(continues on next page)

(continued from previous page)

```
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)

6.23 Reports

There are a few report type commands in the SLCLI.

6.23.1 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.

6.23.2 report bandwidth

Bandwidth report for every pool/server.

This reports on the total data transferred 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]

6.24 Reverse Whois Commands

6.24.1 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

6.24.2 rwhois show

Display the RWhois information for your account.

```
rwhois show [OPTIONS]
```

6.25 Security Groups

6.25.1 securitygroup list

List security groups.

```
securitygroup list [OPTIONS]
```

Options

--sortby <sortby>

Column to sort by

Options idnamedescription

-1, --limit <limit>

How many results to get in one api call, default is 100 [default: 100]

6.25.2 securitygroup detail

Get details about a security group.

```
securitygroup detail [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.25.3 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

6.25.4 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

6.25.5 securitygroup delete

Deletes the given security group

```
securitygroup delete [OPTIONS] SECURITYGROUP_ID
```

Arguments

SECURITYGROUP_ID
Required argument

6.25.6 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

6.25.7 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

6.25.8 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

6.25.9 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

6.25.10 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

6.25.11 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

6.25.12 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

6.25.13 securitygroup event-log

Search for event logs by request id

```
securitygroup event-log [OPTIONS] REQUEST_ID
```

Arguments

REQUEST_ID
Required argument

6.26 SSH Keys

6.26.1 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

6.26.2 sshkey remove

Permanently removes an SSH key.

```
sshkey remove [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.26.3 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

6.26.4 sshkey list

List SSH keys.

```
sshkey list [OPTIONS]
```

Options

--sortby <sortby>
Column to sort by

Options idlabelfingerprinlnotes

6.26.5 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

6.27 SSL Certificates

6.27.1 ssl add

Add and upload SSL certificate details.

```
ssl add [OPTIONS]
```

Options

--cert <cert>
Certificate file

--csr <csr>
Certificate Signing Request file

--icc <icc>
Intermediate Certificate file

--key <key>
Private Key file

--notes <notes>
Additional notes

6.27.2 ssl download

Download SSL certificate and key file.

```
ssl download [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER
Required argument

6.27.3 ssl edit

Edit SSL certificate.

```
ssl edit [OPTIONS] IDENTIFIER
```

Options

--cert <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

6.27.4 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 idcommon_name|days_until_expire|notes

6.27.5 ssl remove

Remove SSL certificate.

```
ssl remove [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.28 Subnets

6.28.1 subnet cancel

Cancel a subnet.

```
subnet cancel [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.28.2 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

6.28.3 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

6.28.4 subnet edit

Edit note and tags of a subnet.

```
subnet edit [OPTIONS] IDENTIFIER
```

Options

-t, --tags <tags>

Comma separated list of tags, enclosed in quotes. "tag1, tag2"

-n, --note <note>

The note

Arguments

IDENTIFIER

Required argument

6.28.5 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

6.28.6 subnet lookup

Find an IP address and display its subnet and device info.

```
subnet lookup [OPTIONS] IP_ADDRESS
```

Arguments

IP_ADDRESS
Required argument

6.28.7 subnet edit-ip

Set the note of the ipAddress

```
subnet edit-ip [OPTIONS] IDENTIFIER
```

Options

--note <note>
set ip address note of subnet

Arguments

IDENTIFIER

Required argument

6.29 Tag Commands

These commands will allow you to interact with the **IMS** provier tagging service. The **IBM Global Search and Tagging API** can be used to interact with both the **GHOST** provider and **IMS** provider. The **GHOST** provider will handle tags for things outside of the Classic Infrastructure (aka SoftLayer) space.

6.29.1 tags list

List Tags.

```
tags list [OPTIONS]
```

Options

-d, --detail

Show information about the resources using this tag.

6.29.2 tags set

Set Tags.

```
tags set [OPTIONS]
```

Options

-t, --tags <tags>

Comma seperated list of tags, enclosed in quotes. "tag1, tag2" [required]

-k, --key-name <key_name>

Key name of a tag type e.g. GUEST, HARDWARE. See slcli tags taggable output. [required]

-r, --resource-id <resource_id>

ID of the object being tagged [required]

6.29.3 tags details

Get details for a Tag. Identifier can be either a name or tag-id

```
tags details [OPTIONS] IDENTIFIER
```

Options

--name

Assume identifier is a tag name. Useful if your tag name is a number.

Arguments

IDENTIFIER

Required argument

6.29.4 tags delete

Delete a Tag. Tag names that contain spaces need to be encased in quotes

```
tags delete [OPTIONS] IDENTIFIER
```

Options

--name

Assume identifier is a tag name. Useful if your tag name is a number.

Arguments

IDENTIFIER

Required argument

6.29.5 tags taggable

List everything that could be tagged.

```
tags taggable [OPTIONS]
```

6.29.6 tags cleanup

Removes all empty tags.

```
tags cleanup [OPTIONS]
```

Options

-d, --dry-run

Don't delete, just show what will be deleted.

6.30 Support Tickets

6.30.1 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

6.30.2 ticket detail

Get details for a ticket.

```
ticket detail [OPTIONS] IDENTIFIER
```

Options

--count <count>

Number of updates [default: 10]

Arguments

IDENTIFIER

Required argument

6.30.3 ticket list

List tickets.


```
ticket list [OPTIONS]
```

Options

--open, --closed

Display only open or closed tickets

6.30.4 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

6.30.5 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

6.30.6 ticket subjects

List Subject IDs for ticket creation.

```
ticket subjects [OPTIONS]
```

6.30.7 ticket summary

Summary info about tickets.

```
ticket summary [OPTIONS]
```

6.30.8 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

6.30.9 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

6.31 Users

Version 5.6.0 introduces the ability to interact with user accounts from the cli.

6.31.1 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]

6.31.2 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

6.31.3 user permissions

User Permissions.

```
user permissions [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.31.4 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

6.31.5 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/ [required]

Arguments

USER

Required argument

6.31.6 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/

-a, --api-key

Create an API key for this user.

Arguments

USERNAME

Required argument

6.31.7 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

6.31.8 user vpn-manual

Enable or disable user vpn subnets manual config

```
user vpn-manual [OPTIONS] USER
```

Options

--enable, --disable

Enable or disable vpn subnets manual config.

Arguments

USER

Required argument

6.31.9 user vpn-subnet

Add or remove subnets access for a user.

```
user vpn-subnet [OPTIONS] USER SUBNET...
```

Options

--add, --remove

Add or remove access to subnets.

Arguments

USER

Required argument

SUBNET

Required argument(s)

6.32 VLANs

6.32.1 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

6.32.2 vlan edit

Edit a vlan's details.

```
vlan edit [OPTIONS] IDENTIFIER
```

Options

- n, --name** <name>
The optional name for this VLAN
- e, --note** <note>
The note for this vlan.
- g, --tags** <tags>
Tags to set e.g. "tag1,tag2", or empty string to remove all

Arguments

- IDENTIFIER**
Required argument

6.32.3 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]

6.33 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                       ↵
↪          :
```

(continues on next page)

(continued from previous page)

```

:          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
↳          :
:          : 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
↳          :

```

(continues on next page)


```
$ 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 :
:      status : 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 :
:.....:.....:
```

6.33.1 virtual 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
```

```
virtual 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.

6.33.2 virtual cancel

Cancel virtual servers.

```
virtual cancel [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.33.3 virtual capture

Capture one or all disks from a virtual server to a SoftLayer image.

```
virtual 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

6.33.4 virtual create

Order/create virtual servers.

```
virtual 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

6.33.5 virtual create-options

Virtual server order options.

```
virtual create-options [OPTIONS]
```

6.33.6 virtual dns-sync

Sync DNS records.

```
virtual 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

6.33.7 virtual edit

Edit a virtual server's details.

```
virtual 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

6.33.8 virtual list

List virtual servers.

```
virtual 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]

6.33.9 virtual pause

Pauses an active virtual server.

```
virtual pause [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.33.10 virtual power-on

Power on a virtual server.

```
virtual power-on [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.33.11 virtual power-off

Power off an active virtual server.

```
virtual power-off [OPTIONS] IDENTIFIER
```

Options

--hard, --soft

Perform a hard shutdown

Arguments

IDENTIFIER

Required argument

6.33.12 virtual resume

Resumes a paused virtual server.

```
virtual resume [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.33.13 virtual rescue

Reboot into a rescue image.

```
virtual rescue [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.33.14 virtual reboot

Reboot an active virtual server.

```
virtual reboot [OPTIONS] IDENTIFIER
```

Options

--hard, --soft

Perform a hard or soft reboot

Arguments

IDENTIFIER

Required argument

6.33.15 virtual ready

Check if a virtual server is ready.

```
virtual ready [OPTIONS] IDENTIFIER
```

Options

--wait <wait>

Seconds to wait [default: 0]

Arguments

IDENTIFIER

Required argument

6.33.16 virtual upgrade

Upgrade a virtual server.

```
virtual 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

6.33.17 virtual usage

Usage information of a virtual server.

```
virtual 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

6.33.18 virtual storage

Get storage details for a virtual server.

```
virtual storage [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.33.19 virtual billing

Get billing for a virtual device.

```
virtual billing [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER

Required argument

6.33.20 virtual detail

Get details for a virtual server.

```
virtual detail [OPTIONS] IDENTIFIER
```

Options

--passwords

Show passwords (check over your shoulder!)

--price

Show associated prices

Arguments

IDENTIFIER

Required argument

6.33.21 virtual reload

Reload operating system on a virtual server.

```
virtual reload [OPTIONS] IDENTIFIER
```

Options

- i, --postinstall** <postinstall>
Post-install script to download
- image** <image>
Image ID. The default is to use the current operating system. See: ‘slcli image list’ for reference
- k, --key** <key>
SSH keys to add to the root user (multiple occurrence permitted)

Arguments

IDENTIFIER
Required argument

6.33.22 virtual credentials

List virtual server credentials.

```
virtual credentials [OPTIONS] IDENTIFIER
```

Arguments

IDENTIFIER
Required argument

6.33.23 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 : ↪  
↪   Created          :          :                   :          :          ↪  
:.....:.....:.....:.....:.....:.....:.....  
↪.....:.....:.....:.....:.....:.....:.....  
: 1234 :  test-capacity   : ####------ : B1.1x2 (1 Year Term) : bcr02a.dal13 : ↪  
↪2018-09-24T16:33:09-06:00 :          :                   :                   :          ↪  
:.....:.....:.....:.....:.....:.....:.....  
↪.....:.....:.....:.....:.....:.....:.....
```

6.33.24 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
:.....:
↪.....:
:
: Placement Groups
↪
:
:.....:
↪.....:
: Id : Name : Backend Router : Rule : Guests : Created
↪
:.....:
↪.....:
: 31741 : fotest : bcr01a.tor01 : SPREAD : 1 : 2018-11-22T14:36:10-
↪06:00 :
: 64535 : testGroup : bcr01a.mex01 : SPREAD : 3 : 2019-01-17T14:36:42-
↪06:00 :
:.....:
↪.....:
```

vs placementgroup detail

This command will provide some detailed information about a specific placement group

```
$ slcli vs placementgroup detail testGroup
:.....:
: Id : Name : Backend Router : Rule : Created :
:.....:
: 64535 : testGroup : bcr01a.mex01 : SPREAD : 2019-01-17T14:36:42-06:00 :
:.....:
:.....:
↪.....:
```

(continues on next page)

(continued from previous page)

Id	FQDN	Primary IP	Backend IP	CPU	Memory	
↳ Provisioned	Transaction					
.....
↳					
: 69134895	: testGroup62.test.com	: 169.57.70.166	: 10.131.11.32	: 1	: 1024	: ↳
↳ 2019-01-17T17:44:50-06:00	: -	:				
: 69134901	: testGroup72.test.com	: 169.57.70.184	: 10.131.11.59	: 1	: 1024	: ↳
↳ 2019-01-17T17:44:53-06:00	: -	:				
: 69134887	: testGroup52.test.com	: 169.57.70.187	: 10.131.11.25	: 1	: 1024	: ↳
↳ 2019-01-17T17:44:43-06:00	: -	:				
.....
↳					

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, 75
SoftLayer.managers.account, 8
SoftLayer.managers.autoscale, 9
SoftLayer.managers.block, 11
SoftLayer.managers.cdn, 17
SoftLayer.managers.dedicated_host, 19
SoftLayer.managers.dns, 22
SoftLayer.managers.event_log, 24
SoftLayer.managers.file, 25
SoftLayer.managers.firewall, 31
SoftLayer.managers.hardware, 33
SoftLayer.managers.image, 38
SoftLayer.managers.ipsec, 40
SoftLayer.managers.load_balancer, 43
SoftLayer.managers.metadata, 46
SoftLayer.managers.network, 47
SoftLayer.managers.ordering, 52
SoftLayer.managers.sshkey, 58
SoftLayer.managers.ssl, 59
SoftLayer.managers.ticket, 60
SoftLayer.managers.vs, 62
SoftLayer.managers.vs_capacity, 71
SoftLayer.managers.vs_placement, 72

Symbols

- aaaa-record
 - hardware-dns-sync command line option, 138
 - virtual-dns-sync command line option, 197
- abuse <abuse>
 - rwhois-edit command line option, 167
- ack
 - account-event-detail command line option, 86
- ack-all
 - account-events command line option, 85
- add, -remove
 - image-datacenter command line option, 141
 - user-vpn-subnet command line option, 188
- address1 <address1>
 - rwhois-edit command line option, 167
- address2 <address2>
 - rwhois-edit command line option, 167
- all
 - account-invoices command line option, 86
- all <all>
 - virtual-capture command line option, 194
- amount <amount>
 - autoscale-scale command line option, 88
- billing <billing>
 - block-volume-duplicate command line option, 98
 - block-volume-order command line option, 100
 - dedicatedhost-create command line option, 109
 - file-volume-duplicate command line option, 124
 - file-volume-order command line option, 126
 - hardware-create command line option, 132
 - order-place command line option, 163
 - virtual-create command line option, 195
- body <body>
 - ticket-create command line option, 182
 - ticket-update command line option, 183
- boot-mode <boot_mode>
 - virtual-create command line option, 195
- by, -to
 - autoscale-scale command line option, 88
- by-id <by_id>
 - dns-record-edit command line option, 113
- by-record <by_record>
 - dns-record-edit command line option, 113
- byol
 - image-import command line option, 140
- capacity <capacity>
 - block-snapshot-order command line option, 96
 - file-snapshot-order command line option, 122
- category <category>
 - order-item-list command line option, 161
- city <city>
 - rwhois-edit command line option, 167
- closed

- account-invoices command line option, 86
- cloud-init
 - image-import command line option, 140
- columns <columns>
 - block-access-list command line option, 90
 - block-replica-locations command line option, 93
 - block-replica-partners command line option, 93
 - block-snapshot-list command line option, 96
 - block-volume-list command line option, 99
 - dedicatedhost-list command line option, 108
 - dedicatedhost-list-guests command line option, 111
 - file-access-list command line option, 117
 - file-replica-locations command line option, 119
 - file-replica-partners command line option, 119
 - file-snapshot-list command line option, 121
 - file-volume-list command line option, 125
 - hardware-list command line option, 135
 - user-list command line option, 185
 - virtual-list command line option, 198
- comment <comment>
 - hardware-cancel command line option, 131
- company <company>
 - rwhois-edit command line option, 167
- complex-type <complex_type>
 - order-place command line option, 163
 - order-place-quote command line option, 166
 - order-quote command line option, 164
- count <count>
 - ticket-detail command line option, 182
- country <country>
 - rwhois-edit command line option, 167
- cpu <cpu>
 - autoscale-edit command line option, 89
 - virtual-upgrade command line option, 201
- crt <crt>
 - ssl-add command line option, 175
 - ssl-edit command line option, 176
- csr <csr>
 - ssl-add command line option, 175
 - ssl-edit command line option, 176
- data <data>
 - dns-record-edit command line option, 113
 - dns-record-list command line option, 113
- day-of-week <day_of_week>
 - block-snapshot-enable command line option, 95
 - file-snapshot-enable command line option, 121
- dedicated, -public
 - virtual-create command line option, 195
- degraded
 - hardware-edit command line option, 134
- dependent-duplicate <dependent_duplicate>
 - block-volume-duplicate command line option, 98
 - file-volume-duplicate command line option, 124
- details
 - account-invoice-detail command line option, 86
- disk <disk>
 - virtual-create command line option, 196
- dry-run
 - dns-import command line option, 111
- enable, -disable
 - hardware-toggle-ipmi command line option, 137
 - user-edit-permissions command line option, 186
 - user-vpn-manual command line option, 188
- end <end>
 - report-bandwidth command line option, 167
- export <export>
 - dedicatedhost-create command line option, 109
 - hardware-create command line option, 132
 - virtual-create command line option, 195

-extras <extras>
 order-place command line option, 163
 order-place-quote command line option, 166
 -firewall-type <firewall_type>
 firewall-add command line option, 127
 -firstname <firstname>
 rwhois-edit command line option, 167
 -flavor <flavor>
 virtual-upgrade command line option, 201
 -fqdn <fqdn>
 order-quote command line option, 164
 -friendly-name <friendly_name>
 ipsec-update command line option, 145
 -guests
 dedicatedhost-detail command line option, 110
 -ha, -high-availability
 firewall-add command line option, 127
 -hard, -soft
 hardware-reboot command line option, 136
 virtual-power-off command line option, 199
 virtual-reboot command line option, 200
 -hardware <hardware_identifier>
 ticket-attach command line option, 184
 ticket-create command line option, 182
 ticket-detach command line option, 184
 -healthInterval <healthinterval>
 loadbal-l7pool-add command line option, 155
 -healthPath <healthpath>
 loadbal-l7pool-add command line option, 155
 -healthRetry <healthretry>
 loadbal-l7pool-add command line option, 155
 -healthTimeout <healthtimeout>
 loadbal-l7pool-add command line option, 155
 -history <history>
 cdn-detail command line option, 103
 -host-id <host_id>
 virtual-create command line option, 195
 -hour <hour>
 block-snapshot-enable command line option, 95
 file-snapshot-enable command line option, 121
 -hourly
 virtual-list command line option, 198
 -ibm-api-key <ibm_api_key>
 image-export command line option, 141
 image-import command line option, 140
 -icc <icc>
 ssl-add command line option, 175
 ssl-edit command line option, 176
 -id <_id>
 call-api command line option, 106
 -identifier <identifier>
 subnet-list command line option, 179
 -image <image>
 order-quote command line option, 164
 virtual-create command line option, 195
 virtual-reload command line option, 203
 -immediate
 block-snapshot-cancel command line option, 94
 block-volume-cancel command line option, 97
 file-snapshot-cancel command line option, 120
 file-volume-cancel command line option, 123
 hardware-cancel command line option, 131
 -iops <iops>
 block-volume-order command line option, 100
 file-volume-order command line option, 126
 -ip-address <ip_address>
 block-access-authorize command line option, 90
 block-access-revoke command line option, 91
 file-access-authorize command line option, 116
 file-access-revoke command line option, 117
 -ip-version <ip_version>
 globalip-list command line option, 130

-ipv4, -v4
 subnet-list command line option, 179

-ipv6
 virtual-create command line option, 196

-ipv6, -v6
 subnet-create command line option, 177
 subnet-list command line option, 179

-is-encrypted
 image-import command line option, 140

-json-filter <json_filter>
 call-api command line option, 106

-key <key>
 ssl-add command line option, 175
 ssl-edit command line option, 176

-keyword <keyword>
 order-item-list command line option, 161
 order-package-list command line option, 159
 order-preset-list command line option, 162

-lastname <lastname>
 rwhois-edit command line option, 167

-like <like>
 virtual-create command line option, 196

-limit <limit>
 account-invoices command line option, 86
 call-api command line option, 106

-location <location>
 block-volume-order command line option, 100
 file-volume-order command line option, 126

-mask <mask>
 call-api command line option, 106

-max <maximum>
 autoscale-edit command line option, 89

-memory <memory>
 autoscale-edit command line option, 89
 virtual-upgrade command line option, 201

-metadata, -no-metadata
 event-log-get command line option, 115

-min <minimum>
 autoscale-edit command line option, 89

-minute <minute>
 block-snapshot-enable command line option, 95
 file-snapshot-enable command line option, 121

-monthly
 virtual-list command line option, 198

-name
 tags-delete command line option, 181
 tags-details command line option, 181

-name <name>
 autoscale-edit command line option, 89
 image-edit command line option, 139
 image-list command line option, 140
 order-place-quote command line option, 166
 ticket-upload command line option, 183
 vlan-list command line option, 190

-network <network>
 virtual-upgrade command line option, 201

-network-space <network_space>
 subnet-list command line option, 179

-no-hardware
 subnet-detail command line option, 178
 vlan-detail command line option, 189

-no-public
 hardware-create command line option, 132

-no-vs
 subnet-detail command line option, 178
 vlan-detail command line option, 189

-note <note>
 image-edit command line option, 139
 image-import command line option, 140
 sshkey-add command line option, 173
 sshkey-edit command line option, 174
 subnet-edit-ip command line option, 179
 virtual-capture command line option, 194

-notes <notes>
 ssl-add command line option, 175
 ssl-edit command line option, 176

-offset <offset>
 call-api command line option, 106

-open, -closed

- ticket-list command line option, 183
- os-code <os_code>
 - image-import command line option, 140
- os-type <os_type>
 - block-replica-order command line option, 92
 - block-volume-order command line option, 100
- output-python, -no-output-python call-api command line option, 106
- package_type <package_type>
 - order-package-list command line option, 159
- passwords
 - hardware-detail command line option, 133
 - virtual-detail command line option, 202
- path <path>
 - ticket-upload command line option, 183
- phase1-auth, -p1-auth <phase1_auth>
 - ipsec-update command line option, 145
- phase1-crypto, -p1-crypto <phase1_crypto>
 - ipsec-update command line option, 145
- phase1-dh, -p1-dh <phase1_dh>
 - ipsec-update command line option, 145
- phase1-key-ttl, -p1-key-ttl <phase1_key_ttl>
 - ipsec-update command line option, 145
- phase2-auth, -p2-auth <phase2_auth>
 - ipsec-update command line option, 145
- phase2-crypto, -p2-crypto <phase2_crypto>
 - ipsec-update command line option, 145
- phase2-dh, -p2-dh <phase2_dh>
 - ipsec-update command line option, 145
- phase2-forward-secrecy, -p2-forward-secrecy <phase2_forward_secrecy>
 - ipsec-update command line option, 145
- phase2-key-ttl, -p2-key-ttl <phase2_key_ttl>
 - ipsec-update command line option, 145
- placementgroup <placementgroup>
 - virtual-create command line option, 196
- port <port>
 - dns-record-add command line option, 112
- port-speed <port_speed>
 - hardware-create command line option, 132
- postal <postal>
 - rwhois-edit command line option, 167
- preset <preset>
 - order-place command line option, 163
 - order-place-quote command line option, 166
- preshared-key <preshared_key>
 - ipsec-update command line option, 145
- price
 - dedicatedhost-detail command line option, 110
 - hardware-detail command line option, 133
 - virtual-detail command line option, 202
- priority <priority>
 - dns-record-add command line option, 112
 - ticket-create command line option, 182
- private
 - virtual-create command line option, 196
 - virtual-upgrade command line option, 201
- private, -public
 - loadbal-member-add command line option, 152
- private-speed <private_speed>
 - hardware-edit command line option, 134
 - virtual-edit command line option, 198
- protocol <protocol>
 - dns-record-add command line option, 112
- ptr
 - hardware-dns-sync command line option, 138
 - virtual-dns-sync command line option, 197
- public
 - loadbal-order command line option,

156

- public, -private
 - image-list command line option, 140
 - rwhois-edit command line option, 167
- public-speed <public_speed>
 - hardware-edit command line option, 134
 - virtual-edit command line option, 197
- quantity <quantity>
 - order-place command line option, 163
 - order-quote command line option, 164
- reason <reason>
 - block-snapshot-cancel command line option, 94
 - block-volume-cancel command line option, 97
 - file-snapshot-cancel command line option, 120
 - file-volume-cancel command line option, 123
 - hardware-cancel command line option, 131
- record <record>
 - dns-record-list command line option, 113
- redundant
 - hardware-edit command line option, 134
- remote-peer <remote_peer>
 - ipsec-update command line option, 145
- replicant-id <replicant_id>
 - block-replica-failover command line option, 92
 - file-replica-failover command line option, 118
- required
 - order-category-list command line option, 160
- retention-count <retention_count>
 - block-snapshot-enable command line option, 95
 - file-snapshot-enable command line option, 121
- root-key-crn <root_key_crn>
 - image-import command line option, 140
- san
 - virtual-create command line option, 195
- schedule-type <schedule_type>
 - block-snapshot-disable command line option, 95
 - block-snapshot-enable command line option, 95
 - file-snapshot-disable command line option, 121
 - file-snapshot-enable command line option, 121
- send-email
 - order-place-quote command line option, 166
- service <service>
 - dns-record-add command line option, 112
- service-offering <service_offering>
 - block-volume-order command line option, 100
 - file-volume-order command line option, 126
- size <size>
 - block-volume-order command line option, 100
 - file-volume-order command line option, 126
- snapshot-size <snapshot_size>
 - block-volume-order command line option, 100
 - file-volume-order command line option, 126
- sortby <sortby>
 - block-access-list command line option, 90
 - block-replica-locations command line option, 93
 - block-replica-partners command line option, 93
 - block-snapshot-list command line option, 96
 - block-volume-count command line option, 97
 - block-volume-limits command line option, 101
 - block-volume-list command line option, 99
 - cdn-list command line option, 104
 - dedicatedhost-list command line option, 108
 - dedicatedhost-list-guests command line option, 111
 - file-access-list command line option, 117
 - file-replica-locations command line option, 119
 - file-replica-partners command line option, 119
 - file-snapshot-list command line

```

    option, 121
file-volume-count command line
    option, 123
file-volume-limits command line
    option, 126
file-volume-list command line
    option, 125
hardware-list command line option,
    135
report-bandwidth command line
    option, 167
securitygroup-interface-list
    command line option, 172
securitygroup-list command line
    option, 168
securitygroup-rule-list command
    line option, 169
sshkey-list command line option, 174
ssl-list command line option, 176
subnet-list command line option, 179
summary command line option, 166
virtual-list command line option,
    198
vlan-list command line option, 189
-start <start>
    report-bandwidth command line
        option, 167
-state <state>
    rwhois-edit command line option, 168
-status <status>
    ssl-list command line option, 176
-storage-type <storage_type>
    block-volume-list command line
        option, 99
    block-volume-order command line
        option, 100
    file-volume-list command line
        option, 125
    file-volume-order command line
        option, 126
-subject-id <subject_id>
    ticket-create command line option,
        182
-subnet-id <subnet_id>
    block-subnets-assign command line
        option, 102
    block-subnets-remove command line
        option, 102
-subnet-private <subnet_private>
    virtual-create command line option,
        196
-subnet-public <subnet_public>
    virtual-create command line option,
        196
-tag <tag>
    dedicatedhost-list command line
        option, 108
    dedicatedhost-list-guests command
        line option, 111
    hardware-list command line option,
        135
    image-edit command line option, 139
    virtual-list command line option,
        198
-test
    hardware-create command line
        option, 132
    subnet-create command line option,
        177
    virtual-create command line option,
        195
-test <test>
    globalip-create command line
        option, 129
-tier <tier>
    block-replica-order command line
        option, 92
    block-snapshot-order command line
        option, 96
    block-volume-order command line
        option, 100
    file-replica-order command line
        option, 118
    file-snapshot-order command line
        option, 122
    file-volume-order command line
        option, 126
-title <title>
    ticket-create command line option,
        182
-transient
    virtual-create command line option,
        196
-transient <transient>
    virtual-list command line option,
        198
-ttl <ttl>
    dns-record-add command line option,
        112
    dns-record-edit command line
        option, 113
    dns-record-list command line
        option, 113
    hardware-dns-sync command line
        option, 138
    virtual-dns-sync command line
        option, 197
-type <type>

```

dns-record-list command line option, 113

-up, -down
autoscale-scale command line option, 88

-upgrade
block-snapshot-order command line option, 96
file-snapshot-order command line option, 122

-userdata <userdata>
autoscale-edit command line option, 89

-uuid <uuid>
loadbal-health command line option, 151

-verify
dedicatedhost-create command line option, 109
loadbal-order command line option, 156
order-place command line option, 163
order-quote command line option, 164

-virtual <virtual_identifier>
ticket-attach command line option, 184
ticket-create command line option, 182
ticket-detach command line option, 184

-vlan-private <vlan_private>
virtual-create command line option, 196

-vlan-public <vlan_public>
virtual-create command line option, 196

-wait <wait>
hardware-create command line option, 132
hardware-ready command line option, 138
virtual-create command line option, 196
virtual-ready command line option, 200

-weight <weight>
dns-record-add command line option, 112

-wrapped-dek <wrapped_dek>
image-import command line option, 140

-zone <zone>
dns-record-add command line option, 112

-D, -date-max <date_max>
event-log-get command line option, 115

-D, -disk <disk>
dedicatedhost-list command line option, 109

-D, -domain <domain>
dedicatedhost-create command line option, 109
dedicatedhost-list-guests command line option, 111
hardware-create command line option, 132
hardware-edit command line option, 133
hardware-list command line option, 134
virtual-create command line option, 195
virtual-edit command line option, 197
virtual-list command line option, 198

-F, -userfile <userfile>
autoscale-edit command line option, 89
hardware-edit command line option, 133
order-quote command line option, 164
virtual-create command line option, 196
virtual-edit command line option, 197

-H, -header <header>
cdn-origin-add command line option, 104

-H, -hostname <hostname>
dedicatedhost-create command line option, 109
dedicatedhost-list-guests command line option, 111
hardware-create command line option, 132
hardware-edit command line option, 134
hardware-list command line option, 134
virtual-create command line option, 195
virtual-edit command line option, 197
virtual-list command line option, 198

-H, -name <name>

dedicatedhost-list command line option, 108

-M, -port-max <port_max>
securitygroup-rule-add command line option, 170
securitygroup-rule-edit command line option, 171

-P, -frontProtocol <frontprotocol>
loadbal-pool-add command line option, 152
loadbal-pool-edit command line option, 153

-P, -protocol <protocol>
cdn-origin-add command line option, 104
loadbal-l7pool-add command line option, 154

-S, -public-security-group <public_security_group>
virtual-create command line option, 196

-S, -server <server>
loadbal-l7pool-add command line option, 154

-a, -a-record
hardware-dns-sync command line option, 138
virtual-dns-sync command line option, 197

-a, -api-key
user-create command line option, 187

-b, -backPort <backport>
loadbal-pool-add command line option, 153
loadbal-pool-edit command line option, 153

-b, -backend <backend>
loadbal-order command line option, 156

-b, -bucket-name <bucket_name>
cdn-origin-add command line option, 104

-c, -cache-query <cache_query>
cdn-origin-add command line option, 104

-c, -connections <connections>
loadbal-pool-add command line option, 153
loadbal-pool-edit command line option, 153

-c, -cpu <cpu>
dedicatedhost-list command line option, 108
dedicatedhost-list-guests command line option, 111
hardware-list command line option, 134
virtual-create command line option, 195
virtual-list command line option, 198

-c, -credential_id <credential_id>
object-storage-credential-delete command line option, 158

-c, -duplicate-size <duplicate_size>
block-volume-duplicate command line option, 98
file-volume-duplicate command line option, 124

-c, -new-size <new_size>
block-volume-modify command line option, 99
file-volume-modify command line option, 125

-d, -datacenter <datacenter>
block-volume-count command line option, 97
block-volume-list command line option, 99
dedicatedhost-create command line option, 109
dedicatedhost-create-options command line option, 109
dedicatedhost-list command line option, 108
file-volume-count command line option, 123
file-volume-list command line option, 124
hardware-create command line option, 132
hardware-list command line option, 134
loadbal-order command line option, 155
loadbal-order-options command line option, 156
subnet-list command line option, 179
virtual-create command line option, 195
virtual-list command line option, 198
vlan-list command line option, 189

-d, -date-min <date_min>
autoscale-logs command line option, 88
event-log-get command line option, 115

-d, -description <description>
securitygroup-create command line option, 169
securitygroup-edit command line option, 169

-d, -detail
tags-list command line option, 180

-d, -direction <direction>
securitygroup-rule-add command line option, 170
securitygroup-rule-edit command line option, 171

-d, -dry-run
tags-cleanup command line option, 181

-e, -email <email>
user-create command line option, 187

-e, -end_date <end_date>
hardware-bandwidth command line option, 130
virtual-bandwidth command line option, 194
virtual-usage command line option, 201

-e, -ethertype <ethertype>
securitygroup-rule-add command line option, 170
securitygroup-rule-edit command line option, 171

-e, -events
user-detail command line option, 185

-e, -extensions <extensions>
cdn-origin-add command line option, 104

-e, -extra <extra>
hardware-create command line option, 132

-e, -note <note>
vlan-edit command line option, 189

-e, -obj_event <obj_event>
event-log-get command line option, 115

-f, -filter <_filters>
call-api command line option, 106

-f, -flavor <flavor>
dedicatedhost-create command line option, 109
dedicatedhost-create-options command line option, 109
virtual-create command line option, 195

-f, -frontPort <frontport>
loadbal-pool-add command line option, 152
loadbal-pool-edit command line option, 153

-f, -frontend <frontend>
loadbal-order command line option, 155

-f, -in-file <in_file>
sshkey-add command line option, 173

-f, -out-file <out_file>
sshkey-print command line option, 175

-g, -tag <tag>
hardware-edit command line option, 133
virtual-create command line option, 196
virtual-edit command line option, 197

-g, -tags <tags>
autoscale-tag command line option, 89
vlan-edit command line option, 189

-h, -hardware
user-detail command line option, 185

-h, -hardware-id <hardware_id>
block-access-authorize command line option, 90
block-access-revoke command line option, 91
file-access-authorize command line option, 116
file-access-revoke command line option, 117

-i, -duplicate-iops <duplicate_iops>
block-volume-duplicate command line option, 98
file-volume-duplicate command line option, 124

-i, -include <include>
ipsec-detail command line option, 143

-i, -interface <interface>
securitygroup-interface-add command line option, 172
securitygroup-interface-remove command line option, 173

-i, -interval <interval>
loadbal-health command line option, 151

-i, -ip-address-id <ip_address_id>
block-access-authorize command line option, 90
block-access-revoke command line option, 91
file-access-authorize command line

option, 116
 file-access-revoke command line option, 117
 -i, -new-iops <new_iops>
 block-volume-modify command line option, 99
 file-volume-modify command line option, 125
 -i, -obj-id <obj_id>
 event-log-get command line option, 115
 -i, -postinstall <postinstall>
 hardware-create command line option, 132
 hardware-reload command line option, 136
 order-quote command line option, 164
 virtual-create command line option, 195
 virtual-reload command line option, 203
 -k, -key <key>
 hardware-create command line option, 132
 hardware-reload command line option, 136
 order-quote command line option, 164
 sshkey-add command line option, 173
 virtual-create command line option, 196
 virtual-reload command line option, 203
 -k, -key-name <key_name>
 tags-set command line option, 180
 -k, -keys
 user-detail command line option, 185
 -k, -label <label>
 sshkey-edit command line option, 174
 -l, -label <label>
 loadbal-order command line option, 155
 -l, -limit <limit>
 event-log-get command line option, 115
 hardware-list command line option, 135
 securitygroup-list command line option, 168
 virtual-list command line option, 198
 vlan-list command line option, 190
 -l, -location <location>
 block-replica-order command line option, 92
 file-replica-order command line option, 118
 -l, -logins
 user-detail command line option, 185
 -m, -member <member>
 loadbal-member-add command line option, 152
 loadbal-member-del command line option, 152
 -m, -memory <memory>
 dedicatedhost-list command line option, 108
 dedicatedhost-list-guests command line option, 111
 hardware-list command line option, 135
 virtual-create command line option, 195
 virtual-list command line option, 198
 -m, -method <method>
 loadbal-l7pool-add command line option, 154
 loadbal-order command line option, 156
 loadbal-pool-add command line option, 153
 loadbal-pool-edit command line option, 153
 -m, -port-min <port_min>
 securitygroup-rule-add command line option, 170
 securitygroup-rule-edit command line option, 171
 -n, -name <name>
 loadbal-l7pool-add command line option, 154
 loadbal-order command line option, 155
 securitygroup-create command line option, 169
 securitygroup-edit command line option, 169
 virtual-capture command line option, 194
 vlan-edit command line option, 189
 -n, -network <network>
 hardware-list command line option, 135
 virtual-create command line option, 196
 virtual-list command line option, 198
 -n, -network-component

<network_component>
securitygroup-interface-add
command line option, 172
securitygroup-interface-remove
command line option, 173
-n, -network-identifier, -network
<network_identifier>
ipsec-subnet-add command line
option, 147
-n, -note <note>
ipsec-translation-add command line
option, 149
ipsec-translation-update command
line option, 150
subnet-edit command line option, 178
-n, -notes <notes>
block-snapshot-create command line
option, 94
file-snapshot-create command line
option, 120
-n, -number <number>
vlan-list command line option, 189
-o, -optimize-for <optimize_for>
cdn-origin-add command line option,
104
-o, -origin-snapshot-id
<origin_snapshot_id>
block-volume-duplicate command
line option, 98
file-volume-duplicate command line
option, 124
-o, -os <os>
hardware-create command line
option, 132
virtual-create command line option,
195
-p, -backProtocol <backprotocol>
loadbal-pool-add command line
option, 152
loadbal-pool-edit command line
option, 153
-p, -password <password>
block-access-password command line
option, 91
user-create command line option, 187
-p, -permission <permission>
user-edit-permissions command line
option, 186
-p, -permissions
user-detail command line option, 185
-p, -port <port>
cdn-origin-add command line option,
104
-p, -protocol <protocol>
securitygroup-rule-add command
line option, 170
securitygroup-rule-edit command
line option, 171
-p, -summary_period <summary_period>
hardware-bandwidth command line
option, 130
virtual-bandwidth command line
option, 194
virtual-usage command line option,
201
-q, -quite_summary
hardware-bandwidth command line
option, 130
virtual-bandwidth command line
option, 194
-r, -remote-ip <remote_ip>
ipsec-translation-add command line
option, 149
ipsec-translation-update command
line option, 150
securitygroup-rule-add command
line option, 170
securitygroup-rule-edit command
line option, 171
-r, -resource-id <resource_id>
tags-set command line option, 180
-r, -retry <retry>
loadbal-health command line option,
151
-r, -router <router>
dedicatedhost-create command line
option, 109
-s, -duplicate-snapshot-size
<duplicate_snapshot_size>
block-volume-duplicate command
line option, 98
file-volume-duplicate command line
option, 124
-s, -private-security-group
<private_security_group>
virtual-create command line option,
196
-s, -remote-group <remote_group>
securitygroup-rule-add command
line option, 170
securitygroup-rule-edit command
line option, 171
-s, -server <server>
securitygroup-interface-add
command line option, 172
securitygroup-interface-remove
command line option, 173
-s, -size <size>


```

hardware-create command line
    option, 132
-s, -snapshot-id <snapshot_id>
    block-snapshot-restore command
        line option, 97
    file-snapshot-restore command line
        option, 122
-s, -snapshot-schedule
    <snapshot_schedule>
    block-replica-order command line
        option, 92
    file-replica-order command line
        option, 118
-s, -start_date <start_date>
    hardware-bandwidth command line
        option, 130
    virtual-bandwidth command line
        option, 194
    virtual-usage command line option,
        201
-s, -static-ip <static_ip>
    ipsec-translation-add command line
        option, 149
    ipsec-translation-update command
        line option, 150
-s, -sticky
    loadbal-l7pool-add command line
        option, 155
    loadbal-pool-add command line
        option, 153
    loadbal-pool-edit command line
        option, 153
-s, -subnet <subnet>
    loadbal-order command line option,
        156
-s, -subnet-id <subnet_id>
    file-access-authorize command line
        option, 116
    file-access-revoke command line
        option, 117
    ipsec-subnet-add command line
        option, 147
    ipsec-subnet-remove command line
        option, 148
-t, -duplicate-tier <duplicate_tier>
    block-volume-duplicate command
        line option, 98
    file-volume-duplicate command line
        option, 124
-t, -new-tier <new_tier>
    block-volume-modify command line
        option, 99
    file-volume-modify command line
        option, 125
-t, -obj-type <obj_type>
    event-log-get command line option,
        115
-t, -origin-type <origin_type>
    cdn-origin-add command line option,
        104
-t, -subnet-type <subnet_type>
    subnet-list command line option, 179
-t, -subnet-type, -type <subnet_type>
    ipsec-subnet-add command line
        option, 147
    ipsec-subnet-remove command line
        option, 148
-t, -tags <tags>
    subnet-edit command line option, 178
    tags-set command line option, 180
-t, -template <template>
    dedicatedhost-create command line
        option, 109
    hardware-create command line
        option, 132
    user-create command line option, 187
    user-edit-details command line
        option, 186
    virtual-create command line option,
        196
-t, -timeout <timeout>
    loadbal-health command line option,
        151
-t, -translation-id <translation_id>
    ipsec-translation-remove command
        line option, 149
    ipsec-translation-update command
        line option, 150
-t, -valid_type <valid_type>
    virtual-usage command line option,
        201
-u, -from-user <from_user>
    user-create command line option, 187
    user-edit-permissions command line
        option, 186
-u, -url <url>
    loadbal-health command line option,
        151
-u, -userdata <userdata>
    hardware-edit command line option,
        134
    order-quote command line option, 164
    virtual-create command line option,
        196
    virtual-edit command line option,
        197
-u, -username <username>
    block-volume-list command line

```

option, 99
 file-volume-list command line
 option, 124
 -v, -virtual
 user-detail command line option, 185
 -v, -virtual-id <virtual_id>
 block-access-authorize command
 line option, 90
 block-access-revoke command line
 option, 91
 file-access-authorize command line
 option, 116
 file-access-revoke command line
 option, 117
 -v6, -ipv6
 globalip-create command line
 option, 129
 -w, -weight <weight>
 loadbal-member-add command line
 option, 152
 -x, -sslCert <sslcert>
 loadbal-pool-add command line
 option, 153
 loadbal-pool-edit command line
 option, 154
 -z, -utc-offset <utc_offset>
 event-log-get command line option,
 115

A

ACCESS_ID
 block-access-password command line
 option, 91
 block-subnets-assign command line
 option, 102
 block-subnets-list command line
 option, 101
 block-subnets-remove command line
 option, 102
 account-cancel-item command line
 option
 IDENTIFIER, 87
 account-event-detail command line
 option
 -ack, 86
 IDENTIFIER, 86
 account-events command line option
 -ack-all, 85
 account-invoice-detail command line
 option
 -details, 86
 IDENTIFIER, 86
 account-invoices command line option
 -all, 86

-closed, 86
 -limit <limit>, 86
 account-item-detail command line
 option
 IDENTIFIER, 87
 AccountManager (class in Soft-
 Layer.managers.account), 8
 ack_event() (SoftLayer.managers.account.AccountManager
 method), 8
 add_certificate() (Soft-
 Layer.managers.ssl.SSLManager method),
 59
 add_event_filter() (Soft-
 Layer.managers.account.AccountManager
 static method), 8
 add_global_ip() (Soft-
 Layer.managers.network.NetworkManager
 method), 47
 add_internal_subnet() (Soft-
 Layer.managers.ipsec.IPSECManger
 method), 40
 add_key() (SoftLayer.managers.sshkey.SshKeyManager
 method), 58
 add_lb_17_pool() (Soft-
 Layer.managers.load_balancer.LoadBalancerManager
 method), 44
 add_lb_listener() (Soft-
 Layer.managers.load_balancer.LoadBalancerManager
 method), 44
 add_lb_member() (Soft-
 Layer.managers.load_balancer.LoadBalancerManager
 method), 44
 add_locations() (Soft-
 Layer.managers.image.ImageManager
 method), 38
 add_origin() (Soft-
 Layer.managers.cdn.CDNManager method),
 17
 add_remote_subnet() (Soft-
 Layer.managers.ipsec.IPSECManger
 method), 40
 add_securitygroup_rule() (Soft-
 Layer.managers.network.NetworkManager
 method), 47
 add_securitygroup_rules() (Soft-
 Layer.managers.network.NetworkManager
 method), 48
 add_service_subnet() (Soft-
 Layer.managers.ipsec.IPSECManger
 method), 41
 add_standard_firewall() (Soft-
 Layer.managers.firewall.FirewallManager
 method), 31
 add_subnet() (Soft-

- Layer.managers.network.NetworkManager method*), 48
- add_vlan_firewall() (Soft-
Layer.managers.firewall.FirewallManager method), 31
- apply_configuration() (Soft-
Layer.managers.ipsec.IPSECManger method), 41
- assign_global_ip() (Soft-
Layer.managers.network.NetworkManager method), 48
- assign_subnets_to_acl() (Soft-
Layer.managers.block.BlockStorageManager method), 11
- attach_hardware() (Soft-
Layer.managers.ticket.TicketManager method), 60
- attach_securitygroup_component() (Soft-
Layer.managers.network.NetworkManager method), 48
- attach_securitygroup_components() (Soft-
Layer.managers.network.NetworkManager method), 48
- attach_virtual_server() (Soft-
Layer.managers.ticket.TicketManager method), 60
- authenticate_with_password() (Soft-
Layer.BaseClient method), 76
- authorize_host_to_volume() (Soft-
Layer.managers.block.BlockStorageManager method), 11
- authorize_host_to_volume() (Soft-
Layer.managers.file.FileStorageManager method), 25
- autoscale-detail command line option
IDENTIFIER, 88
- autoscale-edit command line option
-cpu <cpu>, 89
-max <maximum>, 89
-memory <memory>, 89
-min <minimum>, 89
-name <name>, 89
-userdata <userdata>, 89
-F, -userfile <userfile>, 89
IDENTIFIER, 89
- autoscale-logs command line option
-d, -date-min <date_min>, 88
IDENTIFIER, 88
- autoscale-scale command line option
-amount <amount>, 88
-by, -to, 88
-up, -down, 88
IDENTIFIER, 88
- autoscale-tag command line option
-g, -tags <tags>, 89
IDENTIFIER, 89
- AutoScaleManager (class in Soft-
Layer.managers.autoscale), 9
- ## B
- BaseClient (class in SoftLayer), 75
- BasicAuthentication (class in SoftLayer), 77
- block-access-authorize command line
option
-ip-address <ip_address>, 90
-h, -hardware-id <hardware_id>, 90
-i, -ip-address-id <ip_address_id>, 90
-v, -virtual-id <virtual_id>, 90
VOLUME_ID, 90
- block-access-list command line option
-columns <columns>, 90
-sortby <sortby>, 90
VOLUME_ID, 90
- block-access-password command line
option
-p, -password <password>, 91
ACCESS_ID, 91
- block-access-revoke command line
option
-ip-address <ip_address>, 91
-h, -hardware-id <hardware_id>, 91
-i, -ip-address-id <ip_address_id>, 91
-v, -virtual-id <virtual_id>, 91
VOLUME_ID, 91
- block-replica-failback command line
option
VOLUME_ID, 92
- block-replica-failover command line
option
-replicant-id <replicant_id>, 92
VOLUME_ID, 92
- block-replica-locations command line
option
-columns <columns>, 93
-sortby <sortby>, 93
VOLUME_ID, 93
- block-replica-order command line
option
-os-type <os_type>, 92
-tier <tier>, 92
-l, -location <location>, 92
-s, -snapshot-schedule
<snapshot_schedule>, 92
VOLUME_ID, 92
- block-replica-partners command line
option

-columns <columns>, 93
-sortby <sortby>, 93
VOLUME_ID, 93
block-snapshot-cancel command line
 option
 -immediate, 94
 -reason <reason>, 94
 VOLUME_ID, 94
block-snapshot-create command line
 option
 -n, -notes <notes>, 94
 VOLUME_ID, 94
block-snapshot-delete command line
 option
 SNAPSHOT_ID, 94
block-snapshot-disable command line
 option
 -schedule-type <schedule_type>, 95
 VOLUME_ID, 95
block-snapshot-enable command line
 option
 -day-of-week <day_of_week>, 95
 -hour <hour>, 95
 -minute <minute>, 95
 -retention-count <retention_count>, 95
 -schedule-type <schedule_type>, 95
 VOLUME_ID, 95
block-snapshot-list command line
 option
 -columns <columns>, 96
 -sortby <sortby>, 96
 VOLUME_ID, 96
block-snapshot-order command line
 option
 -capacity <capacity>, 96
 -tier <tier>, 96
 -upgrade, 96
 VOLUME_ID, 96
block-snapshot-restore command line
 option
 -s, -snapshot-id <snapshot_id>, 97
 VOLUME_ID, 97
block-snapshot-schedule-list command
 line option
 VOLUME_ID, 95
block-subnets-assign command line
 option
 -subnet-id <subnet_id>, 102
 ACCESS_ID, 102
block-subnets-list command line option
 ACCESS_ID, 101
block-subnets-remove command line
 option
 -subnet-id <subnet_id>, 102
 ACCESS_ID, 102
block-volume-cancel command line
 option
 -immediate, 97
 -reason <reason>, 97
 VOLUME_ID, 97
block-volume-convert command line
 option
 VOLUME_ID, 103
block-volume-count command line option
 -sortby <sortby>, 97
 -d, -datacenter <datacenter>, 97
block-volume-detail command line
 option
 VOLUME_ID, 98
block-volume-duplicate command line
 option
 -billing <billing>, 98
 -dependent-duplicate
 <dependent_duplicate>, 98
 -c, -duplicate-size
 <duplicate_size>, 98
 -i, -duplicate-iops
 <duplicate_iops>, 98
 -o, -origin-snapshot-id
 <origin_snapshot_id>, 98
 -s, -duplicate-snapshot-size
 <duplicate_snapshot_size>, 98
 -t, -duplicate-tier
 <duplicate_tier>, 98
 ORIGIN_VOLUME_ID, 98
block-volume-limits command line
 option
 -sortby <sortby>, 101
block-volume-list command line option
 -columns <columns>, 99
 -sortby <sortby>, 99
 -storage-type <storage_type>, 99
 -d, -datacenter <datacenter>, 99
 -u, -username <username>, 99
block-volume-modify command line
 option
 -c, -new-size <new_size>, 99
 -i, -new-iops <new_iops>, 99
 -t, -new-tier <new_tier>, 99
 VOLUME_ID, 100
block-volume-order command line option
 -billing <billing>, 100
 -iops <iops>, 100
 -location <location>, 100
 -os-type <os_type>, 100
 -service-offering
 <service_offering>, 100

-size <size>, 100
 -snapshot-size <snapshot_size>, 100
 -storage-type <storage_type>, 100
 -tier <tier>, 100
 block-volume-refresh command line option
 SNAPSHOT_ID, 103
 VOLUME_ID, 103
 block-volume-set-lun-id command line option
 LUN_ID, 101
 VOLUME_ID, 101
 BlockStorageManager (class in SoftLayer.managers.block), 11
 build_filter() (SoftLayer.managers.event_log.EventLogManager static method), 25

C

call() (SoftLayer.BaseClient method), 76
 call-api command line option
 -id <id>, 106
 -json-filter <json_filter>, 106
 -limit <limit>, 106
 -mask <mask>, 106
 -offset <offset>, 106
 -output-python, -no-output-python, 106
 -f, -filter <_filters>, 106
 METHOD, 107
 PARAMETERS, 107
 SERVICE, 107
 cancel_block_volume() (SoftLayer.managers.block.BlockStorageManager method), 11
 cancel_file_volume() (SoftLayer.managers.file.FileStorageManager method), 26
 cancel_firewall() (SoftLayer.managers.firewall.FirewallManager method), 32
 cancel_global_ip() (SoftLayer.managers.network.NetworkManager method), 48
 cancel_guests() (SoftLayer.managers.dedicated_host.DedicatedHostManager method), 19
 cancel_hardware() (SoftLayer.managers.hardware.HardwareManager method), 33
 cancel_host() (SoftLayer.managers.dedicated_host.DedicatedHostManager method), 20
 cancel_instance() (SoftLayer.managers.vs.VSManager method), 62
 cancel_item() (SoftLayer.managers.account.AccountManager method), 8
 cancel_lbaas() (SoftLayer.managers.load_balancer.LoadBalancerManager method), 44
 cancel_snapshot_space() (SoftLayer.managers.block.BlockStorageManager method), 11
 cancel_snapshot_space() (SoftLayer.managers.file.FileStorageManager method), 26
 cancel_subnet() (SoftLayer.managers.network.NetworkManager method), 48
 cancel_volume() (SoftLayer.managers.block.BlockStorageManager method), 12
 cancel_volume() (SoftLayer.managers.file.FileStorageManager method), 26
 CapacityManager (class in SoftLayer.managers.vs_capacity), 71
 capture() (SoftLayer.managers.vs.VSManager method), 62
 cdn-detail command line option
 -history <history>, 103
 UNIQUE_ID, 103
 cdn-list command line option
 -sortby <sortby>, 104
 cdn-origin-add command line option
 -H, -header <header>, 104
 -P, -protocol <protocol>, 104
 -b, -bucket-name <bucket_name>, 104
 -c, -cache-query <cache_query>, 104
 -e, -extensions <extensions>, 104
 -o, -optimize-for <optimize_for>, 104
 -p, -port <port>, 104
 -t, -origin-type <origin_type>, 104
 ORIGIN, 104
 PATH, 104
 UNIQUE_ID, 104
 cdn-origin-list command line option
 UNIQUE_ID, 105
 cdn-origin-remove command line option
 ORIGIN_PATH, 105
 UNIQUE_ID, 105
 cdn-purge command line option
 PATH, 105
 UNIQUE_ID, 105

CDNManager (class in *SoftLayer.managers.cdn*), 17
 change_port_speed() (SoftLayer.managers.hardware.HardwareManager method), 34
 change_port_speed() (SoftLayer.managers.vs.VSManager method), 63
 Client() (in module *SoftLayer*), 77
 CONTEXT_ID
 ipsec-configure command line option, 146
 ipsec-detail command line option, 143
 ipsec-subnet-add command line option, 147
 ipsec-subnet-remove command line option, 148
 ipsec-translation-add command line option, 149
 ipsec-translation-remove command line option, 149
 ipsec-translation-update command line option, 150
 ipsec-update command line option, 145
 convert_dep_dupe() (SoftLayer.managers.block.BlockStorageManager method), 12
 convert_dep_dupe() (SoftLayer.managers.file.FileStorageManager method), 26
 create() (SoftLayer.managers.vs_capacity.CapacityManager method), 71
 create() (SoftLayer.managers.vs_placement.PlacementManager method), 72
 create_client_from_env() (in module *SoftLayer*), 77
 create_guest() (SoftLayer.managers.vs_capacity.CapacityManager method), 71
 create_instance() (SoftLayer.managers.vs.VSManager method), 63
 create_instances() (SoftLayer.managers.vs.VSManager method), 65
 create_or_update_lun_id() (SoftLayer.managers.block.BlockStorageManager method), 12
 create_record() (SoftLayer.managers.dns.DNSManager method), 22
 create_record_mx() (SoftLayer.managers.dns.DNSManager method), 22
 create_record_ptr() (SoftLayer.managers.dns.DNSManager method), 22
 create_record_srv() (SoftLayer.managers.dns.DNSManager method), 22
 create_remote_subnet() (SoftLayer.managers.ipsec.IPSECManger method), 41
 create_securitygroup() (SoftLayer.managers.network.NetworkManager method), 49
 create_snapshot() (SoftLayer.managers.block.BlockStorageManager method), 12
 create_snapshot() (SoftLayer.managers.file.FileStorageManager method), 26
 create_ticket() (SoftLayer.managers.ticket.TicketManager method), 61
 create_translation() (SoftLayer.managers.ipsec.IPSECManger method), 41
 create_zone() (SoftLayer.managers.dns.DNSManager method), 23
D
 DATA
 dns-record-add command line option, 112
 deauthorize_host_to_volume() (SoftLayer.managers.block.BlockStorageManager method), 12
 deauthorize_host_to_volume() (SoftLayer.managers.file.FileStorageManager method), 26
 dedicatedhost-cancel command line option
 IDENTIFIER, 110
 dedicatedhost-cancel-guests command line option
 IDENTIFIER, 110
 dedicatedhost-create command line option
 -billing <billing>, 109
 -export <export>, 109
 -verify, 109
 -D, -domain <domain>, 109
 -H, -hostname <hostname>, 109
 -d, -datacenter <datacenter>, 109
 -f, -flavor <flavor>, 109

- r, -router <router>, 109
- t, -template <template>, 109
- dedicatedhost-create-options command line option
 - d, -datacenter <datacenter>, 109
 - f, -flavor <flavor>, 109
- dedicatedhost-detail command line option
 - guests, 110
 - price, 110
 - IDENTIFIER, 110
- dedicatedhost-list command line option
 - columns <columns>, 108
 - sortby <sortby>, 108
 - tag <tag>, 108
 - D, -disk <disk>, 109
 - H, -name <name>, 108
 - c, -cpu <cpu>, 108
 - d, -datacenter <datacenter>, 108
 - m, -memory <memory>, 108
- dedicatedhost-list-guests command line option
 - columns <columns>, 111
 - sortby <sortby>, 111
 - tag <tag>, 111
 - D, -domain <domain>, 111
 - H, -hostname <hostname>, 111
 - c, -cpu <cpu>, 111
 - m, -memory <memory>, 111
 - IDENTIFIER, 111
- DedicatedHostManager (class in *SoftLayer.managers.dedicated_host*), 19
- del_lb_l7_pool() (*SoftLayer.managers.load_balancer.LoadBalancerManager* method), 44
- delete() (*SoftLayer.managers.vs_placement.PlacementManager* method), 73
- delete_image() (*SoftLayer.managers.image.ImageManager* method), 38
- delete_key() (*SoftLayer.managers.sshkey.SshKeyManager* method), 58
- delete_lb_member() (*SoftLayer.managers.load_balancer.LoadBalancerManager* method), 45
- delete_record() (*SoftLayer.managers.dns.DNSManager* method), 23
- delete_remote_subnet() (*SoftLayer.managers.ipsec.IPSECManager* method), 41
- delete_securitygroup() (*SoftLayer.managers.network.NetworkManager* method), 49
- delete_snapshot() (*SoftLayer.managers.block.BlockStorageManager* method), 12
- delete_snapshot() (*SoftLayer.managers.file.FileStorageManager* method), 27
- delete_zone() (*SoftLayer.managers.dns.DNSManager* method), 23
- detach_hardware() (*SoftLayer.managers.ticket.TicketManager* method), 61
- detach_securitygroup_component() (*SoftLayer.managers.network.NetworkManager* method), 49
- detach_securitygroup_components() (*SoftLayer.managers.network.NetworkManager* method), 49
- detach_virtual_server() (*SoftLayer.managers.ticket.TicketManager* method), 61
- details() (*SoftLayer.managers.autoscale.AutoScaleManager* method), 10
- disable_snapshots() (*SoftLayer.managers.block.BlockStorageManager* method), 12
- disable_snapshots() (*SoftLayer.managers.file.FileStorageManager* method), 27
- dns-import command line option
 - dry-run, 111
 - ZONEFILE, 111
- dns-record-add command line option
 - port <port>, 112
 - priority <priority>, 112
 - protocol <protocol>, 112
 - service <service>, 112
 - ttl <ttl>, 112
 - weight <weight>, 112
 - zone <zone>, 112
 - DATA, 112
 - RECORD, 112
 - RECORD_TYPE, 112
- dns-record-edit command line option
 - by-id <by_id>, 113
 - by-record <by_record>, 113
 - data <data>, 113
 - ttl <ttl>, 113
 - ZONE_ID, 113
- dns-record-list command line option
 - data <data>, 113
 - record <record>, 113
 - ttl <ttl>, 113

- type <type>, 113
 - ZONE, 113
 - dns-record-remove command line option
 - RECORD_ID, 114
 - dns-zone-create command line option
 - ZONE, 114
 - dns-zone-delete command line option
 - ZONE, 114
 - dns-zone-print command line option
 - ZONE, 115
 - DNSManager (*class in SoftLayer.managers.dns*), 22
 - dump_zone() (*SoftLayer.managers.dns.DNSManager method*), 23
- ## E
- edit() (*SoftLayer.managers.autoscale.AutoScaleManager method*), 10
 - edit() (*SoftLayer.managers.hardware.HardwareManager method*), 34
 - edit() (*SoftLayer.managers.image.ImageManager method*), 38
 - edit() (*SoftLayer.managers.network.NetworkManager method*), 49
 - edit() (*SoftLayer.managers.vs.VSManager method*), 65
 - edit_certificate() (*SoftLayer.managers.ssl.SSLManager method*), 59
 - edit_dedicated_fw_rules() (*SoftLayer.managers.firewall.FirewallManager method*), 32
 - edit_key() (*SoftLayer.managers.sshkey.SshKeyManager method*), 58
 - edit_note_subnet() (*SoftLayer.managers.network.NetworkManager method*), 49
 - edit_record() (*SoftLayer.managers.dns.DNSManager method*), 23
 - edit_rwhois() (*SoftLayer.managers.network.NetworkManager method*), 49
 - edit_securitygroup() (*SoftLayer.managers.network.NetworkManager method*), 49
 - edit_securitygroup_rule() (*SoftLayer.managers.network.NetworkManager method*), 50
 - edit_standard_fw_rules() (*SoftLayer.managers.firewall.FirewallManager method*), 32
 - edit_zone() (*SoftLayer.managers.dns.DNSManager method*), 23
 - enable_snapshots() (*SoftLayer.managers.block.BlockStorageManager method*), 13
 - enable_snapshots() (*SoftLayer.managers.file.FileStorageManager method*), 27
 - end_date (*SoftLayer.managers.cdn.CDNManager attribute*), 18
 - ENDPOINT_ID
 - subnet-create command line option, 178
 - event-log-get command line option
 - metadata, -no-metadata, 115
 - D, -date-max <date_max>, 115
 - d, -date-min <date_min>, 115
 - e, -obj-event <obj_event>, 115
 - i, -obj-id <obj_id>, 115
 - l, -limit <limit>, 115
 - t, -obj-type <obj_type>, 115
 - z, -utc-offset <utc_offset>, 115
 - EventManager (*class in SoftLayer.managers.event_log*), 24
 - export_image_to_uri() (*SoftLayer.managers.image.ImageManager method*), 38
- ## F
- failback_from_replicant() (*SoftLayer.managers.block.BlockStorageManager method*), 13
 - failback_from_replicant() (*SoftLayer.managers.file.FileStorageManager method*), 27
 - failover_to_replicant() (*SoftLayer.managers.block.BlockStorageManager method*), 13
 - failover_to_replicant() (*SoftLayer.managers.file.FileStorageManager method*), 27
 - file-access-authorize command line option
 - ip-address <ip_address>, 116
 - h, -hardware-id <hardware_id>, 116
 - i, -ip-address-id <ip_address_id>, 116
 - s, -subnet-id <subnet_id>, 116
 - v, -virtual-id <virtual_id>, 116
 - VOLUME_ID, 116
 - file-access-list command line option
 - columns <columns>, 117
 - sortby <sortby>, 117
 - VOLUME_ID, 117
 - file-access-revoke command line option
 - ip-address <ip_address>, 117

-h, -hardware-id <hardware_id>, 117
 -i, -ip-address-id <ip_address_id>, 117
 -s, -subnet-id <subnet_id>, 117
 -v, -virtual-id <virtual_id>, 117
 VOLUME_ID, 117
 file-replica-failback command line option
 VOLUME_ID, 118
 file-replica-failover command line option
 -replicant-id <replicant_id>, 118
 VOLUME_ID, 118
 file-replica-locations command line option
 -columns <columns>, 119
 -sortby <sortby>, 119
 VOLUME_ID, 119
 file-replica-order command line option
 -tier <tier>, 118
 -l, -location <location>, 118
 -s, -snapshot-schedule <snapshot_schedule>, 118
 VOLUME_ID, 118
 file-replica-partners command line option
 -columns <columns>, 119
 -sortby <sortby>, 119
 VOLUME_ID, 119
 file-snapshot-cancel command line option
 -immediate, 120
 -reason <reason>, 120
 VOLUME_ID, 120
 file-snapshot-create command line option
 -n, -notes <notes>, 120
 VOLUME_ID, 120
 file-snapshot-delete command line option
 SNAPSHOT_ID, 120
 file-snapshot-disable command line option
 -schedule-type <schedule_type>, 121
 VOLUME_ID, 121
 file-snapshot-enable command line option
 -day-of-week <day_of_week>, 121
 -hour <hour>, 121
 -minute <minute>, 121
 -retention-count <retention_count>, 121
 -schedule-type <schedule_type>, 121
 VOLUME_ID, 121
 file-snapshot-list command line option
 -columns <columns>, 121
 -sortby <sortby>, 121
 VOLUME_ID, 122
 file-snapshot-order command line option
 -capacity <capacity>, 122
 -tier <tier>, 122
 -upgrade, 122
 VOLUME_ID, 122
 file-snapshot-restore command line option
 -s, -snapshot-id <snapshot_id>, 122
 VOLUME_ID, 122
 file-snapshot-schedule-list command line option
 VOLUME_ID, 127
 file-volume-cancel command line option
 -immediate, 123
 -reason <reason>, 123
 VOLUME_ID, 123
 file-volume-convert command line option
 VOLUME_ID, 127
 file-volume-count command line option
 -sortby <sortby>, 123
 -d, -datacenter <datacenter>, 123
 file-volume-detail command line option
 VOLUME_ID, 123
 file-volume-duplicate command line option
 -billing <billing>, 124
 -dependent-duplicate <dependent_duplicate>, 124
 -c, -duplicate-size <duplicate_size>, 124
 -i, -duplicate-iops <duplicate_iops>, 124
 -o, -origin-snapshot-id <origin_snapshot_id>, 124
 -s, -duplicate-snapshot-size <duplicate_snapshot_size>, 124
 -t, -duplicate-tier <duplicate_tier>, 124
 ORIGIN_VOLUME_ID, 124
 file-volume-limits command line option
 -sortby <sortby>, 126
 file-volume-list command line option
 -columns <columns>, 125
 -sortby <sortby>, 125
 -storage-type <storage_type>, 125
 -d, -datacenter <datacenter>, 124
 -u, -username <username>, 124
 file-volume-modify command line option

- c, -new-size <new_size>, 125
 - i, -new-iops <new_iops>, 125
 - t, -new-tier <new_tier>, 125
 - VOLUME_ID, 125
 - file-volume-order command line option
 - billing <billing>, 126
 - iops <iops>, 126
 - location <location>, 126
 - service-offering <service_offering>, 126
 - size <size>, 126
 - snapshot-size <snapshot_size>, 126
 - storage-type <storage_type>, 126
 - tier <tier>, 126
 - file-volume-refresh command line option
 - SNAPSHOT_ID, 127
 - VOLUME_ID, 127
 - FileStorageManager (class in SoftLayer.managers.file), 25
 - filter_outlet_packages() (SoftLayer.managers.ordering.OrderingManager static method), 53
 - firewall-add command line option
 - firewall-type <firewall_type>, 127
 - ha, -high-availability, 127
 - TARGET, 128
 - firewall-cancel command line option IDENTIFIER, 128
 - firewall-detail command line option IDENTIFIER, 128
 - firewall-edit command line option IDENTIFIER, 128
 - FirewallManager (class in SoftLayer.managers.firewall), 31
- ## G
- generate_order() (SoftLayer.managers.ordering.OrderingManager method), 53
 - generate_order_template() (SoftLayer.managers.ordering.OrderingManager method), 53
 - get() (SoftLayer.managers.metadata.MetadataManager method), 46
 - get_account_billing_items() (SoftLayer.managers.account.AccountManager method), 8
 - get_adc() (SoftLayer.managers.load_balancer.LoadBalancerManager method), 45
 - get_adcs() (SoftLayer.managers.load_balancer.LoadBalancerManager method), 45
 - get_all_rules() (SoftLayer.managers.vs_placement.PlacementManager method), 73
 - get_available_routers() (SoftLayer.managers.vs_capacity.CapacityManager method), 71
 - get_backend_router_id_from_hostname() (SoftLayer.managers.vs_placement.PlacementManager method), 73
 - get_bandwidth_allocation() (SoftLayer.managers.hardware.HardwareManager method), 34
 - get_bandwidth_allocation() (SoftLayer.managers.vs.VSManager method), 66
 - get_bandwidth_data() (SoftLayer.managers.hardware.HardwareManager method), 34
 - get_bandwidth_data() (SoftLayer.managers.vs.VSManager method), 66
 - get_billing_item() (SoftLayer.managers.account.AccountManager method), 9
 - get_billing_items() (SoftLayer.managers.account.AccountManager method), 9
 - get_block_volume_access_list() (SoftLayer.managers.block.BlockStorageManager method), 13
 - get_block_volume_details() (SoftLayer.managers.block.BlockStorageManager method), 13
 - get_block_volume_snapshot_list() (SoftLayer.managers.block.BlockStorageManager method), 13
 - get_cancellation_reasons() (SoftLayer.managers.hardware.HardwareManager method), 35
 - get_cdn() (SoftLayer.managers.cdn.CDNManager method), 18
 - get_certificate() (SoftLayer.managers.ssl.SSLManager method), 59
 - get_create_options() (SoftLayer.managers.dedicated_host.DedicatedHostManager method), 20
 - get_create_options() (SoftLayer.managers.hardware.HardwareManager method), 35
 - get_create_options() (SoftLayer.managers.vs.VSManager method), 66
 - get_create_options() (SoftLayer.managers.vs_capacity.CapacityManager method), 72

<code>get_dedicated_fw_rules()</code>	(SoftLayer.managers.firewall.FirewallManager method), 32	<code>Layer.managers.ordering.OrderingManager method), 53</code>
<code>get_dedicated_package()</code>	(SoftLayer.managers.firewall.FirewallManager method), 32	<code>get_key() (SoftLayer.managers.sshkey.SshKeyManager method), 58</code>
<code>get_event()</code>	(SoftLayer.managers.account.AccountManager method), 9	<code>get_lb() (SoftLayer.managers.load_balancer.LoadBalancerManager method), 45</code>
<code>get_event_log_types()</code>	(SoftLayer.managers.event_log.EventLogManager method), 25	<code>get_lbaas() (SoftLayer.managers.load_balancer.LoadBalancerManager method), 45</code>
<code>get_event_logs()</code>	(SoftLayer.managers.event_log.EventLogManager method), 25	<code>get_lbaas_by_name() (SoftLayer.managers.load_balancer.LoadBalancerManager method), 45</code>
<code>get_event_logs_by_request_id()</code>	(SoftLayer.managers.network.NetworkManager method), 50	<code>get_lbaas_uuid_id() (SoftLayer.managers.load_balancer.LoadBalancerManager method), 45</code>
<code>get_file_volume_access_list()</code>	(SoftLayer.managers.file.FileStorageManager method), 28	<code>get_local_disks() (SoftLayer.managers.vs.VSManager method), 66</code>
<code>get_file_volume_details()</code>	(SoftLayer.managers.file.FileStorageManager method), 28	<code>get_location_id() (SoftLayer.managers.ordering.OrderingManager method), 54</code>
<code>get_file_volume_snapshot_list()</code>	(SoftLayer.managers.file.FileStorageManager method), 28	<code>get_locations_list() (SoftLayer.managers.image.ImageManager method), 39</code>
<code>get_firewalls()</code>	(SoftLayer.managers.firewall.FirewallManager method), 32	<code>get_logs() (SoftLayer.managers.autoscale.AutoScaleManager method), 10</code>
<code>get_hard_drives()</code>	(SoftLayer.managers.hardware.HardwareManager method), 35	<code>get_nas_credentials() (SoftLayer.managers.network.NetworkManager method), 50</code>
<code>get_hardware()</code>	(SoftLayer.managers.hardware.HardwareManager method), 35	<code>get_object() (SoftLayer.managers.vs_capacity.CapacityManager method), 72</code>
<code>get_host()</code>	(SoftLayer.managers.dedicated_host.DedicatedHostManager method), 20	<code>get_object() (SoftLayer.managers.vs_placement.PlacementManager method), 73</code>
<code>get_image()</code>	(SoftLayer.managers.image.ImageManager method), 39	<code>get_only_active_packages() (SoftLayer.managers.ordering.OrderingManager static method), 54</code>
<code>get_instance()</code>	(SoftLayer.managers.vs.VSManager method), 66	<code>get_order_container() (SoftLayer.managers.ordering.OrderingManager method), 54</code>
<code>get_invoices()</code>	(SoftLayer.managers.account.AccountManager method), 9	<code>get_origins() (SoftLayer.managers.cdn.CDNManager method), 18</code>
<code>get_ip_by_address()</code>	(SoftLayer.managers.network.NetworkManager method), 50	<code>get_package_by_key() (SoftLayer.managers.ordering.OrderingManager method), 54</code>
<code>get_item_capacity()</code>	(SoftLayer.managers.ordering.OrderingManager method), 53	<code>get_package_by_type() (SoftLayer.managers.ordering.OrderingManager method), 54</code>
<code>get_item_price_id()</code>	(SoftLayer.managers.ordering.OrderingManager static method), 53	<code>get_package_id_by_type() (SoftLayer.managers.ordering.OrderingManager method), 54</code>
<code>get_item_prices()</code>	(SoftLayer.managers.ordering.OrderingManager method), 53	<code>get_packages_of_type() (SoftLayer.managers.ordering.OrderingManager method), 54</code>

<i>method</i>), 54		<code>get_securitygroup()</code>	(Soft- <i>Layer.managers.network.NetworkManager</i> <i>method</i>), 50
<code>get_policy()</code>	(Soft- <i>Layer.managers.autoscale.AutoScaleManager</i> <i>method</i>), 10	<code>get_standard_fw_rules()</code>	(Soft- <i>Layer.managers.firewall.FirewallManager</i> <i>method</i>), 32
<code>get_portable_storage()</code>	(Soft- <i>Layer.managers.vs.VSManager</i> <i>method</i>), 67	<code>get_standard_package()</code>	(Soft- <i>Layer.managers.firewall.FirewallManager</i> <i>method</i>), 32
<code>get_preset_by_key()</code>	(Soft- <i>Layer.managers.ordering.OrderingManager</i> <i>method</i>), 54	<code>get_storage_credentials()</code>	(Soft- <i>Layer.managers.hardware.HardwareManager</i> <i>method</i>), 35
<code>get_preset_prices()</code>	(Soft- <i>Layer.managers.ordering.OrderingManager</i> <i>method</i>), 55	<code>get_storage_credentials()</code>	(Soft- <i>Layer.managers.vs.VSManager</i> <i>method</i>), 67
<code>get_price_id_list()</code>	(Soft- <i>Layer.managers.ordering.OrderingManager</i> <i>method</i>), 55	<code>get_storage_details()</code>	(Soft- <i>Layer.managers.hardware.HardwareManager</i> <i>method</i>), 35
<code>get_quote_details()</code>	(Soft- <i>Layer.managers.ordering.OrderingManager</i> <i>method</i>), 55	<code>get_storage_details()</code>	(Soft- <i>Layer.managers.vs.VSManager</i> <i>method</i>), 67
<code>get_quotes()</code>	(Soft- <i>Layer.managers.ordering.OrderingManager</i> <i>method</i>), 55	<code>get_storage_locations()</code>	(Soft- <i>Layer.managers.image.ImageManager</i> <i>method</i>), 39
<code>get_record()</code>	(Soft- <i>Layer.managers.dns.DNSManager</i> <i>method</i>), 23	<code>get_subnet()</code>	(Soft- <i>Layer.managers.network.NetworkManager</i> <i>method</i>), 50
<code>get_records()</code>	(Soft- <i>Layer.managers.dns.DNSManager</i> <i>method</i>), 23	<code>get_subnets_in_acl()</code>	(Soft- <i>Layer.managers.block.BlockStorageManager</i> <i>method</i>), 14
<code>get_replication_locations()</code>	(Soft- <i>Layer.managers.block.BlockStorageManager</i> <i>method</i>), 14	<code>get_summary()</code>	(Soft- <i>Layer.managers.account.AccountManager</i> <i>method</i>), 9
<code>get_replication_locations()</code>	(Soft- <i>Layer.managers.file.FileStorageManager</i> <i>method</i>), 28	<code>get_summary_data_usage()</code>	(Soft- <i>Layer.managers.vs.VSManager</i> <i>method</i>), 67
<code>get_replication_partners()</code>	(Soft- <i>Layer.managers.block.BlockStorageManager</i> <i>method</i>), 14	<code>get_ticket()</code>	(Soft- <i>Layer.managers.ticket.TicketManager</i> <i>method</i>), 61
<code>get_replication_partners()</code>	(Soft- <i>Layer.managers.file.FileStorageManager</i> <i>method</i>), 28	<code>get_tracking_id()</code>	(Soft- <i>Layer.managers.hardware.HardwareManager</i> <i>method</i>), 35
<code>get_request()</code>	(SoftLayer.BasicAuthentication <i>method</i>), 77	<code>get_tracking_id()</code>	(Soft- <i>Layer.managers.vs.VSManager</i> <i>method</i>), 67
<code>get_router_options()</code>	(Soft- <i>Layer.managers.dedicated_host.DedicatedHostManager</i> <i>method</i>), 20	<code>get_translation()</code>	(Soft- <i>Layer.managers.ipsec.IPSECMANAGER</i> <i>method</i>), 41
<code>get_routers()</code>	(Soft- <i>Layer.managers.vs_placement.PlacementManager</i> <i>method</i>), 73	<code>get_translations()</code>	(Soft- <i>Layer.managers.ipsec.IPSECMANAGER</i> <i>method</i>), 42
<code>get_rule_id_from_name()</code>	(Soft- <i>Layer.managers.vs_placement.PlacementManager</i> <i>method</i>), 73	<code>get_tunnel_context()</code>	(Soft- <i>Layer.managers.ipsec.IPSECMANAGER</i> <i>method</i>), 42
<code>get_rwhois()</code>	(Soft- <i>Layer.managers.network.NetworkManager</i> <i>method</i>), 50		

`get_tunnel_contexts()` (SoftLayer.managers.ipsec.IPSECManager method), 42 option, 169

`get_upcoming_events()` (SoftLayer.managers.account.AccountManager method), 9

`get_usage_metrics()` (SoftLayer.managers.cdn.CDNManager method), 18

`get_virtual_guests()` (SoftLayer.managers.autoscale.AutoScaleManager method), 10

`get_vlan()` (SoftLayer.managers.network.NetworkManager method), 50

`get_volume_access_list()` (SoftLayer.managers.block.BlockStorageManager method), 14

`get_volume_access_list()` (SoftLayer.managers.file.FileStorageManager method), 28

`get_volume_count_limits()` (SoftLayer.managers.block.BlockStorageManager method), 14

`get_volume_count_limits()` (SoftLayer.managers.file.FileStorageManager method), 28

`get_volume_details()` (SoftLayer.managers.block.BlockStorageManager method), 14

`get_volume_details()` (SoftLayer.managers.file.FileStorageManager method), 28

`get_volume_snapshot_list()` (SoftLayer.managers.block.BlockStorageManager method), 14

`get_volume_snapshot_list()` (SoftLayer.managers.file.FileStorageManager method), 29

`get_zone()` (SoftLayer.managers.dns.DNSManager method), 24

`globalip-assign` command line option IDENTIFIER, 129 TARGET, 129

`globalip-cancel` command line option IDENTIFIER, 129

`globalip-create` command line option `-test` <test>, 129 `-v6`, `-ipv6`, 129

`globalip-list` command line option `-ip-version` <ip_version>, 130

`globalip-unassign` command line option IDENTIFIER, 130

`GROUP_ID`

`securitygroup-edit` command line

H

`hardware-bandwidth` command line option `-e`, `-end_date` <end_date>, 130 `-p`, `-summary_period` <summary_period>, 130 `-q`, `-quite_summary`, 130 `-s`, `-start_date` <start_date>, 130 IDENTIFIER, 131

`hardware-billing` command line option IDENTIFIER, 133

`hardware-cancel` command line option `-comment` <comment>, 131 `-immediate`, 131 `-reason` <reason>, 131 IDENTIFIER, 131

`hardware-create` command line option `-billing` <billing>, 132 `-export` <export>, 132 `-no-public`, 132 `-port-speed` <port_speed>, 132 `-test`, 132 `-wait` <wait>, 132 `-D`, `-domain` <domain>, 132 `-H`, `-hostname` <hostname>, 132 `-d`, `-datacenter` <datacenter>, 132 `-e`, `-extra` <extra>, 132 `-i`, `-postinstall` <postinstall>, 132 `-k`, `-key` <key>, 132 `-o`, `-os` <os>, 132 `-s`, `-size` <size>, 132 `-t`, `-template` <template>, 132

`hardware-credentials` command line option IDENTIFIER, 133

`hardware-detail` command line option `-passwords`, 133 `-price`, 133 IDENTIFIER, 133

`hardware-dns-sync` command line option `-aaaa-record`, 138 `-ptr`, 138 `-ttl` <ttl>, 138 `-a`, `-a-record`, 138 IDENTIFIER, 138

`hardware-edit` command line option `-degraded`, 134 `-private-speed` <private_speed>, 134 `-public-speed` <public_speed>, 134 `-redundant`, 134 `-D`, `-domain` <domain>, 133 `-F`, `-userfile` <userfile>, 133 `-H`, `-hostname` <hostname>, 134

- g, -tag <tag>, 133
- u, -userdata <userdata>, 134
- IDENTIFIER, 134
- hardware-list command line option
 - columns <columns>, 135
 - sortby <sortby>, 135
 - tag <tag>, 135
 - D, -domain <domain>, 134
 - H, -hostname <hostname>, 134
 - c, -cpu <cpu>, 134
 - d, -datacenter <datacenter>, 134
 - l, -limit <limit>, 135
 - m, -memory <memory>, 135
 - n, -network <network>, 135
- hardware-power-cycle command line option
 - IDENTIFIER, 135
- hardware-power-off command line option
 - IDENTIFIER, 135
- hardware-power-on command line option
 - IDENTIFIER, 136
- hardware-ready command line option
 - wait <wait>, 138
 - IDENTIFIER, 138
- hardware-reboot command line option
 - hard, -soft, 136
 - IDENTIFIER, 136
- hardware-reflash-firmware command line option
 - IDENTIFIER, 137
- hardware-reload command line option
 - i, -postinstall <postinstall>, 136
 - k, -key <key>, 136
 - IDENTIFIER, 136
- hardware-rescue command line option
 - IDENTIFIER, 137
- hardware-storage command line option
 - IDENTIFIER, 139
- hardware-toggle-ipmi command line option
 - enable, -disable, 137
 - IDENTIFIER, 137
- hardware-update-firmware command line option
 - IDENTIFIER, 137
- HardwareManager (class in *SoftLayer.managers.hardware*), 33
- has_firewall() (in *module SoftLayer.managers.firewall*), 33
- |
- IDENTIFIER
 - account-cancel-item command line option, 87
 - account-event-detail command line option, 86
 - account-invoice-detail command line option, 86
 - account-item-detail command line option, 87
 - autoscale-detail command line option, 88
 - autoscale-edit command line option, 89
 - autoscale-logs command line option, 88
 - autoscale-scale command line option, 88
 - autoscale-tag command line option, 89
 - dedicatedhost-cancel command line option, 110
 - dedicatedhost-cancel-guests command line option, 110
 - dedicatedhost-detail command line option, 110
 - dedicatedhost-list-guests command line option, 111
 - firewall-cancel command line option, 128
 - firewall-detail command line option, 128
 - firewall-edit command line option, 128
 - globalip-assign command line option, 129
 - globalip-cancel command line option, 129
 - globalip-unassign command line option, 130
 - hardware-bandwidth command line option, 131
 - hardware-billing command line option, 133
 - hardware-cancel command line option, 131
 - hardware-credentials command line option, 133
 - hardware-detail command line option, 133
 - hardware-dns-sync command line option, 138
 - hardware-edit command line option, 134
 - hardware-power-cycle command line option, 135
 - hardware-power-off command line option, 135

hardware-power-on command line option, 136

hardware-ready command line option, 138

hardware-reboot command line option, 136

hardware-reflash-firmware command line option, 137

hardware-reload command line option, 136

hardware-rescue command line option, 137

hardware-storage command line option, 139

hardware-toggle-ipmi command line option, 137

hardware-update-firmware command line option, 137

image-datacenter command line option, 142

image-delete command line option, 139

image-detail command line option, 139

image-edit command line option, 140

image-export command line option, 141

loadbal-cancel command line option, 156

loadbal-detail command line option, 151

loadbal-health command line option, 151

loadbal-l7pool-add command line option, 155

loadbal-l7pool-del command line option, 155

loadbal-member-add command line option, 152

loadbal-member-del command line option, 152

loadbal-ns-detail command line option, 157

loadbal-pool-add command line option, 153

loadbal-pool-del command line option, 154

loadbal-pool-edit command line option, 154

nas-credentials command line option, 157

object-storage-credential-create command line option, 159

object-storage-credential-delete command line option, 158

object-storage-credential-limit command line option, 158

object-storage-credential-list command line option, 158

securitygroup-detail command line option, 168

sshkey-edit command line option, 174

sshkey-print command line option, 175

sshkey-remove command line option, 174

ssl-download command line option, 175

ssl-edit command line option, 176

ssl-remove command line option, 177

subnet-cancel command line option, 177

subnet-detail command line option, 178

subnet-edit command line option, 178

subnet-edit-ip command line option, 180

tags-delete command line option, 181

tags-details command line option, 181

ticket-attach command line option, 184

ticket-detach command line option, 184

ticket-detail command line option, 182

ticket-update command line option, 183

ticket-upload command line option, 183

user-delete command line option, 188

user-detail command line option, 185

user-edit-permissions command line option, 186

user-permissions command line option, 186

virtual-bandwidth command line option, 194

virtual-billing command line option, 202

virtual-cancel command line option, 194

virtual-capture command line option, 195

virtual-credentials command line option, 203

virtual-detail command line option, 202

virtual-dns-sync command line option, 197
 virtual-edit command line option, 198
 virtual-pause command line option, 199
 virtual-power-off command line option, 199
 virtual-power-on command line option, 199
 virtual-ready command line option, 200
 virtual-reboot command line option, 200
 virtual-reload command line option, 203
 virtual-rescue command line option, 200
 virtual-resume command line option, 200
 virtual-storage command line option, 202
 virtual-upgrade command line option, 201
 virtual-usage command line option, 201
 vlan-detail command line option, 189
 vlan-edit command line option, 189
 image-datacenter command line option
 -add, -remove, 141
 IDENTIFIER, 142
 LOCATIONS, 142
 image-delete command line option
 IDENTIFIER, 139
 image-detail command line option
 IDENTIFIER, 139
 image-edit command line option
 -name <name>, 139
 -note <note>, 139
 -tag <tag>, 139
 IDENTIFIER, 140
 image-export command line option
 -ibm-api-key <ibm_api_key>, 141
 IDENTIFIER, 141
 URI, 141
 image-import command line option
 -byol, 140
 -cloud-init, 140
 -ibm-api-key <ibm_api_key>, 140
 -is-encrypted, 140
 -note <note>, 140
 -os-code <os_code>, 140
 -root-key-crn <root_key_crn>, 140
 -wrapped-dek <wrapped_dek>, 140
 NAME, 141
 URI, 141
 image-list command line option
 -name <name>, 140
 -public, -private, 140
 ImageManager (*class in SoftLayer.managers.image*), 38
 import_image_from_uri() (*SoftLayer.managers.image.ImageManager method*), 39
 IP_ADDRESS
 subnet-lookup command line option, 179
 ip_lookup() (*SoftLayer.managers.network.NetworkManager method*), 51
 ipsec-configure command line option
 CONTEXT_ID, 146
 ipsec-detail command line option
 -i, -include <include>, 143
 CONTEXT_ID, 143
 ipsec-subnet-add command line option
 -n, -network-identifier, -network <network_identifier>, 147
 -s, -subnet-id <subnet_id>, 147
 -t, -subnet-type, -type <subnet_type>, 147
 CONTEXT_ID, 147
 ipsec-subnet-remove command line option
 -s, -subnet-id <subnet_id>, 148
 -t, -subnet-type, -type <subnet_type>, 148
 CONTEXT_ID, 148
 ipsec-translation-add command line option
 -n, -note <note>, 149
 -r, -remote-ip <remote_ip>, 149
 -s, -static-ip <static_ip>, 149
 CONTEXT_ID, 149
 ipsec-translation-remove command line option
 -t, -translation-id <translation_id>, 149
 CONTEXT_ID, 149
 ipsec-translation-update command line option
 -n, -note <note>, 150
 -r, -remote-ip <remote_ip>, 150
 -s, -static-ip <static_ip>, 150
 -t, -translation-id <translation_id>, 150
 CONTEXT_ID, 150
 ipsec-update command line option
 -friendly-name <friendly_name>, 145

- phase1-auth, -p1-auth
 <phase1_auth>, 145
 - phase1-crypto, -p1-crypto
 <phase1_crypto>, 145
 - phase1-dh, -p1-dh <phase1_dh>, 145
 - phase1-key-ttl, -p1-key-ttl
 <phase1_key_ttl>, 145
 - phase2-auth, -p2-auth
 <phase2_auth>, 145
 - phase2-crypto, -p2-crypto
 <phase2_crypto>, 145
 - phase2-dh, -p2-dh <phase2_dh>, 145
 - phase2-forward-secrecy,
 -p2-forward-secrecy
 <phase2_forward_secrecy>, 145
 - phase2-key-ttl, -p2-key-ttl
 <phase2_key_ttl>, 145
 - preshared-key <preshared_key>, 145
 - remote-peer <remote_peer>, 145
 - CONTEXT_ID, 145
 - IPSECManger (class in SoftLayer.managers.ipsec), 40
 - iter_call() (SoftLayer.BaseClient method), 76
- L**
- LABEL
- sshkey-add command line option, 173
 - lbaas_order_options() (SoftLayer.managers.load_balancer.LoadBalancerManager method), 45
 - list() (SoftLayer.managers.autoscale.AutoScaleManager method), 10
 - list() (SoftLayer.managers.vs_capacity.CapacityManager method), 72
 - list() (SoftLayer.managers.vs_placement.PlacementManager method), 73
 - list_block_volume_limit() (SoftLayer.managers.block.BlockStorageManager method), 14
 - list_block_volumes() (SoftLayer.managers.block.BlockStorageManager method), 15
 - list_categories() (SoftLayer.managers.ordering.OrderingManager method), 55
 - list_cdn() (SoftLayer.managers.cdn.CDNManager method), 19
 - list_certs() (SoftLayer.managers.ssl.SSLManager method), 60
 - list_file_volume_limit() (SoftLayer.managers.file.FileStorageManager method), 29
 - list_file_volumes() (SoftLayer.managers.file.FileStorageManager method), 29
 - list_global_ips() (SoftLayer.managers.network.NetworkManager method), 51
 - list_guests() (SoftLayer.managers.dedicated_host.DedicatedHostManager method), 20
 - list_hardware() (SoftLayer.managers.hardware.HardwareManager method), 35
 - list_instances() (SoftLayer.managers.dedicated_host.DedicatedHostManager method), 21
 - list_instances() (SoftLayer.managers.vs.VSManager method), 67
 - list_items() (SoftLayer.managers.ordering.OrderingManager method), 55
 - list_keys() (SoftLayer.managers.sshkey.SshKeyManager method), 58
 - list_packages() (SoftLayer.managers.ordering.OrderingManager method), 55
 - list_presets() (SoftLayer.managers.ordering.OrderingManager method), 55
 - list_private_images() (SoftLayer.managers.image.ImageManager method), 39
 - list_public_images() (SoftLayer.managers.image.ImageManager method), 40
 - list_securitygroup_rules() (SoftLayer.managers.network.NetworkManager method), 51
 - list_securitygroups() (SoftLayer.managers.network.NetworkManager method), 51
 - list_subjects() (SoftLayer.managers.ticket.TicketManager method), 61
 - list_subnets() (SoftLayer.managers.network.NetworkManager method), 51
 - list_tickets() (SoftLayer.managers.ticket.TicketManager method), 61
 - list_vlans() (SoftLayer.managers.network.NetworkManager method), 51
 - list_volume_schedules() (SoftLayer.managers.block.BlockStorageManager method), 15
 - list_volume_schedules() (SoftLayer.managers.block.BlockStorageManager method), 15

- Layer.managers.file.FileStorageManager method*), 29
- list_zones() (*SoftLayer.managers.dns.DNSManager method*), 24
- LISTENER
- loadbal-pool-del command line option, 154
 - loadbal-pool-edit command line option, 154
- loadbal-cancel command line option IDENTIFIER, 156
- loadbal-detail command line option IDENTIFIER, 151
- loadbal-health command line option
- uuid <uuid>, 151
 - i, -interval <interval>, 151
 - r, -retry <retry>, 151
 - t, -timeout <timeout>, 151
 - u, -url <url>, 151
- IDENTIFIER, 151
- loadbal-l7pool-add command line option
- healthInterval <healthinterval>, 155
 - healthPath <healthpath>, 155
 - healthRetry <healthretry>, 155
 - healthTimeout <healthtimeout>, 155
 - P, -protocol <protocol>, 154
 - S, -server <server>, 154
 - m, -method <method>, 154
 - n, -name <name>, 154
 - s, -sticky, 155
- IDENTIFIER, 155
- loadbal-l7pool-del command line option IDENTIFIER, 155
- loadbal-member-add command line option
- private, -public, 152
 - m, -member <member>, 152
 - w, -weight <weight>, 152
- IDENTIFIER, 152
- loadbal-member-del command line option
- m, -member <member>, 152
- IDENTIFIER, 152
- loadbal-ns-detail command line option IDENTIFIER, 157
- loadbal-order command line option
- public, 156
 - verify, 156
 - b, -backend <backend>, 156
 - d, -datacenter <datacenter>, 155
 - f, -frontend <frontend>, 155
 - l, -label <label>, 155
 - m, -method <method>, 156
 - n, -name <name>, 155
 - s, -subnet <subnet>, 156
- loadbal-order-options command line option
- d, -datacenter <datacenter>, 156
- loadbal-pool-add command line option
- P, -frontProtocol <frontprotocol>, 152
 - b, -backPort <backport>, 153
 - c, -connections <connections>, 153
 - f, -frontPort <frontport>, 152
 - m, -method <method>, 153
 - p, -backProtocol <backprotocol>, 152
 - s, -sticky, 153
 - x, -sslCert <sslcert>, 153
- IDENTIFIER, 153
- loadbal-pool-del command line option IDENTIFIER, 154
- LISTENER, 154
- loadbal-pool-edit command line option
- P, -frontProtocol <frontprotocol>, 153
 - b, -backPort <backport>, 153
 - c, -connections <connections>, 153
 - f, -frontPort <frontport>, 153
 - m, -method <method>, 153
 - p, -backProtocol <backprotocol>, 153
 - s, -sticky, 153
 - x, -sslCert <sslcert>, 154
- IDENTIFIER, 154
- LISTENER, 154
- LoadBalancerManager (*class in SoftLayer.managers.load_balancer*), 44
- LOCATION
- order-place command line option, 163
 - order-place-quote command line option, 166
- LOCATIONS
- image-datacenter command line option, 142
- LUN_ID
- block-volume-set-lun-id command line option, 101
- ## M
- metadata command line option
- PROP, 107
- METADATA_ATTRIBUTES (*SoftLayer.managers.metadata attribute*), 47
- MetadataManager (*class in SoftLayer.managers.metadata*), 46
- METHOD
- call-api command line option, 107

N

NAME

image-import command line option,
141

nas-credentials command line option
IDENTIFIER, 157

NETWORK

subnet-create command line option,
178

NetworkManager (class in *Soft-
Layer.managers.network*), 47

O

object-storage-credential-create
command line option
IDENTIFIER, 159

object-storage-credential-delete
command line option
-c, -credential_id <credential_id>,
158
IDENTIFIER, 158

object-storage-credential-limit
command line option
IDENTIFIER, 158

object-storage-credential-list command
line option
IDENTIFIER, 158

order-category-list command line
option
-required, 160
PACKAGE_KEYNAME, 160

order-item-list command line option
-category <category>, 161
-keyword <keyword>, 161
PACKAGE_KEYNAME, 161

order-package-list command line option
-keyword <keyword>, 159
-package_type <package_type>, 159

order-package-locations command line
option
PACKAGE_KEYNAME, 160

order-place command line option
-billing <billing>, 163
-complex-type <complex_type>, 163
-extras <extras>, 163
-preset <preset>, 163
-quantity <quantity>, 163
-verify, 163
LOCATION, 163
ORDER_ITEMS, 163
PACKAGE_KEYNAME, 163

order-place-quote command line option
-complex-type <complex_type>, 166
-extras <extras>, 166

-name <name>, 166
-preset <preset>, 166
-send-email, 166

LOCATION, 166
ORDER_ITEMS, 166
PACKAGE_KEYNAME, 166

order-preset-list command line option
-keyword <keyword>, 162
PACKAGE_KEYNAME, 162

order-quote command line option
-complex-type <complex_type>, 164
-fqdn <fqdn>, 164
-image <image>, 164
-quantity <quantity>, 164
-verify, 164
-F, -userfile <userfile>, 164
-i, -postinstall <postinstall>, 164
-k, -key <key>, 164
-u, -userdata <userdata>, 164
QUOTE, 165

order-quote-detail command line option
QUOTE, 165

order_block_volume() (Soft-
Layer.managers.block.BlockStorageManager
method), 15

order_duplicate_volume() (Soft-
Layer.managers.block.BlockStorageManager
method), 15

order_duplicate_volume() (Soft-
Layer.managers.file.FileStorageManager
method), 29

order_file_volume() (Soft-
Layer.managers.file.FileStorageManager
method), 30

order_guest() (*SoftLayer.managers.vs.VSManager*
method), 68

ORDER_ITEMS
order-place command line option, 163
order-place-quote command line
option, 166

order_lbaas() (Soft-
Layer.managers.load_balancer.LoadBalancerManager
method), 45

order_modified_volume() (Soft-
Layer.managers.block.BlockStorageManager
method), 16

order_modified_volume() (Soft-
Layer.managers.file.FileStorageManager
method), 30

order_quote() (Soft-
Layer.managers.ordering.OrderingManager
method), 55

order_replicant_volume() (Soft-
Layer.managers.block.BlockStorageManager

- method*), 16
- order_replicant_volume() (SoftLayer.managers.file.FileStorageManager *method*), 30
- order_snapshot_space() (SoftLayer.managers.block.BlockStorageManager *method*), 16
- order_snapshot_space() (SoftLayer.managers.file.FileStorageManager *method*), 30
- OrderingManager (class in SoftLayer.managers.ordering), 53
- ORIGIN
- cdn-origin-add command line option, 104
- ORIGIN_PATH
- cdn-origin-remove command line option, 105
- ORIGIN_VOLUME_ID
- block-volume-duplicate command line option, 98
 - file-volume-duplicate command line option, 124
- P**
- PACKAGE_KEYNAME
- order-category-list command line option, 160
 - order-item-list command line option, 161
 - order-package-locations command line option, 160
 - order-place command line option, 163
 - order-place-quote command line option, 166
 - order-preset-list command line option, 162
- package_locations() (SoftLayer.managers.ordering.OrderingManager *method*), 56
- PARAMETERS
- call-api command line option, 107
- PATH
- cdn-origin-add command line option, 104
 - cdn-purge command line option, 105
- place_order() (SoftLayer.managers.dedicated_host.DedicatedHostManager *method*), 21
- place_order() (SoftLayer.managers.hardware.HardwareManager *method*), 36
- place_order() (SoftLayer.managers.ordering.OrderingManager *method*), 56
- place_quote() (SoftLayer.managers.ordering.OrderingManager *method*), 56
- PlacementManager (class in SoftLayer.managers.vs_placement), 72
- private_network() (SoftLayer.managers.metadata.MetadataManager *method*), 47
- PROP
- metadata command line option, 107
- public_network() (SoftLayer.managers.metadata.MetadataManager *method*), 47
- purge_content() (SoftLayer.managers.cdn.CDNManager *method*), 19
- Python Enhancement Proposals
- PEP 8, 211
- Q**
- QUANTITY
- subnet-create command line option, 178
- QUOTE
- order-quote command line option, 165
 - order-quote-detail command line option, 165
- R**
- RECORD
- dns-record-add command line option, 112
- RECORD_ID
- dns-record-remove command line option, 114
- RECORD_TYPE
- dns-record-add command line option, 112
- reflash_firmware() (SoftLayer.managers.hardware.HardwareManager *method*), 36
- refresh_dep_dupe() (SoftLayer.managers.block.BlockStorageManager *method*), 16
- refresh_dep_dupe() (SoftLayer.managers.file.FileStorageManager *method*), 31
- reload() (SoftLayer.managers.hardware.HardwareManager *method*), 37
- reload_instance() (SoftLayer.managers.vs.VSManager *method*), 68

`remove_certificate()` (SoftLayer.managers.ssl.SSLManager method), 60
`remove_internal_subnet()` (SoftLayer.managers.ipsec.IPSECManager method), 42
`remove_lb_listener()` (SoftLayer.managers.load_balancer.LoadBalancerManager method), 46
`remove_locations()` (SoftLayer.managers.image.ImageManager method), 40
`remove_origin()` (SoftLayer.managers.cdn.CDNManager method), 19
`remove_remote_subnet()` (SoftLayer.managers.ipsec.IPSECManager method), 42
`remove_securitygroup_rule()` (SoftLayer.managers.network.NetworkManager method), 52
`remove_securitygroup_rules()` (SoftLayer.managers.network.NetworkManager method), 52
`remove_service_subnet()` (SoftLayer.managers.ipsec.IPSECManager method), 42
`remove_subnets_from_acl()` (SoftLayer.managers.block.BlockStorageManager method), 16
`remove_translation()` (SoftLayer.managers.ipsec.IPSECManager method), 42
report-bandwidth command line option
 `-end <end>`, 167
 `-sortby <sortby>`, 167
 `-start <start>`, 167
REQUEST_ID
 securitygroup-event-log command line option, 173
`rescue()` (SoftLayer.managers.hardware.HardwareManager method), 37
`rescue()` (SoftLayer.managers.vs.VSManager method), 69
`resolve_global_ip_ids()` (SoftLayer.managers.network.NetworkManager method), 52
`resolve_ids()` (SoftLayer.managers.account.AccountManager method), 9
`resolve_ids()` (SoftLayer.managers.block.BlockStorageManager method), 17
`resolve_ids()` (SoftLayer.managers.cdn.CDNManager method), 19
`resolve_ids()` (SoftLayer.managers.dedicated_host.DedicatedHostManager method), 21
`resolve_ids()` (SoftLayer.managers.dns.DNSManager method), 24
`resolve_ids()` (SoftLayer.managers.file.FileStorageManager method), 31
`resolve_ids()` (SoftLayer.managers.firewall.FirewallManager method), 33
`resolve_ids()` (SoftLayer.managers.hardware.HardwareManager method), 37
`resolve_ids()` (SoftLayer.managers.image.ImageManager method), 40
`resolve_ids()` (SoftLayer.managers.ipsec.IPSECManager method), 43
`resolve_ids()` (SoftLayer.managers.load_balancer.LoadBalancerManager method), 46
`resolve_ids()` (SoftLayer.managers.sshkey.SshKeyManager method), 58
`resolve_ids()` (SoftLayer.managers.ticket.TicketManager method), 61
`resolve_ids()` (SoftLayer.managers.vs.VSManager method), 69
`resolve_ids()` (SoftLayer.managers.vs_capacity.CapacityManager method), 72
`resolve_ids()` (SoftLayer.managers.vs_placement.PlacementManager method), 73
`resolve_subnet_ids()` (SoftLayer.managers.network.NetworkManager method), 52
`resolve_vlan_ids()` (SoftLayer.managers.network.NetworkManager method), 52
`restore_from_snapshot()` (SoftLayer.managers.block.BlockStorageManager method), 17
`restore_from_snapshot()` (SoftLayer.managers.file.FileStorageManager method), 31
RULE_ID
 securitygroup-rule-edit command

line option, 171
 securitygroup-rule-remove command
 line option, 171
 rwhois-edit command line option
 -abuse <abuse>, 167
 -address1 <address1>, 167
 -address2 <address2>, 167
 -city <city>, 167
 -company <company>, 167
 -country <country>, 167
 -firstname <firstname>, 167
 -lastname <lastname>, 167
 -postal <postal>, 167
 -public, -private, 167
 -state <state>, 168

S

scale() (*SoftLayer.managers.autoscale.AutoScaleManager*
method), 10
 scale_to() (*SoftLayer.managers.autoscale.AutoScaleManager*
method), 10
 securitygroup-create command line
 option
 -d, -description <description>, 169
 -n, -name <name>, 169
 securitygroup-delete command line
 option
 SECURITYGROUP_ID, 169
 securitygroup-detail command line
 option
 IDENTIFIER, 168
 securitygroup-edit command line option
 -d, -description <description>, 169
 -n, -name <name>, 169
 GROUP_ID, 169
 securitygroup-event-log command line
 option
 REQUEST_ID, 173
 securitygroup-interface-add command
 line option
 -i, -interface <interface>, 172
 -n, -network-component
 <network_component>, 172
 -s, -server <server>, 172
 SECURITYGROUP_ID, 172
 securitygroup-interface-list command
 line option
 -sortby <sortby>, 172
 SECURITYGROUP_ID, 172
 securitygroup-interface-remove command
 line option
 -i, -interface <interface>, 173
 -n, -network-component
 <network_component>, 173

-s, -server <server>, 173
 SECURITYGROUP_ID, 173
 securitygroup-list command line option
 -sortby <sortby>, 168
 -l, -limit <limit>, 168
 securitygroup-rule-add command line
 option
 -M, -port-max <port_max>, 170
 -d, -direction <direction>, 170
 -e, -ethertype <ethertype>, 170
 -m, -port-min <port_min>, 170
 -p, -protocol <protocol>, 170
 -r, -remote-ip <remote_ip>, 170
 -s, -remote-group <remote_group>,
 170
 SECURITYGROUP_ID, 171
 securitygroup-rule-edit command line
 option
 -M, -port-max <port_max>, 171
 -d, -direction <direction>, 171
 -e, -ethertype <ethertype>, 171
 -m, -port-min <port_min>, 171
 -p, -protocol <protocol>, 171
 -r, -remote-ip <remote_ip>, 171
 -s, -remote-group <remote_group>,
 171
 RULE_ID, 171
 SECURITYGROUP_ID, 171
 securitygroup-rule-list command line
 option
 -sortby <sortby>, 169
 SECURITYGROUP_ID, 170
 securitygroup-rule-remove command line
 option
 RULE_ID, 171
 SECURITYGROUP_ID, 171
 SECURITYGROUP_ID
 securitygroup-delete command line
 option, 169
 securitygroup-interface-add
 command line option, 172
 securitygroup-interface-list
 command line option, 172
 securitygroup-interface-remove
 command line option, 173
 securitygroup-rule-add command
 line option, 171
 securitygroup-rule-edit command
 line option, 171
 securitygroup-rule-list command
 line option, 170
 securitygroup-rule-remove command
 line option, 171
 SERVICE

- call-api command line option, 107
- set_credential_password() (SoftLayer.managers.block.BlockStorageManager method), 17
- set_subnet_ipaddress_note() (SoftLayer.managers.network.NetworkManager method), 52
- set_tags() (SoftLayer.managers.network.NetworkManager method), 52
- set_tags() (SoftLayer.managers.vs.VSManager method), 69
- set_tags_subnet() (SoftLayer.managers.network.NetworkManager method), 52
- SNAPSHOT_ID
 - block-snapshot-delete command line option, 94
 - block-volume-refresh command line option, 103
 - file-snapshot-delete command line option, 120
 - file-volume-refresh command line option, 127
- SoftLayer (module), 75
- SoftLayer.managers.account (module), 8
- SoftLayer.managers.autoscale (module), 9
- SoftLayer.managers.block (module), 11
- SoftLayer.managers.cdn (module), 17
- SoftLayer.managers.dedicated_host (module), 19
- SoftLayer.managers.dns (module), 22
- SoftLayer.managers.event_log (module), 24
- SoftLayer.managers.file (module), 25
- SoftLayer.managers.firewall (module), 31
- SoftLayer.managers.hardware (module), 33
- SoftLayer.managers.image (module), 38
- SoftLayer.managers.ipsec (module), 40
- SoftLayer.managers.load_balancer (module), 43
- SoftLayer.managers.metadata (module), 46
- SoftLayer.managers.network (module), 47
- SoftLayer.managers.ordering (module), 52
- SoftLayer.managers.sshkey (module), 58
- SoftLayer.managers.ssl (module), 59
- SoftLayer.managers.ticket (module), 60
- SoftLayer.managers.vs (module), 62
- SoftLayer.managers.vs_capacity (module), 71
- SoftLayer.managers.vs_placement (module), 72
- SoftLayerAPIError, 77
- SoftLayerError, 77
- SoftLayerListResult (class in SoftLayer), 78
- sshkey-add command line option
 - note <note>, 173
 - f, -in-file <in_file>, 173
 - k, -key <key>, 173
 - LABEL, 173
- sshkey-edit command line option
 - note <note>, 174
 - k, -label <label>, 174
 - IDENTIFIER, 174
- sshkey-list command line option
 - sortby <sortby>, 174
- sshkey-print command line option
 - f, -out-file <out_file>, 175
 - IDENTIFIER, 175
- sshkey-remove command line option
 - IDENTIFIER, 174
- SshKeyManager (class in SoftLayer.managers.sshkey), 58
- ssl-add command line option
 - crt <crt>, 175
 - csr <csr>, 175
 - icc <icc>, 175
 - key <key>, 175
 - notes <notes>, 175
- ssl-download command line option
 - IDENTIFIER, 175
- ssl-edit command line option
 - crt <crt>, 176
 - csr <csr>, 176
 - icc <icc>, 176
 - key <key>, 176
 - notes <notes>, 176
 - IDENTIFIER, 176
- ssl-list command line option
 - sortby <sortby>, 176
 - status <status>, 176
- ssl-remove command line option
 - IDENTIFIER, 177
- SSLManager (class in SoftLayer.managers.ssl), 59
- start_data (SoftLayer.managers.cdn.CDNManager attribute), 19
- SUBNET
 - user-vpn-subnet command line option, 188
- subnet-cancel command line option
 - IDENTIFIER, 177
- subnet-create command line option
 - ipv6, -v6, 177
 - test, 177
 - ENDPOINT_ID, 178
 - NETWORK, 178
 - QUANTITY, 178
- subnet-detail command line option
 - no-hardware, 178
 - no-vs, 178

IDENTIFIER, 178

subnet-edit command line option

- n, -note <note>, 178
- t, -tags <tags>, 178

IDENTIFIER, 178

subnet-edit-ip command line option

- note <note>, 179

IDENTIFIER, 180

subnet-list command line option

- identifier <identifier>, 179
- ipv4, -v4, 179
- ipv6, -v6, 179
- network-space <network_space>, 179
- sortby <sortby>, 179
- d, -datacenter <datacenter>, 179
- t, -subnet-type <subnet_type>, 179

subnet-lookup command line option

IP_ADDRESS, 179

summary command line option

- sortby <sortby>, 166

summary_by_datacenter() (SoftLayer.managers.network.NetworkManager method), 52

sync_host_record() (SoftLayer.managers.dns.DNSManager method), 24

sync_ptr_record() (SoftLayer.managers.dns.DNSManager method), 24

T

tags-cleanup command line option

- d, -dry-run, 181

tags-delete command line option

- name, 181

IDENTIFIER, 181

tags-details command line option

- name, 181

IDENTIFIER, 181

tags-list command line option

- d, -detail, 180

tags-set command line option

- k, -key-name <key_name>, 180
- r, -resource-id <resource_id>, 180
- t, -tags <tags>, 180

TARGET

firewall-add command line option, 128

globalip-assign command line option, 129

ticket-attach command line option

- hardware <hardware_identifier>, 184
- virtual <virtual_identifier>, 184

IDENTIFIER, 184

ticket-create command line option

- body <body>, 182
- hardware <hardware_identifier>, 182
- priority <priority>, 182
- subject-id <subject_id>, 182
- title <title>, 182
- virtual <virtual_identifier>, 182

ticket-detach command line option

- hardware <hardware_identifier>, 184
- virtual <virtual_identifier>, 184

IDENTIFIER, 184

ticket-detail command line option

- count <count>, 182

IDENTIFIER, 182

ticket-list command line option

- open, -closed, 183

ticket-update command line option

- body <body>, 183

IDENTIFIER, 183

ticket-upload command line option

- name <name>, 183
- path <path>, 183

IDENTIFIER, 183

TicketManager (class in SoftLayer.managers.ticket), 60

total_count (SoftLayer.SoftLayerListResult attribute), 78

U

unassign_global_ip() (SoftLayer.managers.network.NetworkManager method), 52

UNIQUE_ID

cdn-detail command line option, 103

cdn-origin-add command line option, 104

cdn-origin-list command line option, 105

cdn-origin-remove command line option, 105

cdn-purge command line option, 105

update_firmware() (SoftLayer.managers.hardware.HardwareManager method), 37

update_lb_health_monitors() (SoftLayer.managers.load_balancer.LoadBalancerManager method), 46

update_ticket() (SoftLayer.managers.ticket.TicketManager method), 61

update_translation() (SoftLayer.managers.ipsec.IPSECManger method), 43

- update_tunnel_context() (SoftLayer.managers.ipsec.IPSECManger method), 43
- upgrade() (SoftLayer.managers.vs.VSManager method), 69
- upload_attachment() (SoftLayer.managers.ticket.TicketManager method), 62
- URI
 - image-export command line option, 141
 - image-import command line option, 141
- USER
 - user-edit-details command line option, 187
 - user-vpn-manual command line option, 188
 - user-vpn-subnet command line option, 188
- user-create command line option
 - a, -api-key, 187
 - e, -email <email>, 187
 - p, -password <password>, 187
 - t, -template <template>, 187
 - u, -from-user <from_user>, 187
 USERNAME, 187
- user-delete command line option
 - IDENTIFIER, 188
- user-detail command line option
 - e, -events, 185
 - h, -hardware, 185
 - k, -keys, 185
 - l, -logins, 185
 - p, -permissions, 185
 - v, -virtual, 185
 IDENTIFIER, 185
- user-edit-details command line option
 - t, -template <template>, 186
 USER, 187
- user-edit-permissions command line option
 - enable, -disable, 186
 - p, -permission <permission>, 186
 - u, -from-user <from_user>, 186
 IDENTIFIER, 186
- user-list command line option
 - columns <columns>, 185
- user-permissions command line option
 - IDENTIFIER, 186
- user-vpn-manual command line option
 - enable, -disable, 188
 USER, 188
- user-vpn-subnet command line option
 - add, -remove, 188
 SUBNET, 188
 - USER, 188
 USERNAME
 - user-create command line option, 187
- ## V
- verify_create_instance() (SoftLayer.managers.vs.VSManager method), 70
- verify_order() (SoftLayer.managers.dedicated_host.DedicatedHostManager method), 22
- verify_order() (SoftLayer.managers.hardware.HardwareManager method), 37
- verify_order() (SoftLayer.managers.ordering.OrderingManager method), 57
- verify_quote() (SoftLayer.managers.ordering.OrderingManager method), 57
- virtual-bandwidth command line option
 - e, -end_date <end_date>, 194
 - p, -summary_period <summary_period>, 194
 - q, -quite_summary, 194
 - s, -start_date <start_date>, 194
 IDENTIFIER, 194
- virtual-billing command line option
 - IDENTIFIER, 202
- virtual-cancel command line option
 - IDENTIFIER, 194
- virtual-capture command line option
 - all <all>, 194
 - note <note>, 194
 - n, -name <name>, 194
 IDENTIFIER, 195
- virtual-create command line option
 - billing <billing>, 195
 - boot-mode <boot_mode>, 195
 - dedicated, -public, 195
 - disk <disk>, 196
 - export <export>, 195
 - host-id <host_id>, 195
 - image <image>, 195
 - ipv6, 196
 - like <like>, 196
 - placementgroup <placementgroup>, 196
 - private, 196
 - san, 195
 - subnet-private <subnet_private>, 196

```

-subnet-public <subnet_public>, 196
-test, 195
-transient, 196
-vlan-private <vlan_private>, 196
-vlan-public <vlan_public>, 196
-wait <wait>, 196
-D, -domain <domain>, 195
-F, -userfile <userfile>, 196
-H, -hostname <hostname>, 195
-S, -public-security-group
    <public_security_group>, 196
-c, -cpu <cpu>, 195
-d, -datacenter <datacenter>, 195
-f, -flavor <flavor>, 195
-g, -tag <tag>, 196
-i, -postinstall <postinstall>, 195
-k, -key <key>, 196
-m, -memory <memory>, 195
-n, -network <network>, 196
-o, -os <os>, 195
-s, -private-security-group
    <private_security_group>, 196
-t, -template <template>, 196
-u, -userdata <userdata>, 196
virtual-credentials command line
option
    IDENTIFIER, 203
virtual-detail command line option
    -passwords, 202
    -price, 202
    IDENTIFIER, 202
virtual-dns-sync command line option
    -aaaa-record, 197
    -ptr, 197
    -ttl <ttl>, 197
    -a, -a-record, 197
    IDENTIFIER, 197
virtual-edit command line option
    -private-speed <private_speed>, 198
    -public-speed <public_speed>, 197
    -D, -domain <domain>, 197
    -F, -userfile <userfile>, 197
    -H, -hostname <hostname>, 197
    -g, -tag <tag>, 197
    -u, -userdata <userdata>, 197
    IDENTIFIER, 198
virtual-list command line option
    -columns <columns>, 198
    -hourly, 198
    -monthly, 198
    -sortby <sortby>, 198
    -tag <tag>, 198
    -transient <transient>, 198
    -D, -domain <domain>, 198
    -H, -hostname <hostname>, 198
    -c, -cpu <cpu>, 198
    -d, -datacenter <datacenter>, 198
    -l, -limit <limit>, 198
    -m, -memory <memory>, 198
    -n, -network <network>, 198
virtual-pause command line option
    IDENTIFIER, 199
virtual-power-off command line option
    -hard, -soft, 199
    IDENTIFIER, 199
virtual-power-on command line option
    IDENTIFIER, 199
virtual-ready command line option
    -wait <wait>, 200
    IDENTIFIER, 200
virtual-reboot command line option
    -hard, -soft, 200
    IDENTIFIER, 200
virtual-reload command line option
    -image <image>, 203
    -i, -postinstall <postinstall>, 203
    -k, -key <key>, 203
    IDENTIFIER, 203
virtual-rescue command line option
    IDENTIFIER, 200
virtual-resume command line option
    IDENTIFIER, 200
virtual-storage command line option
    IDENTIFIER, 202
virtual-upgrade command line option
    -cpu <cpu>, 201
    -flavor <flavor>, 201
    -memory <memory>, 201
    -network <network>, 201
    -private, 201
    IDENTIFIER, 201
virtual-usage command line option
    -e, -end_date <end_date>, 201
    -p, -summary_period
        <summary_period>, 201
    -s, -start_date <start_date>, 201
    -t, -valid_type <valid_type>, 201
    IDENTIFIER, 201
vlan-detail command line option
    -no-hardware, 189
    -no-vs, 189
    IDENTIFIER, 189
vlan-edit command line option
    -e, -note <note>, 189
    -g, -tags <tags>, 189
    -n, -name <name>, 189
    IDENTIFIER, 189
vlan-list command line option

```

- name <name>, 190
- sortby <sortby>, 189
- d, -datacenter <datacenter>, 189
- l, -limit <limit>, 190
- n, -number <number>, 189
- VOLUME_ID
 - block-access-authorize command line option, 90
 - block-access-list command line option, 90
 - block-access-revoke command line option, 91
 - block-replica-failback command line option, 92
 - block-replica-failover command line option, 92
 - block-replica-locations command line option, 93
 - block-replica-order command line option, 92
 - block-replica-partners command line option, 93
 - block-snapshot-cancel command line option, 94
 - block-snapshot-create command line option, 94
 - block-snapshot-disable command line option, 95
 - block-snapshot-enable command line option, 95
 - block-snapshot-list command line option, 96
 - block-snapshot-order command line option, 96
 - block-snapshot-restore command line option, 97
 - block-snapshot-schedule-list command line option, 95
 - block-volume-cancel command line option, 97
 - block-volume-convert command line option, 103
 - block-volume-detail command line option, 98
 - block-volume-modify command line option, 100
 - block-volume-refresh command line option, 103
 - block-volume-set-lun-id command line option, 101
 - file-access-authorize command line option, 116
 - file-access-list command line option, 117
 - file-access-revoke command line option, 117
 - file-replica-failback command line option, 118
 - file-replica-failover command line option, 118
 - file-replica-locations command line option, 119
 - file-replica-order command line option, 118
 - file-replica-partners command line option, 119
 - file-snapshot-cancel command line option, 120
 - file-snapshot-create command line option, 120
 - file-snapshot-disable command line option, 121
 - file-snapshot-enable command line option, 121
 - file-snapshot-list command line option, 122
 - file-snapshot-order command line option, 122
 - file-snapshot-restore command line option, 122
 - file-snapshot-schedule-list command line option, 127
 - file-volume-cancel command line option, 123
 - file-volume-convert command line option, 127
 - file-volume-detail command line option, 123
 - file-volume-modify command line option, 125
 - file-volume-refresh command line option, 127
- VSMANAGER (*class in SoftLayer.managers.vs*), 62
- W**
 - wait_for_ready() (*SoftLayer.managers.hardware.HardwareManager method*), 37
 - wait_for_ready() (*SoftLayer.managers.vs.VSMANAGER method*), 70
 - wait_for_transaction() (*SoftLayer.managers.vs.VSMANAGER method*), 70
- Z**
 - ZONE

- dns-record-list command line option, [113](#)
- dns-zone-create command line option, [114](#)
- dns-zone-delete command line option, [114](#)
- dns-zone-print command line option, [115](#)

ZONE_ID

- dns-record-edit command line option, [113](#)

ZONEFILE

- dns-import command line option, [111](#)