Pure Storage REST Client Documentation

Release 1.17.0

Pure Storage

Aug 23, 2019

Contents

1 Installation Guide **1**
- 1.1 Python Package Index Installation . . . 1
- 1.2 Source Code Installation . . . 1

2 Quick Start Guide **3**
- 2.1 Before You Begin . . . 3
- 2.2 Starting A Session . . . 3
- 2.3 Creating Volumes . . . 4
- 2.4 Creating Hosts and Hgroups . . . 4
- 2.5 Connecting Volumes . . . 5
- 2.6 Using Snapshots . . . 6
- 2.7 Disconnecting and Destroying Volumes . . . 7
- 2.8 Enable Secure HTTPS Requests . . . 8

3 API Glossary **9**

4 Indices and tables **55**

5 Change Log **57**

6 Overview **59**

7 Community **61**

8 Changes **63**

9 License **65**

10 Indices and tables **67**

Python Module Index **69**

Index **71**

CHAPTER 1

Installation Guide

The Pure Storage FlashArray REST Client is available through the Python Package Index.

The code is available on github and can optionally be built and installed from source.

1.1 Python Package Index Installation

```
$ pip install purestorage
```

Or

```
$ easy_install purestorage
```

1.2 Source Code Installation

```
$ mkdir purestorage
$ cd purestorage
$ git clone https://github.com/purestorage/rest-client.git
$ cd rest-client
$ python setup.py install
```

Or to build HTML documentation from source:

```
$ cd docs/
$ make html
```

This creates a _build/ directory under docs/.

CHAPTER 2

Quick Start Guide

This guide is intended to give users a basic idea of REST Client usage through examples.

2.1 Before You Begin

You should already have the Pure Storage FlashArray REST Client installed.

This includes installing REST Client package dependencies.

See *Installation Guide* for more information.

2.2 Starting A Session

To verify the REST Client package is installed and importable, try executing the following in a Python interpreter:

```
>>> import purestorage
```

If that succeeds without an ImportError, you are ready to start a REST session using the client.

REST sessions are automatically established when a FlashArray object is instantiated. To instantiate a FlashArray object, provide the IP address or domain name of the target array as well as a username and password, or API token.

```
>>> array = purestorage.FlashArray("localhost", "pureuser", "pureuser")
```

```
>>> array = purestorage.FlashArray("localhost",
        api_token="6e1b80a1-cd63-de90-b74a-7fc16d034016")
```

```
from purestorage import FlashArray

array = FlashArray("localhost", "pureuser", "pureuser")
array_info = array.get()
```

(continues on next page)

(continued from previous page)

```
print "FlashArray {} (version {}) REST session established!".format(
        array_info['array_name'], array_info['version'])
```

To end a session, invalidate your REST cookie:

```
array.invalidate_cookie()
```

Calling any other methods again creates a new cookie automatically.

2.3 Creating Volumes

When creating a volume, specify the volume name and a size.

Size can either be an integer or a string with an optional suffix.

```
>>> array.create_volume("vol1", 1024 ** 3)
{
 u'source': None,
 u'serial': u'DABA29111570F7A4000114C0',
 u'size': 1073741824,
 u'name': u'vol1',
 u'created': u'2014-08-11T17:19:35Z'
}
>>> array.create_volume("vol2", "5M")
{
 u'source': None,
 u'serial': u'DABA29111570F7A4000114C1',
 u'size': 524288000,
 u'name': u'vol2',
 u'created': u'2014-08-11T17:19:51Z'
}
```

2.4 Creating Hosts and Hgroups

Host creation requires a name only.

Optionally IQNs or WWNs can be specified during creation, or they can be set for a particular host after creating.

Similarly, hgroup creation requires a name only and hosts can be added to the hgroup as part of creation or in a subsequent set call.

```
>>> array.create_host("host1", iqnlist=["iqn.2001-04.com.example:diskarrays-sn-
↪a8675308",
                                        "iqn.2001-04.com.example:diskarrays-sn-
↪a8675309"])
{
 u'iqn': [u'iqn.2001-04.com.example:diskarrays-sn-a8675308', u'iqn.2001-04.com.
↪example:diskarrays-sn-a8675309'],
 u'wwn': [],
 u'name': u'host1'
}
>>> array.create_host("host2")
{
```

(continues on next page)

(continued from previous page)

```
 u'iqn': [],
 u'wwn': [],
 u'name':
 u'host2'
}
>>> array.set_host("host2", wwnlist=["1234567812345678"])
{
 u'iqn': [],
 u'wwn': [u'1234567812345678'],
 u'name': u'host2',
 u'hgroup': None
}
>>> array.create_hgroup("hgroup1", hostlist=["host1", "host2"])
{
 u'hosts': [u'host1', u'host2'],
 u'name': u'hgroup1'
}
```

2.5 Connecting Volumes

When connecting volumes to hosts and hgroups, just specify the volume name and the name of the host or hgroup. LUNs may also be specified as optional keyword arguments.

```
>>> array.connect_host("host1", "vol1")
{
 u'vol': u'vol1',
 u'name': u'host1',
 u'lun': 1
}
>>> array.connect_hgroup("hgroup1", "vol2")
{
 u'vol': u'vol2',
 u'name': u'hgroup1',
 u'lun': 10
}
>>> array.list_host_connections("host1")
[
 {
  u'vol': u'vol1',
  u'name': u'host1',
  u'lun': 1,
  u'hgroup': None
 },
 {
  u'vol': u'vol2',
  u'name': u'host1',
  u'lun': 10,
  u'hgroup': u'hgroup1'
 }
]
>>> array.list_hgroup_connections("hgroup1")
[{
  u'vol': u'vol2',
  u'name': u'hgroup1',
```

(continues on next page)

(continued from previous page)

```
  u'lun': 10
}]
```

2.6 Using Snapshots

Snapshots can be taken of individual volumes or collections of volumes. Snapshots of more than one volume are guaranteed to be point in time consistent.

Snapshots (or volumes) can be copied out to new volumes.

```
>>> array.create_snapshot("vol2")
{
 u'source': u'vol2',
 u'serial': u'DABA29111570F7A4000115A3',
 u'size': 5242880,
 u'name': u'vol2.5539',
 u'created': u'2014-08-15T17:21:22Z'
}
>>> array.create_snapshots(["vol1", "vol2"], suffix="together")
[
 {
  u'source': u'vol1',
  u'serial': u'DABA29111570F7A4000115A4',
  u'size': 1073741824,
  u'name': u'vol1.together',
  u'created': u'2014-08-15T17:21:58Z'
 },
 {
  u'source': u'vol2',
  u'serial': u'DABA29111570F7A4000115A5',
  u'size': 5242880,
  u'name': u'vol2.together',
  u'created': u'2014-08-15T17:21:58Z'
 }
]
>>> array.copy_volume("vol1.together", "vol3")
{
 u'source': u'vol1',
 u'serial': u'DABA29111570F7A4000115A6',
 u'size': 1073741824,
 u'name': u'vol3',
 u'created': u'2014-08-15T17:21:58Z'
}
>>> array.list_volumes(snap=True)
[
 {
  u'source': u'vol1',
  u'serial': u'DABA29111570F7A4000115A4',
  u'size': 1073741824,
  u'name': u'vol1.together',
  u'created': u'2014-08-15T17:21:58Z'
 },
 {
  u'source': u'vol2',
```

(continues on next page)

(continued from previous page)

```
  u'serial': u'DABA29111570F7A4000115A5',
  u'size': 5242880,
  u'name': u'vol2.together',
  u'created': u'2014-08-15T17:21:58Z'
 },
 {
  u'source': u'vol2',
  u'serial': u'DABA29111570F7A4000115A3',
  u'size': 5242880,
  u'name': u'vol2.5539',
  u'created': u'2014-08-15T17:21:22Z'
 }
]
```

2.7 Disconnecting and Destroying Volumes

Volumes must be disconnected before they can be destroyed, just as hosts must be disconnected before they can be deleted.

A destroyed volume may be recovered (for up to 24 hours following destruction) or explicitly eradicated.

```
>>> array.disconnect_host("host1", "vol1")
{
 u'vol': u'vol1',
 u'name': u'host1'
}
>>> array.destroy_volume("vol1")
{
 u'name': u'vol1'
}
>>> array.list_volumes(pending_only=True)
[
 {
  u'name': u'vol1',
  u'created': u'2014-08-15T17:13:08Z',
  u'source': None,
  u'time_remaining': 86400,
  u'serial': u'DABA29111570F7A4000115A1',
  u'size': 1073741824
 }
]
>>> array.recover_volume("vol1")
{
 u'name': u'vol1'
}
>>> array.rename_volume("vol1", "renamed")
{
 u'name': u'renamed'
}
>>> array.destroy_volume("renamed")
{
 u'name': u'renamed'
}
>>> array.eradicate_volume("renamed")
```

(continues on next page)

(continued from previous page)

```
{
 u'name': u'renamed'
}
>>> array.list_volumes(pending_only=True)
[]
```

2.8 Enable Secure HTTPS Requests

By default the requests being made will not verify the SSL certificate of the target array. Requests made this way will log a InsecureRequestWarning.

To enable verification use the verify_https flag:

```
>>> array = purestorage.FlashArray("localhost", "pureuser", "pureuser", verify_
↪https=True)
```

This does require that the target array has a trusted certificate and will be validated correctly by the system making the request.

If using an 'untrusted' certificate (e.g. self-signed certificate) you can optionally pass in a path to the certificate file:

```
>>> array = purestorage.FlashArray("localhost", "pureuser", "pureuser", verify_
↪https=True,
                                   ssl_cert="/etc/ssl/certs/pure-self-signed.crt")
```

CHAPTER 3

API Glossary

Contents: This library provides an easy way to script administration tasks for the Pure Storage FlashArray.

When passing arguments to methods that take **kwargs, the exact parameters that can be passed can be found in the REST API guide for the given release of Purity running on the FlashArray.

class purestorage.**FlashArray**(*target*, *username=None*, *password=None*, *api_token=None*, *rest_version=None*, *verify_https=False*, *ssl_cert=None*, *user_agent=None*, *request_kwargs=None*)

Represents a Pure Storage FlashArray and exposes administrative APIs.

Parameters

- **target** (*str*) – IP address or domain name of the target array's management interface.
- **username** (*str, optional*) – Username of the user with which to log in.
- **password** (*str, optional*) – Password of the user with which to log in.
- **api_token** (*str, optional*) – API token of the user with which to log in.
- **rest_version** (*str, optional*) – REST API version to use when communicating with target array.
- **verify_https** (*bool, optional*) – Enable SSL certificate verification for HTTPS requests.
- **ssl_cert** (*str, optional*) – Path to SSL certificate or CA Bundle file. Ignored if verify_https=False.
- **user_agent** (*str, optional*) – String to be used as the HTTP User-Agent for requests.
- **request_kwargs** (*dict, optional*) – Keyword arguments that we will pass into the the call to requests.request.

Raises *PureError*

- If the target array cannot be found.
- If the target array does not support any of the REST versions used by this library.

- If the username and password or api_token are invalid.

Raises `ValueError`

- If no api_token or username and password are specified.
- If an api_token and a username or password are specified.
- If the specified rest_version is not supported by this library or by the target array.

Note: The FlashArray constructor requires either a username and password or an api_token but not both.

Note: If a rest_version is not specified, the FlashArray object uses the highest REST API version supported by both the target array and this library. If the REST API version should become deprecated during the lifetime of the FlashArray object, the object renegotiates a REST version to use and continues running.

Note: If a rest_version is specified, that version is used so long as it is supported by both the target array and this library. In this case, the FlashArray object does not attempt to renegotiate the REST API version.

Note: Valid entries in request_kwargs may vary by your version of requests.

If you wish to use secure connections, we suggest you use an entry in request_kwargs rather than the verify_https and ssl_cert arguments. (e.g. request_kwargs={"verify": "path/to/ca_bundle"}) You should consider these options deprecated, though we will continue to support them for backward compatibility for the foreseeable future.

---

add_hgroup(*hgroup*, *pgroup*)
Add an hgroup to a pgroup.

Parameters

- **hgroup** (*str*) – Name of the hgroup to add to pgroup.
- **pgroup** (*str*) – pgroup to which to add hgroup.

Returns A dictionary mapping "name" to hgroup and "protection_group" to pgroup.

Return type ResponseDict

add_host(*host*, *pgroup*)
Add a host to a pgroup.

Parameters

- **host** (*str*) – Name of the host to add to pgroup.
- **pgroup** (*str*) – pgroup to which to add host.

Returns A dictionary mapping "name" to host and "protection_group" to pgroup.

Return type ResponseDict

add_pod(*pod*, *array*)
Add arrays to a pod.

Parameters

- **pod** (*str*) – Name of the pod.

- **array** (*str*) – Array to add to pod.

Returns A dictionary mapping "name" to pod and "array" to the pod's new array list.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

add_volume(*volume*, *pgroup*)
Add a volume to a pgroup.

Parameters

- **volume** (*str*) – Name of the volume to add to pgroup.
- **pgroup** (*str*) – pgroup to which to add volume.

Returns A dictionary mapping "name" to volume and "protection_group" to pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.7 or later.

---

clear_message(*message_id*)
Clear an alert message or audit record flag.

Parameters **message_id** (*int or str*) – ID of the message to unflag.

Returns A dictionary mapping "id" to message_id.

Return type ResponseDict

clone_pod(*source*, *dest*, ***kwargs*)
Clone an existing pod to a new one.

Parameters

- **source** (*str*) – Name of the pod the be cloned.
- **dest** (*str*) – Name of the target pod to clone into
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST pod/:pod**

Returns A dictionary describing the created pod

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

connect_array(*address*, *connection_key*, *connection_type*, ***kwargs*)
Connect this array with another one.

Parameters

- **address** (*str*) – IP address or DNS name of other array.
- **connection_key** (*str*) – Connection key of other array.
- **connection_type** (*list*) – Type(s) of connection desired.

- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST array/connection**

Returns A dictionary describing the connection to the other array.

Return type ResponseDict

Note: Currently, the only type of connection is "replication".

Note: Requires use of REST API 1.2 or later.

---

connect_azure_offload(*name*, ***kwargs*)
Connect an offload Azure Blob target.

Parameters

- **name** (*str*) – Name of offload Azure Blob target to be connected.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST offload/azure/{}**

Returns A dictionary describing the Azure Blob target.

Return type ResponseDict

connect_hgroup(*hgroup*, *volume*, ***kwargs*)
Create a shared connection between a host group and a volume.

Parameters

- **hgroup** (*str*) – Name of hgroup to connect to volume.
- **volume** (*str*) – Name of volume to connect to hgroup.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST hgroup/:hgroup/volume/:volume**

Returns A dictionary describing the connection between the hgroup and volume.

Return type ResponseDict

connect_host(*host*, *volume*, ***kwargs*)
Create a connection between a host and a volume.

Parameters

- **host** (*str*) – Name of host to connect to volume.
- **volume** (*str*) – Name of volume to connect to host.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST host/:host/volume/:volume**

Returns A dictionary describing the connection between the host and volume.

Return type ResponseDict

connect_nfs_offload(*name*, ***kwargs*)
Connect an offload nfs target.

Parameters

- **name** (*str*) – Name of offload nfs target to be connected.

- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST nfs_offload/{}**

Returns A dictionary describing the nfs target.

Return type ResponseDict

connect_s3_offload(*name*, ***kwargs*)
Connect an offload S3 target.

Parameters

- **name** (*str*) – Name of offload S3 target to be connected.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST s3_offload/{}**

Returns A dictionary describing the S3 target.

Return type ResponseDict

copy_volume(*source*, *dest*, ***kwargs*)
Clone a volume and return a dictionary describing the new volume.

Parameters

- **source** (*str*) – Name of the source volume.
- **dest** (*str*) – Name of the destination volume.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST volume/:volume**

Returns A dictionary describing the destination volume.

Return type ResponseDict

create_admin(*admin*, ***kwargs*)
Create an admin.

Parameters

- **admin** (*str*) – Name of admin.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST admin/:admin**

Returns A dictionary describing the new admin.

Return type ResponseDict

Note: Requires use of REST API 1.14 or later.

---

create_alert_recipient(*address*)
Add an alert recipient.

Parameters **address** (*str*) – Email address of alert recipient to be created.

Returns A dictionary mapping "name" to address and "enabled" to True.

Return type ResponseDict

create_api_token(*admin*, ***kwargs*)
Create an API token for an admin.

Parameters

- **admin** (*str*) – Name of admin for whom to create an API token.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST admin/:admin/apitoken**

Returns A dictionary describing the new API token.

Return type ResponseDict

create_certificate(*name*, ***kwargs*)
Create a new managed certificate.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT cert**

Returns A dictionary describing the configured array certificate.

Return type ResponseDict

Note: Requires use of REST API 1.12 or later.

You may not create the management certificate, as it already exists.

---

create_conglomerate_volume(*volume*)
Create a conglomerate volume and return a dictionary describing it.

Parameters **volume** (*str*) – Name of the volume to be created.

Returns A dictionary describing the created conglomerate volume.

Return type ResponseDict

Note: This is not a typical volume thus there is no size. It's main purpose to connect to a host/hgroup to create a PE LUN. Once the conglomerate volume is connected to a host/hgroup, it is used as a protocol-endpoint to connect a vvol to a host/hgroup to allow traffic.

Note: Requires use of REST API 1.13 or later.

---

create_hgroup(*hgroup*, ***kwargs*)
Create a host group and return a dictionary describing it.

Parameters

- **hgroup** (*str*) – Name of hgroup to be created.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST hgroup/:hgroup**

Returns A dictionary describing the created hgroup

Return type ResponseDict

create_host(*host*, ***kwargs*)
Create a host are return a dictionary describing it.

Parameters

- **host** (*str*) – Name of host to be created.

- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST host/:host**

Returns A dictionary describing the created host.

Return type ResponseDict

create_kmip(*name*, ***kwargs*)
Create a new kmip configuration.

Parameters

- **name** (*string*) – The name of the KMIP config to operate on.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT kmip**

Returns A list of dictionaries describing the kmip configuration.

Return type ResponseList

Note: Requires use of REST API 1.12 or later.

---

create_pgroup(*pgroup*, ***kwargs*)
Create pgroup with specified name.

Parameters

- **pgroup** (*str*) – Name of pgroup to be created.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST pgroup/:pgroup**

Returns A dictionary describing the created pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

---

create_pgroup_snapshot(*source*, ***kwargs*)
Create snapshot of pgroup from specified source.

Parameters

- **source** (*str*) – Name of pgroup of which to take snapshot.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST pgroup**

Returns A dictionary describing the created snapshot.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

---

create_pgroup_snapshots(*sources*, ***kwargs*)
Create snapshots of pgroups from specified sources.

Parameters

- **sources** (*list of str*) – Names of pgroups of which to take snapshots.

- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST pgroup**

Returns A list of dictionaries describing the created snapshots.

Return type ResponseList

Note: Requires use of REST API 1.2 or later.

---

create_pod(*pod*, ***kwargs*)
Create a pod and return a dictionary describing it.

Parameters

- **pod** (*str*) – Name of the pod to be created.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST pod**

Returns A dictionary describing the created pod.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

create_snapshot(*volume*, ***kwargs*)
Create a snapshot of the given volume.

Parameters

- **volume** (*str*) – Name of the volume of which to take a snapshot.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST volume**

Returns A dictionary describing the new snapshot.

Return type ResponseDict

create_snapshots(*volumes*, ***kwargs*)
Create snapshots of the listed volumes.

Parameters

- **volumes** (*list of str*) – List of names of the volumes to snapshot.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST volume**

Returns A list of dictionaries describing the new snapshots.

Return type ResponseDict

create_snmp_manager(*manager*, *host*, ***kwargs*)
Create an SNMP manager.

Parameters

- **manager** (*str*) – Name of manager to be created.
- **host** (*str*) – IP address or DNS name of SNMP server to be used.

- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST snmp/:manager**

Returns A dictionary describing the created SNMP manager.

Return type ResponseDict

create_subnet(*subnet*, *prefix*, ***kwargs*)
Create a subnet.

Parameters

- **subnet** (*str*) – Name of subnet to be created.
- **prefix** (*str*) – Routing prefix of subnet to be created.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST subnet/:subnet**

Returns A dictionary describing the created subnet.

Return type ResponseDict

Note: prefix should be specified as an IPv4 CIDR address. ("xxx.xxx.xxx.xxx/nn", representing prefix and prefix length)

Note: Requires use of REST API 1.5 or later.

---

create_vgroup(*vgroup*, ***kwargs*)
Create a vgroup.

Parameters

- **vgroup** (*str*) – Name of vgroup to be created.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST vgroup/:vgroup**

Returns A dictionary mapping "name" to vgroup.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

create_vlan_interface(*interface*, *subnet*, ***kwargs*)
Create a vlan interface

Parameters

- **interface** (*str*) – Name of interface to be created.
- **subnet** (*str*) – Subnet associated with interface to be created
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST network/vif/:vlan_interface**

Returns A dictionary describing the created interface

Return type ResponseDict

Note: Requires use of REST API 1.5 or later.

create_volume(*volume*, *size*, ***kwargs*)
Create a volume and return a dictionary describing it.

Parameters

- **volume** (*str*) – Name of the volume to be created.
- **size** (*int or str*) – Size in bytes, or string representing the size of the volume to be created.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST volume/:volume**

Returns A dictionary describing the created volume.

Return type ResponseDict

Note: The maximum volume size supported is 4 petabytes (4 * 2^50).

Note: If size is an int, it must be a multiple of 512.

Note: If size is a string, it must consist of an integer followed by a valid suffix.

Accepted Suffixes

Suffix	Size	Bytes
S	Sector	(2^9)
K	Kilobyte	(2^10)
M	Megabyte	(2^20)
G	Gigabyte	(2^30)
T	Terabyte	(2^40)
P	Petabyte	(2^50)

delete_admin(*admin*)
Delete an admin.

Parameters **admin** (*str*) – Name of admin whose API token is to be deleted.

Returns A dictionary mapping "name" to admin and "api_token" to None.

Return type ResponseDict

Note: Requires use of REST API 1.14 or later.

delete_alert_recipient(*address*)
Delete an alert recipient.

Parameters **address** (*str*) – Email address of alert recipient to be deleted.

Returns A dictionary mapping "name" to address.

Return type ResponseDict

delete_api_token(*admin*)
Delete the API token of an admin.

Parameters **admin** (*str*) – Name of admin whose API token is to be deleted.

Returns A dictionary mapping "name" to admin and "api_token" to None.

Return type ResponseDict

delete_certificate(*name*, ***kwargs*)
Delete a managed certificate.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT cert**

Returns A dictionary describing the configured array certificate.

Return type ResponseDict

Note: Requires use of REST API 1.12 or later.

You may not delete the management certificate.

---

delete_hgroup(*hgroup*)
Delete a host group.

Parameters **hgroup** (*str*) – Name of the hgroup to be deleted.

Returns A dictionary mapping "name" to hgroup.

Return type ResponseDict

delete_host(*host*)
Delete a host.

Parameters **host** (*str*) – Name of host to be deleted.

Returns A dictionary mapping "name" to host.

Return type ResponseDict

delete_kmip(*name*)
Delete an existing kmip configuration.

Parameters **name** (*string*) – The name of the KMIP config to operate on.

Returns A dictionary containing the name of the deleted kmip configuration.

Return type ResponseDict

Note: Requires use of REST API 1.12 or later.

---

delete_snmp_manager(*manager*)
Delete an SNMP manager.

Parameters **manager** (*str*) – Name of SNMP manager to be deleted.

Returns A dictionary mapping "name" to manager.

Return type ResponseDict

delete_subnet(*subnet*)
Delete a subnet.

Parameters **subnet** (*str*) – Name of the subnet to be deleted.

Returns A dictionary mapping "name" to subnet.

Return type ResponseDict

Note: Requires use of REST API 1.5 or later.

---

delete_vlan_interface(*interface*)
Delete a vlan interface.

Parameters **interface** (*str*) – Name of the interface to be deleted.

Returns A dictionary mapping "name" to interface.

Return type ResponseDict

Note: Requires use of REST API 1.5 or later.

---

destroy_pgroup(*pgroup*, ***kwargs*)
Destroy an existing pgroup or pgroup snapshot.

Parameters

- **pgroup** (*str*) – Name of pgroup(snap) to be destroyed.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST pgroup/:pgroup**

Returns A dictionary mapping "name" to pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

---

destroy_pod(*pod*)
Destroy an existing pod.

Parameters **pod** (*str*) – Name of the pod to be destroyed.

Returns A dictionary mapping "name" to pod, and the time remaining before the pod is eradicated.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

destroy_vgroup(*vgroup*)
Destroy an existing vgroup.

Parameters **vgroup** (*str*) – Name of vgroup to be destroyed.

Returns A dictionary mapping "name" to vgroup.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

destroy_volume(*volume*)
Destroy an existing volume or snapshot.

Parameters **volume** (*str*) – Name of the volume to be destroyed.

Returns A dictionary mapping "name" to volume.

Return type ResponseDict

disable_alert_recipient(*address*)
Disable alerts to an alert recipient.

Parameters **address** (*str*) – Email address of alert recipient to be disabled.

Returns A dictionary mapping "name" to address and "enabled" to False.

Return type ResponseDict

disable_app(*name*)
Disable the specified app.

Parameters **name** (*str*) – Name of app to be disabled.

Returns A dictionary describing the app.

Return type ResponseList

disable_app_vnc(*name*)
Disable vnc access for the specified app.

Parameters **name** (*str*) – Name of app to disable vnc access.

Returns A dictionary describing the app.

Return type ResponseList

disable_console_lock()
Disable root lockout from the array at the physical console.

Returns A dictionary mapping "console_lock" to "disabled".

Return type ResponseDict

disable_directory_service(*check_peer=False*)
Disable the directory service.

Parameters **check_peer** (*bool, optional*) – If True, disables server authenticity enforcement. If False, disables directory service integration.

Returns A dictionary describing the status of the directory service.

Return type ResponseDict

disable_network_interface(*interface*)
Disable a network interface.

Parameters **interface** (*str*) – Name of network interface to be disabled.

Returns A dictionary describing the interface.

Return type ResponseDict

disable_pgroup_replication(*pgroup*)
Disable replication schedule for pgroup.

Parameters **pgroup** (*str*) – Name of pgroup for which to disable replication schedule.

Returns A dictionary describing pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

disable_pgroup_snapshots(*pgroup*)
Disable snapshot schedule for pgroup.

Return type ResponseDict

Returns A dictionary describing pgroup.

Note: Requires use of REST API 1.2 or later.

disable_phonehome()
Disable hourly phonehome.

Returns A dictionary mapping "phonehome" to "disabled".

Return type ResponseDict

disable_remote_assist()
Disable remote assist.

Returns A dictionary describing the status of the remote assist connection.

Return type ResponseDict

disable_subnet(*subnet*)
Disable a subnet.

Parameters **subnet** (*str*) – Name of subnet to be disabled.

Returns A dictionary describing the subnet.

Return type ResponseDict

Note: Requires use of REST API 1.5 or later.

disconnect_array(*address*)
Disconnect this array from another one.

Parameters **address** (*str*) – IP address or DNS name of other array.

Returns A dictionary mapping "name" to address.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

disconnect_azure_offload(*name*)
Disconnect an Azure Blob offload target.

Parameters **name** (*str*) – Name of Azure Blob offload target to be disconnected.

Returns A dictionary describing the target.

Return type ResponseDict

disconnect_hgroup(*hgroup*, *volume*)
Delete a shared connection between a host group and a volume.

Parameters

- **hgroup** (*str*) – Name of hgroup to be disconnected from volume.
- **volume** (*str*) – Name of volume to be disconnected from hgroup.

Returns A dictionary mapping "name" to hgroup and "vol" to volume.

Return type ResponseDict

disconnect_host(*host*, *volume*)
Delete a connection between a host and a volume.

Parameters

- **host** (*str*) – Name of host to be disconnected from volume.
- **volume** (*str*) – Name of volume to be disconnected from host.

Returns A dictionary mapping "name" to host and "vol" to volume.

Return type ResponseDict

disconnect_nfs_offload(*name*)
Disconnect an nfs offload target.

Parameters **name** (*str*) – Name of nfs offload target to be disconnected.

Returns A dictionary describing the target.

Return type ResponseDict

disconnect_s3_offload(*name*)
Disconnect an S3 offload target.

Parameters **name** (*str*) – Name of S3 offload target to be disconnected.

Returns A dictionary describing the target.

Return type ResponseDict

enable_alert_recipient(*address*)
Enable alerts to an alert recipient.

Parameters **address** (*str*) – Email address of alert recipient to be enabled.

Returns A dictionary mapping "name" to address and "enabled" to True.

Return type ResponseDict

enable_app(*name*)
Enable the specified app.

Parameters **name** (*str*) – Name of app to be enabled.

Returns A dictionary describing the app.

Return type ResponseList

enable_app_vnc(*name*)
Enable vnc access for the specified app.

Parameters **name** (*str*) – Name of app to enable vnc access.

Returns A dictionary describing the app.

Return type ResponseList

enable_console_lock()
Enable root lockout from the array at the physical console.

Returns A dictionary mapping "console_lock" to "enabled".

Return type ResponseDict

enable_directory_service(*check_peer=False*)
Enable the directory service.

Parameters **check_peer** (*bool, optional*) – If True, enables server authenticity enforcement. If False, enables directory service integration.

Returns A dictionary describing the status of the directory service.

Return type ResponseDict

enable_network_interface(*interface*)
Enable a network interface.

Parameters **interface** (*str*) – Name of network interface to be enabled.

Returns A dictionary describing the interface.

Return type ResponseDict

enable_pgroup_replication(*pgroup*)
Enable replication schedule for pgroup.

Parameters **pgroup** (*str*) – Name of pgroup for which to enable replication schedule.

Returns A dictionary describing pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

---

enable_pgroup_snapshots(*pgroup*)
Enable snapshot schedule for pgroup.

Parameters **pgroup** (*str*) – Name of pgroup for which to enable snapshot schedule.

Returns A dictionary describing pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

---

enable_phonehome()
Enable hourly phonehome.

Returns A dictionary mapping "phonehome" to "enabled".

Return type ResponseDict

enable_remote_assist()
Enable remote assist.

Returns A dictionary describing the status of the remote assist connection.

Return type ResponseDict

enable_subnet(*subnet*)
Enable a subnet.

Parameters **subnet** (*str*) – Name of subnet to be enabled.

Returns A dictionary describing the subnet.

Return type ResponseDict

Note: Requires use of REST API 1.5 or later.

---

eradicate_pgroup(*pgroup*, ***kwargs*)
Eradicate a destroyed pgroup.

Parameters

- **pgroup** (*str*) – Name of pgroup to be eradicated.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **DELETE pgroup/:pgroup**

Returns A dictionary mapping "name" to pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

---

eradicate_pod(*pod*)
Eradicate a destroyed pod.

Parameters **pod** (*str*) – Name of the pod to be eradicated.

Returns A dictionary mapping "name" to pod.

Return type ResponseDict

Note: This operation fails if pod is not destroyed.

Note: Requires use of REST API 1.13 or later.

---

eradicate_vgroup(*vgroup*)
Eradicate a destroyed vgroup.

Parameters **vgroup** (*str*) – Name of vgroup to be eradicated.

Returns A dictionary mapping "name" to vgroup.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

eradicate_volume(*volume*)
Eradicate a destroyed volume or snapshot.

Parameters **volume** (*str*) – Name of the volume to be eradicated.

Returns A dictionary mapping "name" to volume.

Return type ResponseDict

Note: This operation fails if volume is not destroyed.

---

extend_volume(*volume*, *size*)
Extend a volume to a new, larger size.

Parameters

- **volume** (*str*) – Name of the volume to be extended.
- **size** (*int or str*) – Size in bytes, or string representing the size of the volume to be created.

Returns A dictionary mapping "name" to volume and "size" to the volume's new size in bytes.

Return type ResponseDict

Note: The new size must be larger than the volume's old size.

Note: The maximum volume size supported is 4 petabytes (4 * 2^50).

Note: If size is an int, it must be a multiple of 512.

Note: If size is a string, it must consist of an integer followed by a valid suffix.

Accepted Suffixes

Suffix	Size	Bytes
S	Sector	(2^9)
K	Kilobyte	(2^10)
M	Megabyte	(2^20)
G	Gigabyte	(2^30)
T	Terabyte	(2^40)
P	Petabyte	(2^50)

flag_message(*message_id*)
Flag an alert message or audit record.

Parameters **message_id** (*int or str*) – ID of message to be flagged.

Returns A dictionary mapping "id" to message_id.

Return type ResponseDict

get(***kwargs*)
Get array attributes.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET array**

Returns A dictionary describing the array or a list of dictionaries describing multiple array attributes, depending on the arguments passed in.

Return type ResponseDict or ResponseList

get_admin(*admin*)
Returns a dictionary describing an admin.

Parameters **admin** (*str*) – Name of admin to get.

Returns A dictionary mapping "name" to admin and "role" to their role.

Return type ResponseDict

Note: Requires use of REST API 1.14 or later.

---

get_admin_lockout_info(*admin*)
Return a dictionary describing lockout information for a specific admin.

Parameters **admin** (*str*) – Name of admin whose lockout info is requested

Returns A dictionary describing a specific locked admin

Return type ResponseDict

Note: Requires use of REST API 1.16 or later.

---

get_alert_recipient(*address*)
Return a dictionary describing an alert recipient.

Parameters **address** (*str*) – Email address of alert recipient to get information about.

Returns A dictionary mapping "name" to address and "enabled" to True if that alert recipient is enabled, False otherwise.

Return type ResponseDict

get_api_token(*admin*, ***kwargs*)
Return a dictionary describing an admin's API token.

Parameters

- **admin** (*str*) – Name of admin whose API token to get.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET admin/:admin/apitoken**

Returns A dictionary describing admin's API token.

Return type ResponseDict

get_app(*name*, ***kwargs*)
Returns a list of dictionaries describing the app.

Parameters

- **name** (*string*) – The name of the app.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET app/:app**

Returns A list of dictionaries describing the app.

Return type ResponseList

get_app_node(*app_name*, ***kwargs*)
Returns a list of dictionaries describing the nodes of an app.

Parameters

- **app_name** (*string*) – The name of the app.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET app/:app/nodes**

Returns A list of dictionaries describing the nodes of an app.

Return type ResponseList

get_app_software(*name*, ***kwargs*)
List the specified app software.

Parameters

- **name** (*string*) – The name of the app.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET software/app/:app**

Returns A list of dictionaries describing the app.

Return type ResponseList

get_azure_offload(*name*, ***kwargs*)
Return a dictionary describing the connected Azure Blob offload target.

Parameters

- **offload** (*str*) – Name of Azure Blob offload target to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET azure_offload/::offload**

Returns A dictionary describing the Azure Blob offload connection.

Return type ResponseDict

get_certificate(***kwargs*)
Get the attributes of the current array certificate.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET cert**

Returns A dictionary describing the configured array certificate.

Return type ResponseDict

Note: Requires use of REST API 1.3 or later.

---

get_certificate_signing_request(***kwargs*)
Construct a certificate signing request (CSR) for signing by a certificate authority (CA).

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET cert/certificate_signing_request**

Returns A dictionary mapping "certificate_signing_request" to the CSR.

Return type ResponseDict

Note: Requires use of REST API 1.3 or later.

In version 1.12, purecert was expanded to allow manipulation of multiple certificates, by name. To preserve backwards compatibility, the default name, if none is specified, for this version is 'management' which acts on the certificate previously managed by this command.

---

get_console_lock_status()
Get console-lock status of the array.

Returns A dictionary mapping "console_lock" to "enabled" if console_lock is enabled, else "disabled".

Return type ResponseDict

get_directory_service(***kwargs*)
Return a dictionary describing directory service configuration.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET directoryservice**

Returns A dictionary describing the status of the directory service.

Return type ResponseDict

get_dns()
Get current DNS settings.

Returns A dictionary describing current DNS settings.

Return type ResponseDict

get_drive(*drive*)
Get drive attributes.

Parameters **drive** (*str*) – Name of drive to get information about.

Returns A dictionary describing drive.

Return type ResponseDict

get_eula()
Get EULA status of the array.

Returns EULA agreement and a dictionary describing the EULA acceptance status on the array.

Return type ResponseDict

get_global_admin_attributes()
Return a dictionary describing the existing global admin attributes.

Returns A dictionary describing the existing global admin attributes.

Return type ResponseDict

Note: Requires use of REST API 1.16 or later.

---

get_hardware(*component*, ***kwargs*)
Returns a dictionary describing a hardware component.

Parameters

- **component** (*str*) – Name of hardware component to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET hardware/:component**

Returns A dictionary describing component.

Return type ResponseDict

get_hgroup(*hgroup*, ***kwargs*)
Return a list of dictionaries describing a host group.

Parameters

- **hgroup** (*str*) – Name of hgroup to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET hgroup/:hgroup**

Returns A dictionary describing hgroup.

Return type ResponseDict

get_host(*host*, ***kwargs*)
Return a dictionary describing a host.

Parameters

- **host** (*str*) – Name of host to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET host/:host**

Returns A dictionary describing host.

Return type ResponseDict

get_kmip(*name*, ***kwargs*)
Show an existing kmip configuration.

Parameters

- **name** (*string*) – The name of the KMIP config to operate on.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET kmip/:kmip**

Returns A list of dictionaries containing the requested kmip configuration.

Return type ResponseList

Note: Requires use of REST API 1.12 or later.

---

get_manual_phonehome_status()
Get manually-initiated phonehome status.

Returns A dictionary describing the current status of a manually-initiated phonehome.

Return type ResponseDict

get_network_interface(*interface*, ***kwargs*)
Return a dictionary describing a network interface.

Parameters

- **interface** (*str*) – Name of network interface to get information about.

- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET network/:network_component**

Returns A dictionary describing the interface.

Return type ResponseDict

get_nfs_offload(*name*, ***kwargs*)
Return a dictionary describing the connected nfs offload target.

Parameters

- **offload** (*str*) – Name of NFS offload target to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET nfs_offload/::offload**

Returns A dictionary describing the nfs offload connection.

Return type ResponseDict

get_offload(*name*, ***kwargs*)
Return a dictionary describing the connected offload target.

Parameters

- **offload** (*str*) – Name of offload target to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET offload/::offload**

Returns A dictionary describing the offload connection.

Return type ResponseDict

get_pgroup(*pgroup*, ***kwargs*)
Return dictionary describing a pgroup or snapshot.

Parameters

- **pgroup** (*str*) – Name of pgroup to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET pgroup**

Returns A list describing snapshots of the pgroup if the paramater snap is passed as True, else a dictionary describing the pgroup.

Return type ResponseDict or ResponseList

Note: Requires use of REST API 1.2 or later.

---

get_phonehome()
Return a dictionary describing if hourly phonehome is enabled.

Returns A dictionary mapping "phonehome" to "enabled" if hourly phonehome is enabled, mapping to "disabled" otherwise.

Return type ResponseDict

get_pod(*pod*, ***kwargs*)
Return a dictionary describing a pod.

Parameters

- **pod** (*str*) – Name of the pod to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET pod/:pod**

Returns A dictionary describing the pod.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

get_publickey(*admin*)
Returns a dictionary describing an admin's public key.

Parameters **admin** (*str*) – Name of admin whose public key to get.

Returns A dictionary mapping "name" to admin and "publickey" to "****".

Return type ResponseDict

get_remote_assist_status()
Return a dictionary describing whether remote assist is enabled.

Returns A dictionary describing the current status of the remote assist connection.

Return type ResponseDict

get_rest_version()
Get the REST API version being used by this object.

Returns The REST API version.

Return type str

get_s3_offload(*name*, ***kwargs*)
Return a dictionary describing the connected S3 offload target.

Parameters

- **offload** (*str*) – Name of S3 offload target to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET s3_offload/::offload**

Returns A dictionary describing the S3 offload connection.

Return type ResponseDict

get_smtp()
Get the attributes of the current smtp server configuration.

Parameters ****kwargs** – See the REST API Guide on your array for the documentation on the request: **GET smtp**

Returns A dictionary describing the smtp server configuration.

Return type ResponseDict

Note: Requires use of REST API 1.14 or later.

---

get_snmp_engine_id()
Return the SNMP v3 engine ID generated for the array.

Returns A dictionary mapping "engine_id" to the array's SNMP engine ID.

Return type ResponseDict

Note: Requires use of SNMP v3.

---

get_snmp_manager(*manager*)
Return a dictionary describing an SNMP manager.

Parameters **manager** (*str*) – Name of SNMP manager to get information about.

Returns A dictionary describing manager.

Return type ResponseDict

get_subnet(*subnet*)
Return a dictionary describing a subnet.

Parameters **subnet** (*str*) – Name of the subnet to get information about.

Returns A dictionary describing the subnet.

Return type ResponseDict

Note: Requires use of REST API 1.5 or later.

---

get_vgroup(*vgroup*, ***kwargs*)
Return dictionary describing a vgroup.

Parameters

- **vgroup** (*str*) – Name of vgroup to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET vgroup**

Returns A list describing a dictionary describing the vgroup.

Return type ResponseDict or ResponseList

Note: Requires use of REST API 1.13 or later.

---

get_volume(*volume*, ***kwargs*)
Return a dictionary describing a volume or snapshot.

Parameters

- **volume** (*str*) – Name of the volume to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET volume/:volume**

Returns A list describing snapshots of the volume if the paramater snap is passed as True, else a dictionary describing the volume.

Return type ResponseDict or ResponseList

install_app(*name*, ***kwargs*)
Install the specified app.

Parameters **name** (*string*) – The name of the app.

Returns A dictionary describing the app.

Return type ResponseList

invalidate_cookie()
End the REST API session by invalidating the current session cookie.

Note: Calling any other methods again creates a new cookie. This method is intended to be called when the FlashArray object is no longer needed.

list_admins(***kwargs*)
Return a list of dictionaries describing local admins.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET admin**

Returns A list of dictionaries mapping "name" to a username and "role" to their role for each local admin on the array.

Return type ResponseList

Note: Requires use of REST API 1.14 or later.

list_alert_recipients()
Return a list of dictionaries describing alert recipients.

Returns A list of dictionaries mapping "name" to a recipient's address and "enabled" to True if that recipient is enabled, False otherwise, for each alert recipient.

Return type ResponseList

list_api_tokens(***kwargs*)
Return a list of dictionaries describing REST API tokens.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET admin**

Returns A list of dictionaries describing the API token of each admin with an API token set.

Return type ResponseList

Note: The API tokens are replaced with "****" unless the parameter expose is passed as True.

list_app_nodes(***kwargs*)
Returns a list of dictionaries describing app nodes.

Parameters ****kwargs** (*optional*) – See the REST API Guid on your array for the documentation on the request: **GET app/nodes**

Returns A list of dictionaries describing app nodes.

Return type ResponseList

list_app_software(***kwargs*)
List app software.

Parameters

- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET software/app**
- **app** (*str*) – Name of app to get information about.

Returns A dictionary describing app.

Return type ResponseDict

list_apps(***kwargs*)
Returns a list of dictionaries describing apps.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET app**

Returns A list of dictionaries describing each app.

Return type ResponseList

list_array_connections(***kwargs*)
Return list of connected arrays.

Returns A list of dictionaries describing each connection to another array.

Return type ResponseList

Note: Requires use of REST API 1.2 or later.

---

list_azure_offload(***kwargs*)
Return a list of dictionaries describing connected Azure Blob offload targets.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET azure_offload**

Returns A list of dictionaries describing Azure Blob offload connections.

Return type ResponseList

list_certificates()
Get the attributes of the current array certificate.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET cert**

Returns A list of dictionaries describing all configured certificates.

Return type ResponseList

Note: Requires use of REST API 1.12 or later.

---

list_directory_service_roles(***kwargs*)
Get directory service groups for roles.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET directoryservice/role**

Returns A list of dictionaries describing the group and group base for each role.

Return type ResponseList

Note: Requires use of REST API 1.16 or later.

list_drives()
Returns a list of dictionaries describing SSD and NVRAM modules.

Returns A list of dictionaries describing each drive.

Return type ResponseList

list_hardware(***kwargs*)
Returns a list of dictionaries describing hardware.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET hardware**

Returns A list of dictionaries describing each hardware component.

Return type ResponseList

list_hgroup_connections(*hgroup*)
Return a list of dictionaries describing shared connected volumes.

Parameters **hgroup** (*str*) – Name of hgroup for which to list connections.

Returns A list of dictionaries describing hgroup's connections.

Return type ResponseList

list_hgroups(***kwargs*)
Return a list of dictionaries describing each host group.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET hgroup**

Returns A list of dictionaries describing each hgroup.

Return type ResponseList

list_host_connections(*host*, ***kwargs*)
Return a list of dictionaries describing connected volumes.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET host/:host/volume**

Returns A list of dictionaries describing host's connections.

Return type ResponseList

list_hosts(***kwargs*)
Return a list of dictionaries describing each host.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET host**

Returns A list of dictionaries describing each host.

Return type ResponseList

list_kmip()
Show all existing kmip configurations.

Returns A list of dictionaries containing the requested kmip configuration.

Return type ResponseList

Note: Requires use of REST API 1.12 or later.

list_locked_admins_lockout_info()
Return a list of dictionaries describing lockout information for locked admins.

Returns A list of dictionaries describing all the locked admins

Return type ResponseList

Note: Requires use of REST API 1.16 or later.

list_maintenance_windows(***kwargs*)
Return a list of dictionaries describing each maintenance window.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET maintenance_window**

Returns A list of dictionaries describing each maintenance window.

Return type ResponseList

list_messages(***kwargs*)
Return a list of alert messages.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET message**

Returns A list of dictionaries describing each message.

Return type ResponseList

list_network_interfaces(***kwargs*)
Get a list of dictionaries describing network interfaces.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET network**

Returns A list of dictionaries describing each network interface.

Return type ResponseList

list_nfs_offload(***kwargs*)
Return a list of dictionaries describing connected nfs offload targets.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET nfs_offload**

Returns A list of dictionaries describing NFS offload connections.

Return type ResponseList

list_offload(***kwargs*)
Return a list of dictionaries describing connected offload targets.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET offload**

Returns A list of dictionaries describing offload connections.

Return type ResponseList

list_pgroups(***kwargs*)
Return list dictionaries describing each pgroup.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET pgroup**

Returns A list of dictionaries describing each pgroup.

Return type ResponseList

Note: Requires use of REST API 1.2 or later.

---

list_pods(***kwargs*)
Return a list of dictionaries describing each pod.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET pod**

Returns A list of dictionaries describing each pod.

Return type ResponseList

Note: Requires use of REST API 1.13 or later.

---

list_ports(***kwargs*)
List SAN ports.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET port**

Returns A list of dictionaries describing each port.

Return type ResponseList

list_publickeys()
Return a list of dictionaries describing public keys.

Returns A list of dictionaries mapping "name" to a username and "publickey" to "****" for each admin with a public key set.

Return type ResponseList

list_s3_offload(***kwargs*)
Return a list of dictionaries describing connected S3 offload targets.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET se_offload**

Returns A list of dictionaries describing S3 offload connections.

Return type ResponseList

list_snmp_managers()
Return a list of dictionaries describing SNMP managers.

Returns A list of dictionaries describing each SNMP manager.

Return type ResponseList

list_subnets(***kwargs*)
Get a list of dictionaries describing subnets.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET subnet**

Returns A list of dictionaries describing each subnet.

Return type ResponseList

Note: Requires use of REST API 1.5 or later.

---

list_vgroups(***kwargs*)
Return list dictionaries describing each vgroup.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET vgroup**

Returns A list of dictionaries describing each vgroup.

Return type ResponseList

Note: Requires use of REST API 1.13 or later.

---

list_volume_block_differences(*volume*, ***kwargs*)
Return a list of block differences for the specified volume.

Parameters

- **volume** (*str*) – Name of the volume to get information about.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET volume/:volume/diff**

Returns A list of dictionaries describing block differences between the specified volume and the base volume.

Return type ResponseList

Note: Requires use of REST API 1.3 or later.

---

list_volume_private_connections(*volume*, ***kwargs*)
Return a list of dictionaries describing connected hosts.

Parameters **volume** (*str*) – Name of the volume for which to list the private connections.

Returns A list of dictionaries describing the volume's private connections.

Return type ResponseList

list_volume_shared_connections(*volume*, ***kwargs*)
Return a list of dictionaries describing connected host groups.

Parameters **volume** (*str*) – Name of the volume for which to list the shared connections.

Returns A list of dictionaries describing the volume's shared connections.

Return type ResponseList

list_volumes(***kwargs*)
Return a list of dictionaries describing each volume.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET volume**

Returns A list of dictionaries describing each volume.

Return type ResponseList

move_volume(*volume*, *container*)
Move a volume to a new pod or vgroup

Parameters

- **volume** (*str*) – Name of the volume to move.
- **container** (*str*) – Destination container of the move, either a pod, a vgroup or "" for the local array.

Returns a dictionary describing the volume, with new container reflected in new name.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

static page_through(*page_size*, *function*, **args*, ***kwargs*)
Return an iterator over all pages of a REST operation.

Parameters

- **page_size** – Number of elements to retrieve per call.
- **function** – FlashArray function that accepts limit as an argument.
- ***args** – Positional arguments to be passed to function.
- ****kwargs** – Keyword arguments to be passed to function.

Returns An iterator of tuples containing a page of results for the function(*args, **kwargs) and None, or None and a PureError if a call to retrieve a page fails.

Return type iterator

Note: Requires use of REST API 1.7 or later.

Only works with functions that accept limit as an argument.

Iterator will retrieve page_size elements per call

Iterator will yield None and an error if a call fails. The next call will repeat the same call, unless the caller sends in an alternate page token.

---

phonehome(*action*)
Manually initiate or cancel a phonehome action.

Parameters **action** (*str*) – The timeframe of logs to phonehome or cancel the current phonehome.

Note: action must be one of: ("send_today", "send_yesterday", "send_all", "cancel").

Returns A dictionary describing the current status of the phonehome request.

Return type ResponseDict

recover_pgroup(*pgroup*, ***kwargs*)
Recover a destroyed pgroup that has not yet been eradicated.

Parameters

- **pgroup** (*str*) – Name of pgroup to be recovered.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT pgroup/:pgroup**

Returns A dictionary mapping "name" to pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

---

recover_pod(*pod*)
Recover a pod that has been destroyed but not eradicated.

Parameters **pod** (*str*) – Name of pod to be recovered.

Returns A dictionary mapping "name" to pod, and the time remaining which will now be null.

Return type ResponseDict

Note: This must be done within the 24 hours following a pod's destruction or it will be eradicated.

Note: Requires use of REST API 1.13 or later.

---

recover_vgroup(*vgroup*)
Recover a destroyed vgroup that has not yet been eradicated.

Parameters **vgroup** (*str*) – Name of vgroup to be recovered.

Returns A dictionary mapping "name" to vgroup.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

recover_volume(*volume*)
Recover a volume that has been destroyed but not eradicated.

Parameters **volume** (*str*) – Name of volume to be recovered.

Returns A dictionary mapping "name" to volume.

Return type ResponseDict

Note: This must be done within the 24 hours following a volume's destruction or it will be eradicated.

---

refresh_admin(*admin*, ***kwargs*)
Refresh the admin permission cache for the specified admin.

Parameters

- **admin** (*str*) – Name of admin whose permission cache is to be refreshed.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **GET admin**

Returns A dictionary mapping "name" to admin and "role" to the admin's role.

Return type ResponseDict

Note: Setting the optional parameter clear to True only clears the cache without doing an LDAP lookup to get new permissions.

---

refresh_admins()
Clear the admin permission cache.

Returns A dictionary mapping "name" to "[ALL]" and "role" to None.

Return type ResponseDict

Note: Does not immediately do any LDAP lookups to get new permissions.

---

remove_hgroup(*hgroup*, *pgroup*)
Remove an hgroup from a pgroup.

Parameters

- **hgroup** (*str*) – Name of the hgroup to remove from pgroup.
- **pgroup** (*str*) – pgroup from which to remove hgroup.

Returns A dictionary mapping "name" to hgroup and "protection_group" to pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.7 or later.

---

remove_host(*host*, *pgroup*)
Remove a host from a pgroup.

Parameters

- **host** (*str*) – Name of the host to remove from pgroup.
- **pgroup** (*str*) – pgroup from which to remove host.

Returns A dictionary mapping "name" to host and "protection_group" to pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.7 or later.

---

remove_pod(*pod*, *array*, ***kwargs*)
Remove arrays from a pod.

Parameters

- **pod** (*str*) – Name of the pod.

- **array** (*str*) – Array to remove from pod.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **DELETE pod/:pod/**array/:array**

Returns A dictionary mapping "name" to pod and "array" to the pod's new array list.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

remove_volume(*volume*, *pgroup*)
Remove a volume from a pgroup.

Parameters

- **volume** (*str*) – Name of the volume to remove from pgroup.
- **pgroup** (*str*) – pgroup from which to remove volume.

Returns A dictionary mapping "name" to volume and "protection_group" to pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.7 or later.

---

rename(*name*)
Rename the array.

Parameters

- **name** – The new name for the array.
- **type** – str

Returns A dictionary mapping "array_name" to name.

Return type ResponseDict

rename_hgroup(*hgroup*, *name*)
Rename a host group.

Parameters

- **hgroup** (*str*) – Name of hgroup to be renamed.
- **name** (*str*) – New name of hgroup to be renamed.

Returns A dictionary mapping "name" to name.

Return type ResponseDict

rename_host(*host*, *name*)
Rename a host.

Parameters

- **host** (*str*) – Name of host to be renamed.
- **name** (*str*) – New name of host to be renamed.

Returns A dictionary mapping "name" to name.

Return type ResponseDict

rename_pgroup(*pgroup*, *name*)
Rename a pgroup.

Parameters

- **pgroup** (*str*) – Current name of pgroup to be renamed.
- **name** (*str*) – New name of pgroup to be renamed.

Returns A dictionary mapping "name" to name.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

rename_pod(*pod*, *name*)
Rename a pod.

Parameters

- **pod** (*str*) – Name of the pod to be renamed.
- **name** (*str*) – New name of pod to be renamed.

Returns A dictionary mapping "name" to name.

Return type ResponseDict

Note: All pod objects in the named pod also are renamed to the new name, followed by its previous suffix.

Note: Requires use of REST API 1.13 or later.

rename_snmp_manager(*manager*, *name*)
Rename an SNMP manager.

Parameters

- **manager** (*str*) – Current name of the SNMP manager to be renamed.
- **name** (*str*) – New name of the SNMP manager to be renamed.

Returns A dictionary describing the renamed SNMP manager.

Return type ResponseDict

rename_subnet(*subnet*, *name*)
Rename a subnet.

Parameters

- **subnet** (*str*) – Current name of the subnet to be renamed.
- **name** (*str*) – New name of the subnet to be renamed.

Returns A dictionary describing the renamed subnet.

Return type ResponseDict

Note: Requires use of REST API 1.5 or later.

rename_vgroup(*vgroup*, *name*)
Rename a vgroup.

Parameters

- **vgroup** (*str*) – Current name of vgroup to be renamed.
- **name** (*str*) – New name of vgroup to be renamed.

Returns A dictionary mapping "name" to name.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

rename_volume(*volume*, *name*)
Rename a volume.

Parameters

- **volume** (*str*) – Name of the volume to be renamed.
- **name** (*str*) – New name of volume to be renamed.

Returns A dictionary mapping "name" to name.

Return type ResponseDict

Note: All snapshots of the named volume also are renamed to the new name, followed by its previous suffix.

---

schedule_maintenance_window(***kwargs*)
Schedule a maintenance window and return a dictionary describing it.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST maintenance_window**

Returns A dictionary describing the scheduled maintenance window.

Return type ResponseDict

send_pgroup_snapshot(*source*, ***kwargs*)
Send an existing pgroup snapshot to target(s)

Parameters

- **source** (*str*) – Name of pgroup snapshot to send.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **POST pgroup**

Returns A list of dictionaries describing the sent snapshots.

Return type ResponseList

Note: Requires use of REST API 1.16 or later.

---

set(***kwargs*)
Set array attributes.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT array**

Returns A dictionary mapping the parameter that was set to its new value.

Return type ResponseDict

set_admin(*admin*, ***kwargs*)
Set an attribute of an admin.

Parameters

- **admin** (*str*) – Name of admin for whom to set an attribute.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT admin/:admin**

Returns A dictionary describing the admin.

Return type ResponseDict

Note: Requires use of REST API 1.14 or later.

---

set_certificate(***kwargs*)
Modify an existing certificate, creating a new self signed one or importing a certificate signed by a certificate authority (CA).

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT cert**

Returns A dictionary describing the configured array certificate.

Return type ResponseDict

Note: Requires use of REST API 1.3 or later.

In version 1.12, purecert was expanded to allow manipulation of multiple certificates, by name. To preserve backwards compatibility, the default name, if none is specified, for this version is 'management' which acts on the certificate previously managed by this command.

---

set_directory_service(***kwargs*)
Set an attribute of the directory service configuration.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT directoryservice**

Returns A dictionary describing the status of the directory service.

Return type ResponseDict

set_directory_service_roles(***kwargs*)
Set directory service groups for roles.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT directoryservice/role**

Returns A list of dictionaries describing the group and group base for each role changed.

Return type ResponseList

Note: Requires use of REST API 1.16 or later.

set_dns(***kwargs*)
Set DNS settings.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT dns**

Returns A dictionary describing current DNS settings.

Return type ResponseDict

set_eula(***kwargs*)
Accept EULA.

Parameters ****kwargs** – See the REST API Guide on your array for the documentation on the request: **PUT array/eula**

Returns EULA agreement and a dictionary describing the EULA acceptance status on the array.

Return type ResponseDict

set_global_admin_attributes(***kwargs*)
Set the global admin attributes.

Parameters ****kwargs** – See the REST API Guide on your array for the documentation on the request: **PUT admin/settings**

Returns A dictionary describing the global admin attributes.

Return type ResponseDict

Note: Requires use of REST API 1.16 or later.

set_hardware(*component*, ***kwargs*)
Set an attribute of a hardware component.

Parameters

- **component** (*str*) – Name of component for which to set attribute.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT hardware/:component**

Returns A dictionary describing component.

Return type ResponseDict

set_hgroup(*hgroup*, ***kwargs*)
Set an attribute of a host group.

Parameters

- **hgroup** (*str*) – Name of hgroup for which to set attribute.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT hgroup/:hgroup**

Returns A dictionary describing hgroup.

Return type ResponseDict

set_host (*host*, ***kwargs*)
Set an attribute of a host.

Parameters

- **host** (*str*) – Name of host for which to set attribute.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT host/:host**

Returns A dictionary describing host.

Return type ResponseDict

set_kmip (*name*, ***kwargs*)
Modify an existing kmip configuration.

Parameters

- **name** (*string*) – The name of the KMIP config to operate on.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT kmip**

Returns A list of dictionaries describing the modified kmip configuration.

Return type ResponseList

Note: Requires use of REST API 1.12 or later.

---

set_network_interface (*interface*, ***kwargs*)
Set network interface attributes.

Parameters

- **interface** (*str*) – Name of network interface for which to set attribute.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT network/:network_component**

Returns A dictionary describing the interface.

Return type ResponseDict

set_password (*admin*, *new_password*, *old_password*)
Set an admin's password.

Parameters

- **admin** (*str*) – Name of admin whose password is to be set.
- **new_password** (*str*) – New password for admin.
- **old_password** (*str*) – Current password of admin.

Returns A dictionary mapping "name" to admin.

Return type ResponseDict

set_pgroup (*pgroup*, ***kwargs*)
Set an attribute of a pgroup.

Parameters

- **pgroup** (*str*) – Name of pgroup for which to set attribute.

- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT pgroup/:pgroup**

Returns A dictionary describing pgroup.

Return type ResponseDict

Note: Requires use of REST API 1.2 or later.

---

set_pod(*pod*, ***kwargs*)
Perform actions on a pod and return a dictionary describing it.

Parameters

- **pod** (*str*) – Name of the for which to set attribute.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT pod/:pod**

Returns A dictionary describing the created pod.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

set_publickey(*admin*, *key*)
Set the public key associated with an admin.

Parameters

- **admin** (*str*) – Name of admin whose public key is to be set.
- **key** (*str*) – New public key for admin.

Returns A dictionary mapping "name" to admin and "publickey" to "****"

Return type ResponseDict

set_smtp(***kwargs*)
Set the attributes of the current smtp server configuration.

Parameters ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT smtp**

Returns A dictionary describing the smtp server configuration.

Return type ResponseDict

Note: Requires use of REST API 1.14 or later.

---

set_snmp_manager(*manager*, ***kwargs*)
Set an attribute of an SNMP manager.

Parameters

- **manager** (*str*) – Name of the SNMP manager for which to set an attribute.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT snmp/:manager**

Returns A dictionary describing manager.

Return type ResponseDict

set_subnet(*subnet*, ***kwargs*)
Set subnet attributes.

Parameters

- **subnet** (*str*) – Name of subnet for which to set attribute.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT subnet/:subnet**

Returns A dictionary describing the subnet.

Return type ResponseDict

Note: Requires use of REST API 1.5 or later.

---

set_vgroup(*vgroup*, ***kwargs*)
Set an attribute of a vgroup.

Parameters

- **vgroup** (*str*) – Name of vgroup for which to set attribute.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT vgroup/:vgroup**

Returns A dictionary describing vgroup.

Return type ResponseDict

Note: Requires use of REST API 1.13 or later.

---

set_volume(*volume*, ***kwargs*)
Perform actions on a volume and return a dictionary describing it.

Parameters

- **volume** (*str*) – Name of the volume to be modified.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT volume/:volume**

Returns A dictionary describing the created volume.

Return type ResponseDict

test_alert()
Send test alerts to all recipients.

Returns A list of dictionaries describing the test outcome for each recipient.

Return type ResponseList

test_alert_recipient(*address*)
Send a test alert to the specified recipient.

Parameters **address** (*str*) – Address of recipient of test alert.

Returns A dictionary describing the test outcome.

Return type ResponseDict

test_directory_service()
Test the directory service.

Returns A dictionary mapping "output" to the output of the directory service test.

Return type ResponseDict

test_kmip(*name*)
Test a given kmip configuration.

Parameters **name** (*string*) – The name of the KMIP config to operate on.

Returns A list of dictionaries containing per-server kmip test results.

Return type ResponseList

Note: Requires use of REST API 1.12 or later.

---

test_snmp_manager(*manager*)
Send a test trap to a manager.

Parameters **manager** (*str*) – SNMP manager to which to send a test trap.

Returns A dictionary mapping "output" to the output of the test.

Return type ResponseDict

throttle_array_connection(*address*, ***kwargs*)
Set bandwidth limits on a connection.

Parameters

- **address** (*str*) – IP address or DNS name of other array.
- ****kwargs** (*optional*) – See the REST API Guide on your array for the documentation on the request: **PUT array/connection/:address**

Returns A dictionary describing the connection to the other array.

Return type ResponseDict

Note: Requires use of REST API 1.5 or later.

---

truncate_volume(*volume*, *size*)
Truncate a volume to a new, smaller size.

Parameters

- **volume** (*str*) – Name of the volume to truncate.
- **size** (*int or str*) – Size in bytes, or string representing the size of the volume to be created.

Returns A dictionary mapping "name" to volume and "size" to the volume's new size in bytes.

Return type ResponseDict

Note: A snapshot of the volume in its previous state is taken and immediately destroyed, but it is available for recovery for the 24 hours following the truncation.

---

uninstall_app(*name*, ***kwargs*)
Uninstall the specified app.

Parameters **name** (*string*) – The name of the app.

Returns A dictionary describing the app.

Return type ResponseList

unlock_admin(*admin*)
Unlocks an admin

Parameters **admin** (*str*) – Name of admin to unlock

Returns A dictionary describing the newly unlocked admin

Return type ResponseDict

Note: Requires use of REST API 1.16 or later.

---

unschedule_maintenance_window(***kwargs*)
Unschedule a maintenance window and return a dictionary describing it.

param **kwargs: See the REST API Guide on your array for the documentation on the request: **DELETE maintenance_window**

Returns A dictionary describing the unscheduled maintenance window.

Return type ResponseDict

class purestorage.**PureError**(*reason*)
Exception type raised by FlashArray object.

Parameters **reason** (*str*) – A message describing why the error occurred.

Variables **reason** (*str*) – A message describing why the error occurred.

class purestorage.**PureHTTPError**(*target*, *rest_version*, *response*)
Exception raised as a result of non-200 response status code.

Parameters

- **target** (*str*) – IP or DNS name of the array that received the HTTP request.
- **rest_version** (*str*) – The REST API version that was used when making the request.
- **response** (requests.Response) – The response of the HTTP request that caused the error.

Variables

- **target** (*str*) – IP or DNS name of the array that received the HTTP request.
- **rest_version** (*str*) – The REST API version that was used when making the request.
- **code** (*int*) – The HTTP response status code of the request.
- **headers** (*dict*) – A dictionary containing the header information. Keys are case-insensitive.
- **reason** (*str*) – The textual reason for the HTTP status code (e.g. "BAD REQUEST").
- **text** (*str*) – The body of the response which may contain a message explaining the error.

Note: The error message in text is not guaranteed to be consistent across REST versions, and thus should not be programmed against.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

CHAPTER 5

Change Log

A log of changes by version and date.

Version	Date	Notes
1.2.0	10/16/2014	Initial Version
1.4.0	12/15/2014	Add support for REST 1.3 and 1.4
1.6.0	05/17/2016	Add support for REST 1.5 and 1.6
1.6.1	05/30/2016	Minor bug fixes and sync code between GitHub and PyPI
1.8.0	11/02/2016	Add support for REST 1.7 and 1.8
1.11.0	09/05/2017	Add support for REST 1.9, 1.10, and 1.11
1.11.1	09/06/2017	Minor setup.py change for new PyPI interface
1.11.2	09/07/2017	Minor setup.py change for new PyPI interface
1.11.3	09/07/2017	Minor setup.py change for new PyPI interface
1.14.0	04/27/2018	Add support for REST 1.12, 1.13, and 1.14
1.14.1	07/09/2018	Add support for managing offload targets.
1.16.0	10/26/2018	Add support for REST 1.15 and 1.16
1.17.0	08/23/2019	Add support for REST 1.17

CHAPTER 6

Overview

The **Pure Storage FlashArray REST Client** is a python module that simplifies integration with the Pure Storage FlashArray REST interface.

It wraps REST calls with simple APIs and abstracts the HTTP request and response handling. For specifics on API arguments, consult the REST API guide for the Purity release currently running on the target array.

This documentation should be supplemental and attempts to explain installation, basic usage, and provides a glossary of the exposed APIs.

Installation Guide How to get the source code, and how to build or install the python package.

Quick Start Guide A quick start guide for the REST client.

API Glossary A glossary of all exposed REST client APIs.

CHAPTER 7

Community

To learn what other Pure Storage customers are doing with the client, to share an implementation or contribute code, or to interact with Pure Storage, visit the Pure Storage Community.

CHAPTER 8

Changes

See the *Change Log* for a list of changes to the REST Client.

CHAPTER 9

License

This code is licensed under the simple BSD 2-Clause License.

See the LICENSE.txt file in the top level of the source tree.

CHAPTER 10

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

purestorage, 9

Index

A

add_hgroup() (*purestorage.FlashArray method*), 10
add_host() (*purestorage.FlashArray method*), 10
add_pod() (*purestorage.FlashArray method*), 10
add_volume() (*purestorage.FlashArray method*), 11

C

clear_message() (*purestorage.FlashArray method*), 11
clone_pod() (*purestorage.FlashArray method*), 11
connect_array() (*purestorage.FlashArray method*), 11
connect_azure_offload() (*purestorage.FlashArray method*), 12
connect_hgroup() (*purestorage.FlashArray method*), 12
connect_host() (*purestorage.FlashArray method*), 12
connect_nfs_offload() (*purestorage.FlashArray method*), 12
connect_s3_offload() (*purestorage.FlashArray method*), 13
copy_volume() (*purestorage.FlashArray method*), 13
create_admin() (*purestorage.FlashArray method*), 13
create_alert_recipient() (*purestorage.FlashArray method*), 13
create_api_token() (*purestorage.FlashArray method*), 13
create_certificate() (*purestorage.FlashArray method*), 14
create_conglomerate_volume() (*purestorage.FlashArray method*), 14
create_hgroup() (*purestorage.FlashArray method*), 14
create_host() (*purestorage.FlashArray method*), 14
create_kmip() (*purestorage.FlashArray method*), 15
create_pgroup() (*purestorage.FlashArray method*), 15
create_pgroup_snapshot() (*purestorage.FlashArray method*), 15
create_pgroup_snapshots() (*purestorage.FlashArray method*), 15
create_pod() (*purestorage.FlashArray method*), 16
create_snapshot() (*purestorage.FlashArray method*), 16
create_snapshots() (*purestorage.FlashArray method*), 16
create_snmp_manager() (*purestorage.FlashArray method*), 16
create_subnet() (*purestorage.FlashArray method*), 17
create_vgroup() (*purestorage.FlashArray method*), 17
create_vlan_interface() (*purestorage.FlashArray method*), 17
create_volume() (*purestorage.FlashArray method*), 18

D

delete_admin() (*purestorage.FlashArray method*), 18
delete_alert_recipient() (*purestorage.FlashArray method*), 18
delete_api_token() (*purestorage.FlashArray method*), 19
delete_certificate() (*purestorage.FlashArray method*), 19
delete_hgroup() (*purestorage.FlashArray method*), 19
delete_host() (*purestorage.FlashArray method*), 19
delete_kmip() (*purestorage.FlashArray method*), 19
delete_snmp_manager() (*purestorage.FlashArray method*), 19
delete_subnet() (*purestorage.FlashArray method*), 19
delete_vlan_interface() (*purestorage.FlashArray method*), 20

destroy_pgroup() (*purestorage.FlashArray method*), 20
destroy_pod() (*purestorage.FlashArray method*), 20
destroy_vgroup() (*purestorage.FlashArray method*), 20
destroy_volume() (*purestorage.FlashArray method*), 21
disable_alert_recipient() (*purestorage.FlashArray method*), 21
disable_app() (*purestorage.FlashArray method*), 21
disable_app_vnc() (*purestorage.FlashArray method*), 21
disable_console_lock() (*purestorage.FlashArray method*), 21
disable_directory_service() (*purestorage.FlashArray method*), 21
disable_network_interface() (*purestorage.FlashArray method*), 21
disable_pgroup_replication() (*purestorage.FlashArray method*), 21
disable_pgroup_snapshots() (*purestorage.FlashArray method*), 22
disable_phonehome() (*purestorage.FlashArray method*), 22
disable_remote_assist() (*purestorage.FlashArray method*), 22
disable_subnet() (*purestorage.FlashArray method*), 22
disconnect_array() (*purestorage.FlashArray method*), 22
disconnect_azure_offload() (*purestorage.FlashArray method*), 22
disconnect_hgroup() (*purestorage.FlashArray method*), 23
disconnect_host() (*purestorage.FlashArray method*), 23
disconnect_nfs_offload() (*purestorage.FlashArray method*), 23
disconnect_s3_offload() (*purestorage.FlashArray method*), 23

E

enable_alert_recipient() (*purestorage.FlashArray method*), 23
enable_app() (*purestorage.FlashArray method*), 23
enable_app_vnc() (*purestorage.FlashArray method*), 23
enable_console_lock() (*purestorage.FlashArray method*), 24
enable_directory_service() (*purestorage.FlashArray method*), 24
enable_network_interface() (*purestorage.FlashArray method*), 24
enable_pgroup_replication() (*purestorage.FlashArray method*), 24
enable_pgroup_snapshots() (*purestorage.FlashArray method*), 24
enable_phonehome() (*purestorage.FlashArray method*), 24
enable_remote_assist() (*purestorage.FlashArray method*), 24
enable_subnet() (*purestorage.FlashArray method*), 25
eradicate_pgroup() (*purestorage.FlashArray method*), 25
eradicate_pod() (*purestorage.FlashArray method*), 25
eradicate_vgroup() (*purestorage.FlashArray method*), 25
eradicate_volume() (*purestorage.FlashArray method*), 25
extend_volume() (*purestorage.FlashArray method*), 26

F

flag_message() (*purestorage.FlashArray method*), 26
FlashArray (*class in purestorage*), 9

G

get() (*purestorage.FlashArray method*), 26
get_admin() (*purestorage.FlashArray method*), 27
get_admin_lockout_info() (*purestorage.FlashArray method*), 27
get_alert_recipient() (*purestorage.FlashArray method*), 27
get_api_token() (*purestorage.FlashArray method*), 27
get_app() (*purestorage.FlashArray method*), 27
get_app_node() (*purestorage.FlashArray method*), 28
get_app_software() (*purestorage.FlashArray method*), 28
get_azure_offload() (*purestorage.FlashArray method*), 28
get_certificate() (*purestorage.FlashArray method*), 28
get_certificate_signing_request() (*purestorage.FlashArray method*), 28
get_console_lock_status() (*purestorage.FlashArray method*), 29
get_directory_service() (*purestorage.FlashArray method*), 29
get_dns() (*purestorage.FlashArray method*), 29
get_drive() (*purestorage.FlashArray method*), 29
get_eula() (*purestorage.FlashArray method*), 29

get_global_admin_attributes() (*purestorage.FlashArray method*), 29
get_hardware() (*purestorage.FlashArray method*), 29
get_hgroup() (*purestorage.FlashArray method*), 30
get_host() (*purestorage.FlashArray method*), 30
get_kmip() (*purestorage.FlashArray method*), 30
get_manual_phonehome_status() (*purestorage.FlashArray method*), 30
get_network_interface() (*purestorage.FlashArray method*), 30
get_nfs_offload() (*purestorage.FlashArray method*), 31
get_offload() (*purestorage.FlashArray method*), 31
get_pgroup() (*purestorage.FlashArray method*), 31
get_phonehome() (*purestorage.FlashArray method*), 31
get_pod() (*purestorage.FlashArray method*), 31
get_publickey() (*purestorage.FlashArray method*), 32
get_remote_assist_status() (*purestorage.FlashArray method*), 32
get_rest_version() (*purestorage.FlashArray method*), 32
get_s3_offload() (*purestorage.FlashArray method*), 32
get_smtp() (*purestorage.FlashArray method*), 32
get_snmp_engine_id() (*purestorage.FlashArray method*), 32
get_snmp_manager() (*purestorage.FlashArray method*), 33
get_subnet() (*purestorage.FlashArray method*), 33
get_vgroup() (*purestorage.FlashArray method*), 33
get_volume() (*purestorage.FlashArray method*), 33

I

install_app() (*purestorage.FlashArray method*), 33
invalidate_cookie() (*purestorage.FlashArray method*), 34

L

list_admins() (*purestorage.FlashArray method*), 34
list_alert_recipients() (*purestorage.FlashArray method*), 34
list_api_tokens() (*purestorage.FlashArray method*), 34
list_app_nodes() (*purestorage.FlashArray method*), 34
list_app_software() (*purestorage.FlashArray method*), 34
list_apps() (*purestorage.FlashArray method*), 35
list_array_connections() (*purestorage.FlashArray method*), 35
list_azure_offload() (*purestorage.FlashArray method*), 35
list_certificates() (*purestorage.FlashArray method*), 35
list_directory_service_roles() (*purestorage.FlashArray method*), 35
list_drives() (*purestorage.FlashArray method*), 36
list_hardware() (*purestorage.FlashArray method*), 36
list_hgroup_connections() (*purestorage.FlashArray method*), 36
list_hgroups() (*purestorage.FlashArray method*), 36
list_host_connections() (*purestorage.FlashArray method*), 36
list_hosts() (*purestorage.FlashArray method*), 36
list_kmip() (*purestorage.FlashArray method*), 36
list_locked_admins_lockout_info() (*purestorage.FlashArray method*), 37
list_maintenance_windows() (*purestorage.FlashArray method*), 37
list_messages() (*purestorage.FlashArray method*), 37
list_network_interfaces() (*purestorage.FlashArray method*), 37
list_nfs_offload() (*purestorage.FlashArray method*), 37
list_offload() (*purestorage.FlashArray method*), 37
list_pgroups() (*purestorage.FlashArray method*), 37
list_pods() (*purestorage.FlashArray method*), 38
list_ports() (*purestorage.FlashArray method*), 38
list_publickeys() (*purestorage.FlashArray method*), 38
list_s3_offload() (*purestorage.FlashArray method*), 38
list_snmp_managers() (*purestorage.FlashArray method*), 38
list_subnets() (*purestorage.FlashArray method*), 38
list_vgroups() (*purestorage.FlashArray method*), 39
list_volume_block_differences() (*purestorage.FlashArray method*), 39
list_volume_private_connections() (*purestorage.FlashArray method*), 39
list_volume_shared_connections() (*purestorage.FlashArray method*), 39
list_volumes() (*purestorage.FlashArray method*), 39

M

move_volume() (*purestorage.FlashArray method*), 40

P

page_through() (*purestorage.FlashArray static method*), 40
phonehome() (*purestorage.FlashArray method*), 40
PureError (*class in purestorage*), 52
PureHTTPError (*class in purestorage*), 52
purestorage (*module*), 9

R

recover_pgroup() (*purestorage.FlashArray method*), 41
recover_pod() (*purestorage.FlashArray method*), 41
recover_vgroup() (*purestorage.FlashArray method*), 41
recover_volume() (*purestorage.FlashArray method*), 41
refresh_admin() (*purestorage.FlashArray method*), 41
refresh_admins() (*purestorage.FlashArray method*), 42
remove_hgroup() (*purestorage.FlashArray method*), 42
remove_host() (*purestorage.FlashArray method*), 42
remove_pod() (*purestorage.FlashArray method*), 42
remove_volume() (*purestorage.FlashArray method*), 43
rename() (*purestorage.FlashArray method*), 43
rename_hgroup() (*purestorage.FlashArray method*), 43
rename_host() (*purestorage.FlashArray method*), 43
rename_pgroup() (*purestorage.FlashArray method*), 43
rename_pod() (*purestorage.FlashArray method*), 44
rename_snmp_manager() (*purestorage.FlashArray method*), 44
rename_subnet() (*purestorage.FlashArray method*), 44
rename_vgroup() (*purestorage.FlashArray method*), 44
rename_volume() (*purestorage.FlashArray method*), 45

S

schedule_maintenance_window() (*purestorage.FlashArray method*), 45
send_pgroup_snapshot() (*purestorage.FlashArray method*), 45
set() (*purestorage.FlashArray method*), 45
set_admin() (*purestorage.FlashArray method*), 46
set_certificate() (*purestorage.FlashArray method*), 46
set_directory_service() (*purestorage.FlashArray method*), 46
set_directory_service_roles() (*purestorage.FlashArray method*), 46
set_dns() (*purestorage.FlashArray method*), 47
set_eula() (*purestorage.FlashArray method*), 47
set_global_admin_attributes() (*purestorage.FlashArray method*), 47
set_hardware() (*purestorage.FlashArray method*), 47
set_hgroup() (*purestorage.FlashArray method*), 47
set_host() (*purestorage.FlashArray method*), 47
set_kmip() (*purestorage.FlashArray method*), 48
set_network_interface() (*purestorage.FlashArray method*), 48
set_password() (*purestorage.FlashArray method*), 48
set_pgroup() (*purestorage.FlashArray method*), 48
set_pod() (*purestorage.FlashArray method*), 49
set_publickey() (*purestorage.FlashArray method*), 49
set_smtp() (*purestorage.FlashArray method*), 49
set_snmp_manager() (*purestorage.FlashArray method*), 49
set_subnet() (*purestorage.FlashArray method*), 50
set_vgroup() (*purestorage.FlashArray method*), 50
set_volume() (*purestorage.FlashArray method*), 50

T

test_alert() (*purestorage.FlashArray method*), 50
test_alert_recipient() (*purestorage.FlashArray method*), 50
test_directory_service() (*purestorage.FlashArray method*), 50
test_kmip() (*purestorage.FlashArray method*), 51
test_snmp_manager() (*purestorage.FlashArray method*), 51
throttle_array_connection() (*purestorage.FlashArray method*), 51
truncate_volume() (*purestorage.FlashArray method*), 51

U

uninstall_app() (*purestorage.FlashArray method*), 51
unlock_admin() (*purestorage.FlashArray method*), 52
unschedule_maintenance_window() (*purestorage.FlashArray method*), 52