Actifio python Client Library Documentation

Release v.0.9.0

Kosala Atapattu (kosala.atapattu@actifio.com)

Oct 06, 2020

Contents:

CHAPTER 1

Actifio Class

This class instantiate a Actifio appliance object. The implimented methods correspond to regular appliance operations.

class Actifio.**Actifio**(*appliance*, *username=''*, *password=''*, *token=''*, *cert_validation=False*, *verbose=True*)

Actifio instance:

Attributes:

appliance IP or FQDN of the appliance

username Username to login to the appliance

password Password

cert_validation Certificate validation for SSL connections. Defaults to false.

run_uds_command(*cmdType*, *cmdUDS*, *cmdArgs={}*)

Wrapper function to convert CLI commands to the rest API.

Args:

cmdType info / task

cmdUDS Command to use (eg. lsuser, lshost, mkapplication... etc.)

cmdArgs Dictionary with arguments to the command

Returns:

Returns a dictionary of API response.

Example:

vmdiscovery -discovercluster -host 1234

{ 'discovercluster': None, 'host': 1234 }

lsapplication -filtervalues "appname=mydb&hostname=myhost"

{ 'filtervalues': { 'appname': 'mydb', 'hostname': 'myhost' } }

lshost 123

{ 'argument': 123 }

Note: RESTfulAPI_*.pdf would be good referecne point for the __SIMILARITY__ and __PATTERN__ of the cmdArgs.

---

run_sarg_command(*cmdSARG*, *cmdArgs={}*)

Wrapper function to convert CLI commands to the rest API.

Args:

cmdSARG Command to use (eg. reportsnaps, reportapps... etc.)

cmdArgs Dictionary with arguments to the command

Return:

return a dictionary of the SARG command, mapping to the same JSON response from the API.

Example:

reportapps -a 1234 -x

self.run_sarg_command("reportapps", { 'a': 1234, 'x': None })

get_hosts(***kwargs*)

This method query for the hosts registered in Actifio appliance. You can specify a combination of following filter attributes.

Attributes:

alternateip Specifies the alternate IP address of the host. Multiple alternate can be specified in a comma-delimited list. To remove the alternate IP address, use an empty field with double quotes.

description Description of the host.

diskpref Specifies preference (BLOCK or NFS) for presenting the staging disk. Default value is BLOCK.

friendlypath Friendly path for the host.

hasagent Tells us whether the host has an agent. 0= none, 1= yes <– this is true/false

hostname Host name

hosttype Host type, for example generic, hmc, hpux, hyperv, isilon, netapp svm, netapp 7 mode, openvms, tpgs, or vcenter.

isclusterhost Host is a clustered host.

ipaddress IP address of the host.

isesxhost Whether the host is an esx server.

isvcenterhost Whether the host is a management server, such as a vCenter.

isvm Whether the host is a VM.

originalhostid Identifies original host id for shadow host.

osrelease Operating system release.

ostype Operating system type.

osversion Operating system version.

sourcecluster Identifies the original cluster ID for shadow host

svcname Specifies the SVC host name, which limits to 15 characters, first character cannot be a number, and no space, or '.' is allowed.

uniquename Unique name for the host.

vcenterhostid The vCenter host ID.

Returns:

Returns the ActHostCollection object with a list of Host entries to satisfy the filter criteria.

get_applications(***kwargs*)

This method query for the registered applications within a Actifio appliance. You can specify a combination of following filter attributes.

Attributes:

appname Application name

apptype Application type

appversion Whatever we glean during discovery, and it is not always available.

auxinfo For internal use, not likely to be useful.

description Description of the application.

friendlytype Friendly type for the application

hostid Host id.

hostname Host name.

id Application id.

ignore Allows the user to ignore the application (when set), so application will not show up in the UI.

isclustered Specifies if the application is part of a cluster.

networkip The network IP of the application

networkname The network name of the application.

originalappid Original application id.

pathname The path name of the application

protectable None means you cannot protect it, fully means you can, partial means there is limited support.

sourcecluster Identifies the original cluster ID for shadow host (when we create a shadow application or shadow host, this tells us where it originates from).

Returns:

Instance of ActAppCollection object with a collection of all the application matching the filter criteria.

get_images(***kwargs*)

Queries Actifio appliance with matching backups images as specified by the filter criteria. if no filter criteria specified will return all the backup images.

Args:

appid Application object ID.

appname Application name

apptype Application type

backupdate Start date [usage: 'backupdate since 24 hours' for backups started since last 24 hours,'backupdate before 7 days' for backups started older than 7 days]

backupname Image name.

characteristic Charchteristic for of backup type (in addition to jobclass [PRIMARY | MOUNT | UNMOUNT | VDISK | CLONE]

consistencydate consistency date of the backup

consistency-mode Consistency mode of image (for example, application consistent or crash consistent).

expiration Date and time when this should expire. Images with an enforced retention (including remote images) cannot be expired before they reach the immutability date.

hostid Application ID of the host where the backup image ??? <– host ID of the capture job host

hostname Name of the host where the backup image is???? <– host name of the capture job host

jobclass Type of jobs [snapshot | dedup | dedupasync | clone | liveclone | syncback]

label label of the backup that user specified.

mappedhost ID of the host to which backup image is mapped.

mountedhost ID of host where backup image is mounted.

policyname Name of the policy on which this object is created.

prepdate Date when LiveClone image is created.

slpname Profile name used while creating this image.

sltname SLA template name used while creating this image.

sourceimage obsolete

sourceuds Cluster ID of the source cluster

targetuds Cluster ID of the target cluster

virtualsize Application size

Returns:

Return the backups image collection in ActImgCollection object.

get_jobs(***kwargs*)

This method query for the jobs, running and archived. The following filter arguments can be used to refine the output. Returns ActJobCollection object.

Args:

appid

appname

component

enddate

errorcode

expirationdate

hostname

isscheduled [true | false]

jobclass

jobname

jobtag

parentid

policyname

priority

progress

queuedate

relativesize

retrycount

sltname

startdate

status [running | queued | paused | interrupted | stalled]

sourceid

virtualsize

Returns:

ActJobCollection object with a collection of jobs as per the selection criteria.

get_image_bytime(*application*, *restoretime*, *strict_policy=False*, *job_class='snapshot'*)

This method returns a ActImage object with a single image to the specified restore time.

Args:

application should be the application in the form of ActApplication object.

strict_policy [True | False] If set to true, the image will be selected from log recovery range, with the closest image to replay the logs on.

restoretime can be datetime obect or string with the format [YYYY-MM-DD HH:mm:ss] job_class: Defaults to snapshot. Should be string type, to any supported image jobclass.

Returns:

ActImage object to the specified *restoretime*. If strict_policy is set to *True*, the image will selected to the closest *restoretime*, where redo logs can be played up to the *restoretime*. If *strict_policy* is set to *False*, then the closest image to the restore time will be selected. When *strict_policy* is *False*, the recovery image consistencytime could be ahead of the *restoretime*, however *strict_policy* is True would ensure image consistency time is always lower than the *restoretime*.

clone_database(*source_hostname=''*, *source_appname=''*, *source_application=None*, *target_hostname=''*, *target_host=None*, *restoretime=''*, *strict_policy=True*, *pre_script=''*, *post_script=''*, *nowait=True*, ***kwargs*)

This method creates a virtual clone of Oracle or SQL server database.

Agrs:

source_hostname Hostname of the source host where the database was captured from

source_appname source application name, or the database name

target_hostname target host where the virtual clone need to be created on

Miscelaneous Parameters

restoretime Point in time the database needs to be recovered to.

strict_policy Defaults to True, If set to True (only for applications with log database backups), :databases will be cloned to the time specified.

nowait defaults to True, if True, this method will be non-blocking mode.

Oracle Related Parameters

oracle_home (required) ORACLE_HOME

oracle_db_name (required) SID of the target clone

oracle_user (optional) Defaults to "oracle".

oracle_tns_admin (optional) TNS admin path, defaults to $ORACLE_HOME/network/admin.

oracle_db_mem (optional) Total Memory Target for the database, defaults to 512MB.

oracle_sga_pct (optional) Memory Percentage to allocate for SGA

oracle_redo_size (optional) Redo Log size in MB, defaults to 500

oracle_shared_pool (optional) Oracle Shared Pool size

oracle_db_cache_size (optional) Oracle DB Cache size

oracle_recover_dest_size (optional) Oracle Parameter db_recover_dest_size. Defaults to 5000

oracle_diagnostic_dest (optional) Oracle Diagnostic Destination

oracle_nprocs (optional) Num of Max processes

oracle_open_cursors (optional) Number of open_cursors. defaults to 1000.

oracle_char_set (optional) Characterset. Defaults to 'AL32UTF8'

oracle_tns_ip (optional) TNS IP Address

oracle_tns_port (optional) TNS Port

oracle_tns_domain (optional) TNS Domain

oracle_no_nid (optional) Do not change the DBID of the new clone. Will maintain same DBID as the source. Defaults to FALSE

oracle_no_tns_update (optional) Do not update TNS records. Defaults to FALSE

oracle_restore_recov (optional) Recover the oracle database. Defaults to TRUE

oracle_no_rac (optional) Treat as Oracle RAC. Defaults to TRUE

SQLServer Related

sql_instance_name (required) Target SQL Server instance name

sql_recover_userlogins (optional) Recover user logins of the database. Defaults to FALSE

sql_username (optional) Username for database provisioning

sql_password (optional) Password for the specified user

sql_recover_db (optional) Recover database after mount

SQLServer DB Application

sql_db_name (reuired) Database name at the target instance. (Only required if the source application is database or single database mount from instance.)

SQLServer Instance (Single DB)

sql_source_dbnames (required) Source database names if the source application is SQL instance. Use ',' as delimiter for multiple databases. (Only required if the source application is SQL server instance.)

sql_db_name (reuired) Database name at the target instance. (Only required if the source application is database or single database mount from instance.)

sql_dbname_prefix (optional) Prefix of database name for multiple database mount

sql_dbname_suffix (optional) Suffix of database name for multiple database mount

SQLServer Instance (Multiple DBs)

sql_source_dbnames (required) Source database names if the source application is SQL instance. Use ',' as delimiter for multiple databases. (Only required if the source application is SQL server instance.)

sql_cg_name (required) Consistency group name. (Only required if the source application is SQL Server instance and mount multiple databases at a time.)

sql_dbname_prefix (optional) Prefix of database name for multiple database mount

sql_dbname_suffix (optional) Suffix of database name for multiple database mount

Returns:

This method returns a tuple of (ActJob,ActImage), respectively the resulting Job and Image.

simple_mount(*source_application=None*, *target_host=None*, *mount_image=None*, *restore-time=''*, *strict_policy=False*, *pre_script=''*, *post_script=''*, *nowait=True*, *job_class='snapshot'*, *mount_mode='nfs'*, *label='Python Library'*, ***kwargs*)

This method mounts a simple mount operation, for a application type. This mount will not create a virtual clone (if you need to create a virtual clone look into clone_database() instead).

Args:

If not mount_image is None

mount_image (required) ActImage object refering to mount image

ElseIf not source_application is None

source_hostname (required) hostname where the server was backed up from.

source_appname (required) name of the application

Else

source_application (required) ActApplication object refereing to source application

If not target-host is None

target_hostname (required) hostname of the target host

Else

target_host (required) ActHost object refering to the target host

restoretime (optional) recovery time of the mount image, depending on the strict_policy, the closest image will be selected.

strict_policy (optional) Boolean, defaults to False, if True, application is treated as transaction log capable and image is selected to a level where recoverable to restore-time. Else closest image to the time will be selected

pre_script (optional) Pre Script for the mount operation

post-script (optional) Post Script for the mount operation

nowait (optional) defaults to True, mount job will not wait till the completion, if False, this method will be blocking until the job completion.

job_class (optional) Defaults to "snapshot", valid jobclasses are, [snapshot | dedup | dedupasync | OnVault]. job_class is applicable only when the restoretime is specified.

mount_mode (optional) Takes the value, physical (pRDM), independentvirtual (vRDM), or nfs (requires 9.0)

maptoallesxhosts (optional) Defaults to False. Map to all the ESXi hosts in the cluster.

Restore Options

vm_name (optional) For a new vm, VM name.

vm_poweron (optional) Defaults to False. Poweron the VM upon mount.

Any of the restoreoptions as listed in the **udsinfo lsrestoreoptions** can be specified as key=value command arguments.

Note: For more information on the restore options refer to the Appendix F on the RESTfulAPI.pdf.

Returns:

This method returns a tuple of (ActJob,ActImage), respectively the resulting Job and Image.

unmount_image(*image=None*, *delete=True*, *nowait=True*, *pre_script=''*, *post_script=''*)

Unmount a mounted image.

Args:

image ActImage object of the mounted image.

delete Delete the image after unmount

nowait Don't wait for the job completion.

pre_script (optional) Pre Script for the mount operation

post-script (optional) Post Script for the mount operation

Return:

Returns ActJob image with the resulting job

failover_database(*source_application=None*, *target_host=None*, *pre_script=''*, *post_script=''*, *mount_mode='physical'*, *label='Python Library'*, ***kwargs*)

This method would failover SQL database application from a

CHAPTER 2

ActImageCollection Class

Iterable class of *ActImage* collections. Returned by :ref: *Actifio.get_images()* method.

2.1 ActImage Class

ActImage object represent a Actifio backup image. This is a returned as part of a ActImageCollection through interable protocol, or through and index.

Method Actifio.get_image_bytime() would return the `ActImage`, instead of the `ActImageCollection`.

class `Actifio.`**ActImage**(*applaince*, *imgdata*)

details()
Fetch further details of the backups image.

Args:

None

Returns:

None

restoreoptions(*action*, *targethost*)
Retrieve restore options for a ActImage for mount / clone / restore operations

Args:

action (required) operation [mount, restore , clone]

targethost (required) Host ID of the targethost, ActHost type

Returns:

Returns a ActRestoreoptionCollection object with the relavant restore options for this image, for the specified action.

provisioningoptions()
Retrieve restore xoptions for a ActImage for mount / clone / restore operations

Args:

None:

Returns:

Returns a ActProvisiningptionCollection object with the relavant provisioning options for this appclass.

2.1.1 Example:

```
images = appliance.get_images(appname="mydb", jobclass="OnVault")

firstimage = images[0]

print(type(firstimage))

>>> Actifio.ActSupportClasses.ActImage
```

class Actifio.**ActImageCollection**(*appliance*, *lsbackupdata*)

2.2 Example:

```
images = appliance.get_images(appname="mydb", jobclass="OnVault")

for image in images:
  print(image)

# or to list all backups taken last 24 hours.

from datetime import datetime

yesterday = datetime.today() - datetime.timedelta(day=1)

for image in images:
  consistencydate = datetime.strptime(image.consistencydate[:-4], "%Y-%m-%d %H:%M:%S")
  if consistencydate > yesterday:
    print(image)
```

CHAPTER 3

ActJobsCollection Class

ActJobsCollection defines a iterable collection of Actifio jobs. ActJobCollection class objects are returned fron Actifio.get_jobs() method.

3.1 ActJob Class

ActJob object represent a Actifio backup image. This is returned as part of a ActJobsCollection through interable protocol, or through and index.

Method `Actifio.clone_database()` and `Actifio.simple_mount()` would return the `ActJob` and :doc: *ActImage </actimage>* in a tuple

class `Actifio.`**ActJob**(*applaince*, *jobdata*)

refresh()
Method to refresh the job details.

Args:

None

Returns:

None

3.1.1 Example:

```
jobs = appliance.get_jobs(hostname="myVM", jobclass="dedup", status="running")

if len(jobs) > 0:
  firstjob = jobs[0]
```

(continues on next page)

(continued from previous page)

```
# or from a mount operation

job, image = appliance.simple_mount(source_application=apps[0], target_host=hosts[0])

while job.status == "running":
  print("Still Running")
print("Phew, it's done.")
```

class Actifio.**ActJobsCollection**(*appliance*, *lsjobsalldata*)

Iterable collection of jobs.

refresh()

Method to refresh the job details, for each job.

3.2 Example:

```
jobs = appliance.get_images(appname="mydb", jobclass="snapshot")

for job in jobs:
  print(image)

# or refine further to find out running jobs

jobs = appliance.get_images(appname="mydb", jobclass="snapshot", status="running")

for job in jobs:
  print(image)
```

CHAPTER 4

ActAppCollection Class

ActAppsCollection represents a collection of Actifio applications generated using :doc: *Actifio.get_applications() </actifio>* method. ActAppsCollection is a iterable collection.

4.1 ActApplication Class

ActImage object represent a Actifio backup image. This is a returned as part of a ActImageCollection through interable protocol, or through and index.

Method Actifio.get_image_bytime() would return the `ActImage`, instead of the `ActImageCollection`.

class `Actifio.`**ActApplication**(*applaince*, *appdata*)

provisioningoptions()

Retrieve restore options for a ActImage for mount / clone / restore operations

Args:

None:

Returns:

Returns a ActProvisiningptionCollection object with the relavant provisioning options for this appclass.

4.1.1 Example:

```
apps = appliance.get_applications(appname="mydb")

if len(apps) > 0:
  myapp = apps[0]
```

(continues on next page)

(continued from previous page)

```
print(type(myapp))

>>> Actifio.ActSupportClasses.ActApplication
```

class Actifio.**ActAppCollection**(*appliance*, *lsapplicationdata*)

4.2 Example:

```
# to get all SQL Server..

sql_apps = appliance.get_applications(friendlytype="SQLServer")

# or to get both SQL Server and SQL Instance types

all_sql_apps = appliance.get_applications(friendlytype="SQLServer*")

for app in sql_apps:
  print(app)
```

CHAPTER 5

ActHostCollection Class

ActHostCollection represents a collection of Actifio hosts generated using :doc: *Actifio.get_hosts() </actifio>* method. ActHostCollection is a iterable collection.

5.1 ActHost Class

ActHost object represent a Actifio host entry. This is returned as part of a ActHostsCollection through interable protocol, or through and index.

Method `Actifio.clone_database()` and `Actifio.simple_mount()` would take `ActHost` as an argument.

class Actifio.**ActHost**(*applaince*, *hostdata*)

5.1.1 Example:

```
hosts = appliance.get_hosts(hostname="myVM", isvm="true")

for host in hosts:
  print(host)

print(type(host))

>>> Actifio.ActSupportClasses.ActHost
```

class Actifio.**ActHostCollection**(*appliance*, *lshostdata*)

5.2 Example:

```
hosts = appliance.get_hosts(hostname="myVM", isvm="true")

print(hosts)

>>> Collection of 1 hosts.

print(len(hosts))

>>> 1
```

CHAPTER 6

How to install?

To install the Actifio module, from a command line interaface:

```
→ pip install Actifio
```

This will ensure all the dependencies are managed and installed with the Actifio module installation. Once installed:

```
→ pip show Actifio
Name: Actifio
Version: 0.9.0
Summary: Actifio Restful API wrapper for Python.
Home-page: https://github.com/Actifio/actifio-python-package
Author: Kosala Atapattu
Author-email: kosala.atapattu@actifio.com
License: MIT
Location: /usr/local/lib/python2.7/site-packages
Requires: urllib3
Required-by:
```

CHAPTER 7

Geting started!!!

By design philosophy of this library is to make sure that the user experience is consistent to the actual product. Start Python and then you can import the module by:

```
from Actifio import Actifio
```

The library supports two modes of authentication, either using username or password, or using token generated locally.

- Username and Password

With the same information you use to login to the appliance, you can create a appliance object.

```
appliance = Actifio("myappliance", "my_scripting_user", "super_secret")
```

Or

- With Token

You can generate a token using the script in ‘’‘bin/’‘’ folder, or using the command. To generate a token:

```
$ bin/actgentoken
Username: demo
Password:
Confirm password:



================Token====================

b'eyAidXNlcm5hbWUiOiAiZGVtbyIsICJwYXNzd29yZCI6ICJkZW1vIiB9\n'

=========================================
```

Once the token is generated, appliance object can be instantiated as following:

```
appliance = Actifio("myappliance", token=b
↪'eyAidXNlcm5hbWUiOiAiZGVtbyIsICJwYXNzd29yZCI6ICJkZW1vIiB9\n')
```

Once the appliance object is instatiated, we can perform the operations we perform on the applaince.

List all the hosts, for example:

```
hosts = appliance.get_hosts(hostname="my_host")

# and to see the top host in my List

host = hosts[0]

# or refine further

hosts = appliance.get_hosts(hostname="my_host", isvm="true")
```

Or find an application:

```
apps = appliance.get_applications(appname="mydb")

# and to see the top application in my List

if len(apps) > 0:
  app = apps[0]

# or refine further, and get my Oracle database

hosts = appliance.get_applications(appname="mydb", friendlytype="Oracle")
```

Once I have that, then I can perform the actions I usually perform... create a virtual clone of a DB `appliance.clone_database()` or create a instant mount `appliance.clone_database()`. Checkout the examples section for more details.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index