
Cloudforms API Python Client Documentation

Release 0.1a1

Joshua Cornutt

Sep 21, 2017

Contents

| | | |
|----------|--|-----------|
| 1 | API Documentation | 3 |
| 1.1 | Cloudforms.ProviderManager | 3 |
| 1.2 | Cloudforms.ProvisionRequestManager | 6 |
| 1.3 | Cloudforms.TagManager | 7 |
| 1.4 | Cloudforms.TaskManager | 9 |
| 1.5 | Cloudforms.VSManager | 10 |
| 2 | Examples | 13 |
| 3 | External Links | 15 |
| | Python Module Index | 17 |

This is the documentation for the community-supported Red Hat Cloudforms (ManageIQ) API v2 Python client.

Contents:

Available managers:

Cloudforms.ProviderManager

Provider Manager

license MIT, see LICENSE for more details.

class `Cloudforms.managers.provider.ProviderManager` (*client*)
Manages Providers.

Parameters `client` (*Cloudforms.API.Client*) – an API client instance

Example:

```
import Cloudforms
client = Cloudforms.Client()
provider_mgr = Cloudforms.ProviderManager(client)
```

create (*params=None*)

Creates a new provider on the account (pass-through params)

Parameters `params` (*dict*) – Additional POST request data

Returns Provider dictionary object

Example:

```
# Creates a new provider using pass-through params
provider = provider_mgr.create({
    'type': 'EmsRedhat',
    'name': 'rhevm101',
    'hostname': 'rhevm101',
    'ipaddress': '127.0.0.1',
    'credentials': {
```

```

        'userid': 'admin',
        'password': 'smartvm'
    }
})
# Refresh the provider
provider_mgr.refresh(provider['id'])

```

create_amazon (*name, region, access_key, secret_key, params=None*)
 Creates a new Amazon (AWS) provider on the account

Parameters

- **string** (*secret_key*) – Display name of the provider
- **string** – AWS region (ex. us-east-1)
- **string** – AWS API Access Key ID
- **string** – AWS API Access Secret Key
- **params** (*dict*) – Additional POST request data

Returns Provider dictionary object

Example:

```

# Creates a new Amazon (AWS) provider
provider = provider_mgr.create_amazon(
    name='MyAWSProvider',
    region='us-east-1',
    access_key='MY4CC3SSK3Y',
    secret_key='My$Freak1shly/L0ng=S3cr3t&K3y'
)
# Refresh the provider
provider_mgr.refresh(provider['id'])

```

delete (*_id, params=None*)
 Sends a request to delete a provider

Parameters

- **_id** (*string*) – Specifies which provider the request is for
- **params** (*dict*) – Additional POST request data

Returns Task request dictionary (see TaskManager)

Example:

```

# Gets a list of all providers
for provider in provider_mgr.list():
    # Send requests to delete all providers
    task = provider_mgr.delete(provider['id'])
    # Wait for the request to be processed
    task_mgr.wait_for_task(task.get('task_id'))

```

get (*_id, params=None*)
 Retrieve details about a provider on the account

Parameters

- **_id** (*string*) – Specifies which provider the request is for

- **params** (*dict*) – response-level options (attributes, limit, etc.)

Returns Dictionary representing the matching provider

Example:

```
# Gets a list of all providers (returns IDs only)
providers = provider_mgr.list({'attributes': 'id'})
for provider in providers:
    provider_details = provider_mgr.get(provider['id'])
```

list (*params=None*)

Retrieve a list of all providers on the account

Parameters **params** (*dict*) – response-level options (attributes, limit, etc.)

Returns List of dictionaries representing the matching providers

Example:

```
# Gets a list of all providers (returns IDs only)
providers = provider_mgr.list({'attributes': 'id'})
```

perform_action (*_id, action, params=None*)

Sends a request to perform an action on a provider

Parameters

- **_id** (*string*) – Specifies which provider the request is for
- **action** (*string*) – The action to request (delete, refresh, etc.)
- **params** (*dict*) – Additional POST request data

Returns Task request, or Provider, dictionary object

Example:

```
# Gets a list of all providers
for provider in provider_mgr.list():
    # Send requests to refresh all providers
    provider_mgr.perform_action(provider['id'], 'refresh')
```

refresh (*_id, params=None*)

Sends a request to refresh a provider

Parameters

- **_id** (*string*) – Specifies which provider the request is for
- **params** (*dict*) – Additional POST request data

Returns JSON object with a ‘success’ key

Example:

```
# Gets a list of all providers
for provider in provider_mgr.list():
    # Send requests to refresh all providers
    res = provider_mgr.refresh(provider['id'])
    if not res or not res.get('success'):
        raise RuntimeError('An error occurred')
```

update (*_id*, *params=None*)

Sends a request to update a provider

Parameters

- **_id** (*string*) – Specifies which provider the request is for
- **params** (*dict*) – Additional POST request data

Returns Provider dictionary object (with updates applied)

Example:

```
# Gets a list of all providers
for provider in provider_mgr.list():
    # Send requests to update all providers
    provider_mgr.update(
        provider['id'],
        params={
            'credentials': [{
                'userid': 'metrics_userid',
                'password': 'metrics_password',
                'auth_type': 'metrics'
            }]
        }
    )
```

Cloudforms.ProvisionRequestManager

Provision Request Manager

license MIT, see LICENSE for more details.

class `Cloudforms.managers.provision_request.ProvisionRequestManager` (*client*)
 Manages Provision Requests.

Parameters **client** (*Cloudforms.API.Client*) – an API client instance

Example:

```
import Cloudforms
client = Cloudforms.Client()
preq_mgr = Cloudforms.ProvisionRequestManager(client)
```

create (*params=None*)

Creates a new provision request on the account

Parameters **params** (*dict*) – Additional POST request data

Returns ProvisionRequest dictionary object

get (*_id*, *params=None*)

Retrieve details about a provision request on the account

Parameters

- **_id** (*string*) – Specifies which provision request the request is for
- **params** (*dict*) – response-level options (attributes, limit, etc.)

Returns Dictionary representing the matching provision request

Example:

```
# Gets a list of all provision requests (returns IDs only)
preqs = preq_mgr.list({'attributes': 'id'})
for preq in preqs:
    preq_details = preq_mgr.get(preq['id'])
```

list (*params=None*)

Retrieve a list of all provision requests on the account

Parameters **params** (*dict*) – response-level options (attributes, limit, etc.)

Returns List of dictionaries representing the matching provision requests

Example:

```
# Gets a list of all provision requests (returns IDs only)
preqs = preq_mgr.list({'attributes': 'id'})
```

perform_action (*_id, action, params=None*)

Sends a request to perform an action on a provision request

Parameters

- **_id** (*string*) – Specifies which provision request the request is for
- **action** (*string*) – The action to request (delete, refresh, etc.)
- **params** (*dict*) – Additional POST request data

Returns ProvisionRequest dictionary object

wait (*_id, timeout=30, request_state='finished', params=None*)

Waits for a provision request to reach a certain request_state

Parameters

- **request_state** (*string*) – wait until the provision request reaches this request_state (case insensitive)
- **timeout** (*integer*) – operation timeout (in seconds)
- **params** (*dict*) – response-level options (attributes, limit, etc.)

Returns bool **True** on success, **False** on error or timeout

Cloudforms.TagManager

Tag Manager

license MIT, see LICENSE for more details.

class `Cloudforms.managers.tag.ServiceTagManager` (*client, svc*)

Manages Tags for Services.

Parameters

- **client** (*Cloudforms.API.Client*) – an API client instance
- **svc** (*string*) – a service name to bind to

assign (*_id, names*)

Assigns one or more tags to a service

Parameters

- **_id** (*string*) – Specifies which service item the request is for
- **names** (*list*) – Names of tags to assign (['/my/tag', '/a/tag'])

Example:

```
# Add the tag /environment/prod to all providers
for prov in prov_mgr.list_providers():
    prov_mgr.tags.assign(prov.get('id'), [
        '/environment/prod'
    ])
```

unassign (*_id, names*)

Un-assigns one or more tags to a service

Parameters

- **_id** (*string*) – Specifies which service item the request is for
- **names** (*list*) – Names of tags to un-assign (['/my/tag', '/a/tag'])

Example:

```
# Removes the tag /environment/prod from all providers
for prov in prov_mgr.list_providers():
    prov_mgr.tags.unassign(prov.get('id'), [
        '/environment/prod'
    ])
```

class `Cloudforms.managers.tag.TagManager` (*client*)

Manages Tags.

Parameters **client** (*Cloudforms.API.Client*) – an API client instance

Example:

```
import Cloudforms
client = Cloudforms.Client()
tag_mgr = Cloudforms.TagManager(client)
```

get (*_id, params=None*)

Retrieve details about a tag on the account

Parameters

- **_id** (*string*) – Specifies which tag the request is for
- **params** (*dict*) – response-level options (attributes, limit, etc.)

Returns Dictionary representing the matching tag

Example:

```
# Gets a list of all tags (returns IDs only)
tags = tag_mgr.list({'attributes': 'id'})
for tag in tags:
    tag_details = tag_mgr.get(tag['id'])
```

list (*params=None*)

Retrieve a list of all tags on the account

Parameters **params** (*dict*) – response-level options (attributes, limit, etc.)

Returns List of dictionaries representing the matching tags

Example:

```
# Gets a list of all tags (returns IDs only)
tags = tag_mgr.list({'attributes': 'id'})
```

Cloudforms.TaskManager

Task Manager

license MIT, see LICENSE for more details.

class `Cloudforms.managers.task.TaskManager` (*client*)
Manages Tasks.

Parameters `client` (*Cloudforms.API.Client*) – an API client instance

Example:

```
import Cloudforms
client = Cloudforms.Client()
task_mgr = Cloudforms.TaskManager(client)
```

get (*_id*, *params=None*)

Retrieve details about a task on the account

Parameters

- **_id** (*string*) – Specifies which task the request is for
- **params** (*dict*) – response-level options (attributes, limit, etc.)

Returns Dictionary representing the matching task

Example:

```
# Gets a list of all tasks (returns IDs only)
tasks = task_mgr.list({'attributes': 'id'})
for task in tasks:
    task_details = task_mgr.get(task['id'])
```

list (*params=None*)

Retrieve a list of all tasks on the account

Parameters `params` (*dict*) – response-level options (attributes, limit, etc.)

Returns List of dictionaries representing the matching tasks

Example:

```
# Gets a list of all tasks (returns IDs only)
tasks = task_mgr.list({'attributes': 'id'})
```

wait (*_id*, *timeout=30*, *state='finished'*, *params=None*)

Waits for a task to reach a certain state

Parameters

- **state** (*string*) – wait until the task reaches this state (case insensitive)

- **timeout** (*integer*) – operation timeout (in seconds)
- **params** (*dict*) – response-level options (attributes, limit, etc.)

Returns **bool** **True** on success, **False** on error or timeout

Example:

```
# Gets a list of all virtual servers
vms = vs_mgr.list()
for vm in vms:
    # Send a request to stop the virtual server
    task = vs_mgr.stop(vms.get('id'))
    # Wait for the task to finish and collect the result
    task_succeeded = task_mgr.wait(task.get('task_id'))
```

Cloudforms.VSManager

VS Manager (abstracts virtual machines and cloud instances)

license MIT, see LICENSE for more details.

class `Cloudforms.managers.vs.VSManager` (*client*)
 Manages Virtual Servers.

Parameters **client** (*Cloudforms.API.Client*) – an API client instance

Example:

```
import Cloudforms
client = Cloudforms.Client()
vs_mgr = Cloudforms.VSManager(client)
```

delete (*_id*, *params=None*)
 Sends a request to delete a virtual server

Parameters

- **_id** (*string*) – Specifies which virtual server the request is for
- **params** (*dict*) – Additional POST request data

Returns Task request dictionary (see TaskManager)

Example:

```
# Gets a list of all virtual server instances
for vsi in vs_mgr.list():
    # Send requests to delete all virtual server instances
    vs_mgr.delete(vsi['id'])
```

get (*_id*, *params=None*)
 Retrieve details about a virtual server on the account

Parameters

- **_id** (*string*) – Specifies which virtual server the request is for
- **params** (*dict*) – response-level options (attributes, limit, etc.)

Returns Dictionary representing the matching virtual server

Example:

```
# Gets a list of all virtual server instances (returns IDs only)
instances = vs_mgr.list({'attributes': 'id'})
for instance in instances:
    vs_details = vs_mgr.get(instance['id'])
```

list (*params=None*)

Retrieve a list of all virtual servers on the account

Parameters *params* (*dict*) – response-level options (attributes, limit, etc.)

Returns List of dictionaries representing the matching virtual server

Example:

```
# Gets a list of all virtual server instances (returns IDs only)
instances = vs_mgr.list({'attributes': 'id'})
```

perform_action (*_id, action, params=None*)

Sends a request to perform an action on a virtual server

Parameters

- **_id** (*string*) – Specifies which virtual server the request is for
- **action** (*string*) – The action to request (start, stop, suspend)
- **params** (*dict*) – Additional POST request data

Returns Task request dictionary (see TaskManager)

Example:

```
# Gets a list of all virtual server instances
for vsi in vs_mgr.list():
    # Send requests to start all virtual server instances
    vs_mgr.perform_action(vsi['id'], 'start')
```

start (*_id, params=None*)

Sends a request to start a virtual server

Parameters

- **_id** (*string*) – Specifies which virtual server the request is for
- **params** (*dict*) – Additional POST request data

Returns Task request dictionary (see TaskManager)

Example:

```
# Gets a list of all virtual server instances
for vsi in vs_mgr.list():
    # Send requests to start all virtual server instances
    vs_mgr.start(vsi['id'])
```

stop (*_id, params=None*)

Sends a request to stop a virtual server

Parameters

- **_id** (*string*) – Specifies which virtual server the request is for

- **params** (*dict*) – Additional POST request data

Returns Task request dictionary (see TaskManager)

Example:

```
# Gets a list of all virtual server instances
for vsi in vs_mgr.list():
    # Send requests to stop all virtual server instances
    vs_mgr.stop(vsi['id'])
```

suspend (*_id*, *params=None*)

Sends a request to suspend a virtual server

Parameters

- **_id** (*string*) – Specifies which virtual server the request is for
- **params** (*dict*) – Additional POST request data

Returns Task request dictionary (see TaskManager)

Example:

```
# Gets a list of all virtual server instances
for vsi in vs_mgr.list():
    # Send requests to suspend all virtual server instances
    vs_mgr.suspend(vsi['id'])
```


List all virtual servers for the account:

```
from Cloudforms import (
    Client,
    VSManager
)

client = Client(
    username='admin',
    password='smartvm',
    host='127.0.0.1'
)

vs_mgr = VSManager(client)
instances = vs_mgr.list()
for instance in instances:
    print 'Server #%s: %s (%s)' % (
        instance.get('id'),
        instance.get('name'),
        instance.get('raw_power_state')
    )
```


CHAPTER 3

External Links

C

`Cloudforms.managers.provider`, 3
`Cloudforms.managers.provision_request`,
6
`Cloudforms.managers.tag`, 7
`Cloudforms.managers.task`, 9
`Cloudforms.managers.vs`, 10

A

assign() (Cloudforms.managers.tag.ServiceTagManager method), 7

C

Cloudforms.managers.provider (module), 3

Cloudforms.managers.provision_request (module), 6

Cloudforms.managers.tag (module), 7

Cloudforms.managers.task (module), 9

Cloudforms.managers.vs (module), 10

create() (Cloudforms.managers.provider.ProviderManager method), 3

create() (Cloudforms.managers.provision_request.ProvisionRequestManager method), 6

create_amazon() (Cloudforms.managers.provider.ProviderManager method), 4

D

delete() (Cloudforms.managers.provider.ProviderManager method), 4

delete() (Cloudforms.managers.vs.VSManager method), 10

G

get() (Cloudforms.managers.provider.ProviderManager method), 4

get() (Cloudforms.managers.provision_request.ProvisionRequestManager method), 6

get() (Cloudforms.managers.tag.TagManager method), 8

get() (Cloudforms.managers.task.TaskManager method), 9

get() (Cloudforms.managers.vs.VSManager method), 10

L

list() (Cloudforms.managers.provider.ProviderManager method), 5

list() (Cloudforms.managers.provision_request.ProvisionRequestManager method), 7

list() (Cloudforms.managers.tag.TagManager method), 8

list() (Cloudforms.managers.task.TaskManager method), 9

list() (Cloudforms.managers.vs.VSManager method), 11

P

perform_action() (Cloudforms.managers.provider.ProviderManager method), 5

perform_action() (Cloudforms.managers.provision_request.ProvisionRequestManager method), 7

perform_action() (Cloudforms.managers.vs.VSManager method), 11

ProviderManager (class in Cloudforms.managers.provider), 3

ProvisionRequestManager (class in Cloudforms.managers.provision_request), 6

R

refresh() (Cloudforms.managers.provider.ProviderManager method), 5

S

ServiceTagManager (class in Cloudforms.managers.tag), 7

start() (Cloudforms.managers.vs.VSManager method), 11

start() (Cloudforms.managers.vs.VSManager method), 11

suspend() (Cloudforms.managers.vs.VSManager method), 12

T

TagManager (class in Cloudforms.managers.tag), 8

TaskManager (class in Cloudforms.managers.task), 9

U

unassign() (Cloudforms.managers.tag.ServiceTagManager method), 8

update() (Cloudforms.managers.provider.ProviderManager
method), 5

V

VSManger (class in Cloudforms.managers.vs), 10

W

wait() (Cloudforms.managers.provision_request.ProvisionRequestManager
method), 7

wait() (Cloudforms.managers.task.TaskManager
method), 9