
py-genesiscloud

Release 0.2.1

Apr 21, 2021

Contents:

1	Introduction	1
2	Quick start	3
3	Instance creation example	5
4	API	11
4.1	genesiscloud package	11
	Python Module Index	13
	Index	15

CHAPTER 1

Introduction

py-genesiscloud is a client library for interacting with GenesisCloud's public API. It allows you to create resources in GenesisCloud via Python programs.

CHAPTER 2

Quick start

Before you can use the client, you need to create yourself an API token. Use the web console to create one. Keep this API token safe and for your eyes only. Save this token in a file which is readable for your user. Now, let's start.

You can create an instance of the client with:

```
>>> from genesiscloud.client import Client
>>> client = Client(os.getenv("GENESISCLOUD_API_KEY"))
```

The client has one public method, with it, you can verify that the API key is working:

```
>>> client.connect()
<Response [200]>
```

If the API key is invalid, you will get an HTTP error code. The client has property for each GenesisCloud resources which allows you to do CRUD operations on these resources. Note that currently not all resources support all CRUD operations via the API. Nevertheless, the methods exist for all resources.

Each resource has the following methods *get*, *find*, *list*, *create*, *delete*:: The *get* methods accept one parameter, the resource id:

```
>>> client.Instances.get('ee7a8707-33ef-49ae-9930-a4fb10c1b16a')
Instance({'id': '86d7b549-930e-44d1-8a42-a483292adf63', 'name': 'folding@home via API_
↳ 2', 'hostname': 'folding-at-home-api-2', 'type': 'vcpu-4_memory-12g_disk-80g_
↳ nvidia1080ti-1', 'allowed_actions': ['start', 'shutoff', 'reset'], 'ssh_keys': [{'id
↳ ': 'd57521aa-36fd-4cbf-a1a1-8344391a7893', 'name': 'cschmidbauer@gc'}]}, 'image': {
↳ 'id': 'edba720a-ba6b-4552-81eb-16fb91460e31', 'name': 'folding@home 1 gpu image'},
↳ 'security_groups': [{'id': '2472c0bb-1fa9-4dcc-a658-4268e78ad907', 'name': 'default
↳ '}, {'id': 'd3040f01-3b12-4712-9e8e-8ecblae7ba04', 'name': 'standard'}, {'id':
↳ '56370632-ceeb-4357-a5d3-f2c3acf9d69e', 'name': 'Folding@home'}]}, 'status': 'active
↳ ', 'private_ip': '192.168.10.108', 'public_ip': '194.61.20.206', 'created_at':
↳ '2020-03-26T18:49:18.771Z', 'updated_at': '2020-03-26T18:50:12.957Z'})
```

The *get* method always returns an instance of *ItemView*, which is special dictionary like class, which allows the user to conveniently to access attribute via dot notation. Some Instance properties are themselves *ItemViews*:

```
>>> instance = client.Instances.get('ee7a8707-33ef-49ae-9930-a4fb10c1b16a')
>>> instance.ssh_keys
[SSHKey({'id': 'd57521aa-36fd-4cbf-a1a1-8344391a7893', 'name': 'cschmidbauer@gc'})]
>>> instance.security_groups
[SecurityGroup({'id': '2472c0bb-1fa9-4dcc-a658-4268e78ad907', 'name': 'default'}),
 ↪ SecurityGroup({'id': 'd3040f01-3b12-4712-9e8e-8ecblae7ba04', 'name': 'standard'}),
 ↪ SecurityGroup({'id': '56370632-ceeb-4357-a5d3-f2c3acf9d69e', 'name': 'Folding@home'}
 ↪)]
>>> instance.security_groups[0].name
'default'
```

The *list* method returns a lazy generator to list all the resources or filter some:

```
>>> client.SecurityGroups.list()
<generator object GenesisResource.__list at 0x7f97f9405b48>
>>> list(client.SecurityGroups.list())
[SecurityGroup({'id': 'd3040f01-3b12-4712-9e8e-8ecblae7ba04', 'name': 'standard',
 ↪ 'description': 'The recommended security group default applied by Genesis Cloud',
 ↪ 'created_at': '2019-11-21T13:43:22.824Z'}), SecurityGroup({'id': '258afd7b-fb40-
 ↪ 439b-b146-bldcdbfead8c', 'name': 'iperf', 'description': 'iperf default port_
 ↪ settings', 'created_at': '2020-01-12T22:15:35.871Z'}), SecurityGroup({'id':
 ↪ 'f26a9bec-c254-4804-843e-d3d179464ec2', 'name': 'outbound-fully-opened',
 ↪ 'description': '', 'created_at': '2020-01-14T10:00:02.371Z'}), SecurityGroup({'id':
 ↪ '56370632-ceeb-4357-a5d3-f2c3acf9d69e', 'name': 'Folding@home', 'description': '',
 ↪ 'created_at': '2020-03-04T14:04:28.004Z'}), SecurityGroup({'id': '2781a739-bdd2-
 ↪ 44b7-ac3a-9d1d38999738', 'name': 'VNC', 'description': '', 'created_at': '2020-05-
 ↪ 03T17:23:00.822Z'}), SecurityGroup({'id': '8a8472a3-9af8-4f7e-a0fc-d8f7b40fa56c',
 ↪ 'name': 'cao-test', 'description': '', 'created_at': '2020-05-08T01:43:41.109Z'})]
```

The *list* method accepts parameters for page number and number of items per page:

```
>>> list(client.SecurityGroups.list(page=2, item=40))
```

The *find* method allows one to search for specific resources:

```
>>> list(client.SecurityGroups.find({'name': 'standard'}))
<generator object GenesisResource.find at 0x7f97f9405b48>
>>> list(client.SecurityGroups.find({'name': 'standard'}))
[SecurityGroup({'id': 'd3040f01-3b12-4712-9e8e-8ecblae7ba04', 'name': 'standard',
 ↪ 'description': 'The recommended security group default applied by Genesis Cloud',
 ↪ 'created_at': '2019-11-21T13:43:22.824Z'})]
```

The *delete* method allowed one to delete a specific resource using the id of the resource (currently only Snapshots and Instances):

```
>>> client.Instances.delete('ee7a8707-33ef-49ae-9930-a4fb10c1b16a')
```

These are the basics of using the client. In the next chapter, you can see a fully working example.

CHAPTER 3

Instance creation example

```
# MIT License
#
# Copyright (c) 2020 Genesis Cloud Ltd. <opensource@genesiscloud.com>
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
#
# Authors:
#   Oz Tiram <otiram@genesiscloud.com>
#
"""
An example script to show how to start a Genesis Cloud GPU instance
with custom user data to install the NVIDIA GPU driver.

Grab your API key from the UI and save it in a safe place.
on the shell before running this script

$ export GENESISCLOUD_API_KEY=secretkey
"""
```

(continues on next page)

(continued from previous page)

```

import os
import textwrap
import time
import subprocess as sp

from genesiscloud.client import Client, INSTANCE_TYPES

def simple_startup_script():
    """see the documentation of cloud init"""
    return textwrap.dedent("""
#cloud-config
hostname: mytestubuntu
runcmd:
- [ "apt", "install", "-y", "vim" ]
    """)

def get_startup_script():
    return """#!/bin/bash
set -eux

IS_INSTALLED=false
NVIDIA_SHORT_VERSION=430

manual_fetch_install() {
    __nvidia_full_version="430_430.50-0ubuntu2"
    for i in $(seq 1 5)
    do
        echo "Connecting to http://archive.ubuntu.com site for $i time"
        if curl -s --head --request GET http://archive.ubuntu.com/ubuntu/pool/
↪restricted/n/nvidia-graphics-drivers-"${NVIDIA_SHORT_VERSION}" | grep "HTTP/1.1" > /
↪dev/null ;
        then
            echo "Connected to http://archive.ubuntu.com. Start downloading and
↪installing the NVIDIA driver..."
            __tempdir="$(mktemp -d)"
            apt-get install -y --no-install-recommends "linux-headers-$(uname -r)" dkms
            wget -P "${__tempdir}" http://archive.ubuntu.com/ubuntu/pool/restricted/n/
↪nvidia-graphics-drivers-${NVIDIA_SHORT_VERSION}/nvidia-kernel-common-${__nvidia_
↪full_version}_amd64.deb
            wget -P "${__tempdir}" http://archive.ubuntu.com/ubuntu/pool/restricted/n/
↪nvidia-graphics-drivers-${NVIDIA_SHORT_VERSION}/nvidia-kernel-source-${__nvidia_
↪full_version}_amd64.deb
            wget -P "${__tempdir}" http://archive.ubuntu.com/ubuntu/pool/restricted/n/
↪nvidia-graphics-drivers-${NVIDIA_SHORT_VERSION}/nvidia-dkms-${__nvidia_full_
↪version}_amd64.deb
            dpkg -i "${__tempdir}"/nvidia-kernel-common-${__nvidia_full_version}_amd64.
↪deb "${__tempdir}"/nvidia-kernel-source-${__nvidia_full_version}_amd64.deb "${__
↪tempdir}"/nvidia-dkms-${__nvidia_full_version}_amd64.deb
            wget -P "${__tempdir}" http://archive.ubuntu.com/ubuntu/pool/restricted/n/
↪nvidia-graphics-drivers-${NVIDIA_SHORT_VERSION}/nvidia-utils-${__nvidia_full_
↪version}_amd64.deb
            wget -P "${__tempdir}" http://archive.ubuntu.com/ubuntu/pool/restricted/n/
↪nvidia-graphics-drivers-${NVIDIA_SHORT_VERSION}/libnvidia-compute-${__nvidia_full_
↪version}_amd64.deb

```

(continues on next page)

(continued from previous page)

```

        dpkg -i "$(__tempdir)"/nvidia-utils-${__nvidia_full_version}_amd64.deb "$(__
↪tempdir)"/libnvidia-compute-${__nvidia_full_version}_amd64.deb
        IS_INSTALLED=true
        rm -r "$(__tempdir)"
        break
    fi
    sleep 2
done
}

apt_fetch_install() {
    add-apt-repository -s -u -y restricted

    # Ubuntu has only a single version in the repository marked as "latest" of
    # this series.
    for _ in $(seq 1 5)
    do
        if apt-get install -y --no-install-recommends nvidia-utils-${NVIDIA_SHORT_
↪VERSION} libnvidia-compute-${NVIDIA_SHORT_VERSION} \
            nvidia-kernel-common-${NVIDIA_SHORT_VERSION} \
            nvidia-kernel-source-${NVIDIA_SHORT_VERSION} \
            nvidia-dkms-${NVIDIA_SHORT_VERSION} \
            "linux-headers-${(uname -r)} dkms; then
            IS_INSTALLED=true
            break
        fi
        sleep 2
    done
}

main() {
    apt-get update
    if grep xenial /etc/os-release; then
        manual_fetch_install
    else
        apt_fetch_install
    fi
    # remove the module if it is inserted, blacklist it
    rmmod nouveau || echo "nouveau kernel module not loaded ..."
    echo "blacklist nouveau" > /etc/modprobe.d/nouveau.conf

    # log insertion of the nvidia module
    # this should always succeed on customer instances
    if modprobe -vi nvidia; then
        nvidia-smi
        modinfo nvidia
        gpu_found=true
    else
        gpu_found=false
    fi

    if [ "${IS_INSTALLED}" = true ]; then
        echo "NVIDIA driver has been successfully installed."
    else
        echo "NVIDIA driver has NOT been installed."
    fi
}

```

(continues on next page)

(continued from previous page)

```

fi

if [ "${gpu_found}" ]; then
    echo "NVIDIA GPU device is found and ready"
else
    echo "WARNING: NVIDIA GPU device is not found or is failed"
fi
}

main
"""

def create_instance():
    client = Client(os.getenv("GENESISCLOUD_API_KEY"))

    # before we continue to create objects, we check that we can communicate
    # with the API, if the connect method does not succeed it will throw an
    # error and the script will terminate
    if client.connect():
        pass
    # To create an instance you will need an SSH public key.
    # Upload it via the Web UI, you can now find it with.
    # replace this to match your key
    SSHKEYNAME = 'YourKeyName'

    # genesiscloud.client.Resource.find methods returns generators - that is,
    # they are lazy per-default.

    sshkey_gen = client.SSHKeys.find({"name": SSHKEYNAME})
    sshkey = list(sshkey_gen)[0]

    # You need to tell the client which OS should be used for your instance
    # One can use a snapshot or a base-os to create a new instance
    ubuntu_18 = [image for image in client.Images.find({"name": 'Ubuntu 18.04'})][0]

    # choose the most simple instance type
    # to see the instance properties, use
    # list(INSTANCE_TYPES.items())[0]
    #
    # ('vcpu-4_memory-12g_disk-80g_nvidia1080ti-1',
    # {'vCPUs': 4, 'RAM': 12, 'Disk': 80, 'GPU': 1})

    instace_type = list(INSTANCE_TYPES.keys())[0]
    # To create an instance use Instances.create
    # You must pass a ssh key to SSH into the machine. Currently, only one
    # SSH key is supported. If you need more use the command
    # `ssh-import-id-gh oz123`
    # it can fetch public key from github.com/oz123.keys
    # *Obviously* __replace__ my user name with YOURS or anyone you TRUST.
    # You should put this in the user_data script. You can add this in the
    # text block that the function `get_startup_script` returns.
    # NOTE:
    # you can also create an instance with SSH password enabled, but you should
    # prefer SSH key authentication. If you choose to use password, you should
    # not pass ssh_keys
    my_instance = client.Instances.create(

```

(continues on next page)

(continued from previous page)

```

name="demo",
hostname="demo",
ssh_keys=[sshkey.id], # comment this to enable password
image=ubuntu_18.id,
type=instance_type,
metadata={"startup_script":
    simple_startup_script()},
    #password="yourSekretPassword#12!"
)
# my_instance is a dictionary containing information about the instance
# that was just created.
print(my_instance)
while my_instance['status'] != 'active':
    time.sleep(1)
    my_instance = client.Instances.get(my_instance.id)
    print(f"{my_instance['status']}\r", end="")
print("")
# yay! the instance is active
# let's ssh to the public IP of the instance
public_ip = my_instance.public_ip
print(f"The ssh address of the Instance is: {public_ip}")

# wait for ssh to become available, this returns exit code other
# than 0 as long the ssh connection isn't available
while sp.run(
    ("ssh -l ubuntu -o StrictHostKeyChecking=accept-new "
     "-o ConnectTimeout=50 "
     f"{public_ip} hostname"), shell=True).returncode:
    time.sleep(1)

print("Congratulations! You genesiscloud instance has been created!")
print("You can ssh to it with:")
print(f"ssh -l ubuntu {public_ip}")
print("Some interesting commands to try at first:")
print("cloud-init stats # if this is still running, NVIDIA driver is still"
      " installing")
print("use the following to see cloud-init output in real time:")
print("sudo tail -f /var/log/cloud-init-output.log")
return my_instance

def destroy(instance_id):
    # finally destroy this instance, when you no longer need it
    client = Client(os.getenv("GENESISCLOUD_API_KEY"))
    client.Instances.delete(id=instance_id)

if __name__ == "__main__":
    instance = create_instance()
    instance_id = instance['id']
    # destroy(instance_id)

```


4.1 genesiscloud package

4.1.1 genesiscloud.client Module

exception genesiscloud.client.**APIError** (*code, message*)
An API Error Exception

class genesiscloud.client.**Client** (*apikey*)

connect ()

headers

class genesiscloud.client.**GenesisResource** (*apikey*)
Template class to represent an API end point

create (***kwargs*)

delete (*id, **kwargs*)

find (*filter*)

get (*id*)

headers

list (*page=1, items=10, json=False, raw=False*)

munchify (*item*)

class genesiscloud.client.**Image** (**args, **kwargs*)

class genesiscloud.client.**Images** (*apikey*)

class genesiscloud.client.**Instance** (**args, **kwargs*)

class genesiscloud.client.**Instances** (*apikey*)

```
class genesiscloud.client.ItemView(*args, **kwargs)
    Template class to represent an item returned from the API

    api_to_resource = {'security_groups': 'SecurityGroup', 'ssh_keys': 'SSHKey'}

class genesiscloud.client.SSHKey(*args, **kwargs)
class genesiscloud.client.SSHKeys(apikey)
class genesiscloud.client.SecurityGroup(*args, **kwargs)
class genesiscloud.client.SecurityGroups(apikey)
class genesiscloud.client.Snapshot(*args, **kwargs)
class genesiscloud.client.Snapshots(apikey)

    create(*kwargs)
genesiscloud.client.create_snapshot(obj, **kwargs)
```


g

`genesiscloud.client`, 11

A

api_to_resource (*genesiscloud.client.ItemView* attribute), 12

APIError, 11

C

Client (*class in genesiscloud.client*), 11

connect() (*genesiscloud.client.Client* method), 11

create() (*genesiscloud.client.GenesisResource* method), 11

create() (*genesiscloud.client.Snapshots* method), 12

create_snapshot() (*in module genesiscloud.client*), 12

D

delete() (*genesiscloud.client.GenesisResource* method), 11

F

find() (*genesiscloud.client.GenesisResource* method), 11

G

genesiscloud.client (*module*), 11

GenesisResource (*class in genesiscloud.client*), 11

get() (*genesiscloud.client.GenesisResource* method), 11

H

headers (*genesiscloud.client.Client* attribute), 11

headers (*genesiscloud.client.GenesisResource* attribute), 11

I

Image (*class in genesiscloud.client*), 11

Images (*class in genesiscloud.client*), 11

Instance (*class in genesiscloud.client*), 11

Instances (*class in genesiscloud.client*), 11

ItemView (*class in genesiscloud.client*), 11

L

list() (*genesiscloud.client.GenesisResource* method), 11

M

munchify() (*genesiscloud.client.GenesisResource* method), 11

S

SecurityGroup (*class in genesiscloud.client*), 12

SecurityGroups (*class in genesiscloud.client*), 12

Snapshot (*class in genesiscloud.client*), 12

Snapshots (*class in genesiscloud.client*), 12

SSHKey (*class in genesiscloud.client*), 12

SSHKeys (*class in genesiscloud.client*), 12