

Using the Cloud SDK

This document gives a quick introduction to using the Openstack Python client to discover and consume compute resources.

Before beginning, get an account as described in the [Introduction](#). This is the right place to read about writing python scripts. A companion document shows how to use the [Command Line](#).

Installation

From a suitable host (this was done on `linux.cs.uchicago.edu`), create your environment.

```
$ python3 -m venv cloudsdk
$ source cloudsdk/bin/activate
$ pip3 install openstacksdk
$ python -m openstack version
```

Configuration

Obtain your configuration from the [Web Interface](#). The preceding link is for authenticated and authorized users only. Please read [Intro](#) for access. After logging in to the web interface at <https://overcloud.cs.uchicago.edu>, click the menu on the left side, Project→API Access. On the right side of the top of the page, click "Download Openstack RC File"→"Openstack clouds.yaml file".

Save the clouds.yaml file to the current working directory, or see the [Upstream Docs](#) about config file search order:

```
USER_CONFIG_DIR
  Linux: ~/.config/openstack
  OSX: ~/Library/Application Support/openstack
  Windows: C:\Users\USERNAME\AppData\Local\OpenStack\openstack

SITE_CONFIG_DIR
  Linux: /etc/openstack
  OSX: /Library/Application Support/openstack
  Windows: C:\ProgramData\OpenStack\openstack
```

Sample Configuration File

```
clouds:
  openstack:
    auth:
      auth_url: https://overcloud.cs.uchicago.edu:5000
      username: "CNetID"
      password: "sekret"
```

```
project_id: YOUR PROJECT UUID
project_name: "CNetID"
user_domain_name: "CS_LDAP"
region_name: "RegionOne"
interface: "public"
identity_api_version: 3
```

The main flaw here is that you are taking risks by storing your University credentials in a file. Instead, you can use the API or web interface to create an Application Credential. For example,

```
openstack application credential create --secret sekret --role member --
expiration 2020-10-10:10:10:10 --restricted myapp
```

Testing and Usage

Ensure you can load the python module

```
python -m openstack version
```

Check your configuration

```
python -m openstack.config.loader
```

Check the exact error if one is displayed. For example, if you see

```
keystoneauth1.exceptions.auth_plugins.MissingRequiredOptions: Auth plugin
requires parameters which were not given: auth_url
```

You have failed to provide a configuration file. This is different from a syntax error, but not so different from a permission problem, for example.

Demo Code

The following Python code shows how to create and destroy a server, and obtain information about Networks, Images, and Flavors. Consult the [API Resource Documentation](#) to see what else is possible.

For your learning, contrast this with the equivalent actions that are taken in the [Command Line Usage Guide](#).

```
import openstack
from openstack.config import loader
# openstack.enable_logging(True)
config = loader.OpenStackConfig()
cloud = openstack.connect(cloud='openstack')
flavor = cloud.get_flavor_by_ram(2048)
```

```
# print(flavor.name)
image = cloud.get_image('20.04')
# cloud pprint(image)
# network = cloud.network.find_network('campus37')
network = cloud.get_network('campus37')
# cloud pprint(network)
try:
    server = cloud.create_server('myserver', image=image, flavor=flavor,
network=network, wait=True, auto_ip=True)
    # cloud pprint(server)
    # cloud pprint(cloud.get_server('myserver', detailed=False))
    cloud pprint(cloud.get_server('myserver', bare=True))
finally:
    cloud.delete_server('myserver', wait=True, delete_ips=True)
```

From:

<https://howto.cs.uchicago.edu/> - **How do I?**

Permanent link:

<https://howto.cs.uchicago.edu/cloud:recipe:coding?rev=1589485886>

Last update: **2020/05/14 14:51**

