

# Lab 6.1: The OpenStack SDK

## ID720 Virtualisation

### Introduction

Last time we logged onto our `cloudapi` server, created a Python *virtualenv* and installed the OpenStack CLI tools in it. It turns out that those tools are implemented the Python OpenStack SDK. In this lab we will start writing our own code to interact with our OpenStack project using this SDK. The SDK is already installed in our *virtualenvs* as a side effect of installing the CLI tools last time.

### 1 Create a credentials file

When we use the OpenStack SDK we need to supply credentials to access our project. The easiest way to do this is to use a `clouds.yaml` file. We will place this file in the same directory where we place our Python code.

First, log into the `cloudapi` server we used in the previous lab. Create a subdirectory of your home directory called `openstack`. We'll place all of our files for this lab in that directory.

Next, log into the web dashboard at <https://catalystcloud.nz>. Go to the API Access item on the main menu. From there you can pull down the menu labeled "Download OpenStack RC File" and select the "OpenStack clouds.yaml File" option. Download this file and copy it into the new working directory on the `cloudapi` server. Don't forget to add your password to this file.

### 2 Use the SDK to connect to OpenStack

Now write a small Python script, `connect.py` to try out connecting to our OpenStack project. Don't forget to activate your Python virtual environment with the command

```
source ~/id720/bin/activate
```

In your Python file, import the SDK by starting with the line

```
import openstack
```

then, connect to an OpenStack service with a line like

```
conn = openstack.connect(cloud_name='catalystcloud')
```

With this connection object we can do things like query the compute service to see what servers are present:

```
for server in conn.compute.servers():
    print(server)
```

We can connect to other services in the same way by replacing `compute` with other service types. Experiment with these services by consulting the documentation at <https://docs.openstack.org/openstacksdk/xena/user/index.html>

**N.B.:** During this lab, restrict your coding to methods that query OpenStack for information about our project. Don't write any code that creates OpenStack resources or alters their states.

### 3 A little more in depth

You can use the API to get information about a specific server using the `find_server` method. Use this method to “find” the `cloudapi` server in your zone. Query that server instance (see the docs above) to find information about the server, like it’s IP address(es).