Lab 8.2: The OpenStack CLI
IN720 Virtualisation

Introduction

In the previous lab we got started using the OpenStack web dashboard which works well for most of the system tasks we need to perform. However, in this lab we will see that there is also a comprehensive set of CLI tools that let us operate OpenStack very quickly and easily (at least once you get the hang of using them).

All of the tasks in this lab must be done on `cloudapi.foo.org.nz`, since the service provider's firewall has whitelisted only that address for remote CLI/API access.

Two notes:

1. If you removed your key pair after the last lab, you'll need to set up a new pair for this lab, then leave that key pair in place for future work. Otherwise just note your key pair name from last time and use it.

2. Preface all OpenStack resources you create with your user name so we can easily tell to whom things belong.

1 Set up your CLI environment

There are three things you need to do to prepare your command line environment on the server:

1. Create a Python virtual environment
2. Install OpenStack tools
3. Download and run your credentials rc file

1.1 Python virtual environment

Python virtual environments just let you install and use Python modules without changing the system-wide Python libraries. Create a virtual environment in your home directory on cloudapi with the command

```
virtualenv in720
```

Then, activate the environment with the command

```
source in720/bin/activate
```

1.2 OpenStack tools

Once this is done you can install the OpenStack tools into your virtualenv with the command

```
pip install python-{openstackclient,ceilometerclient,heatclient,neutronclient,swiftclient,octaviaclient}
```

1.3 OpenStack credentials

Log into the OpenStack web console. **Be sure to set your zone to the same one you used in the last lab.** At the top right you'll see your email address and a dropdown menu. Expand is and you'll see "OpenStack RC File v3" Select that to download the file which should be named `otago-polytechnic-openrc.sh`. Copy that file to your home directory on cloudapi.

Back on cloudapi, run the command

```
source otago-polytechnic-openrc.sh
```

You will be prompted for your OpenStack password. When you type it, nothing will be displayed in the terminal.

Important note: Settings from these scripts do not persist across login sessions. If you log out and log back in, you will need to run the commands

```
source in720/bin/activate
source otago-polytechnic-openrc.sh
```

before you can use the OpenStack tools.

2 Run a VM

From the section title it doesn't sound like there is much to do, but there are several subtasks.

1. Create a network for our VM;
2. Create a router and connect it to the appropriate networks;
3. Create the VM;
4. Allocate a public IP address and associate it with the VM.

2.1 Create a network

First we need to create a private network for our VM with a subnet, 192.168.X.0/24, but we need to find a free value for X. See what subnets are already in use with the command

```
openstack network list
```

Once we have a suitable value for the third octet, we can create our network and subnet with the commands

```
openstack network create --internal <username>-net
openstack subnet create --network <username>-net  --subnet-range 192.168.X.0/24 <username>-subnet
```

substituting our chosen value for X.

To make our subnet reachable, we also need to create and configure a router to connect our network to `public-net`.

```
openstack router create <username>-router
openstack router add-subnet <username>-router  <username>-subnet
openstack router set --external-gateway public-net <username>-router
```

2.2 Launch a VM

We can launch our VM with one command, but we have to specify a lot of options. Basically, we have to specify the options we chose in the setup dialogues last time.

```
openstack server create --image ubuntu-16.04-x86_64 \
                                        --security-group lab8.1-secgrp \
                                        --key-name <your-key> \
                                        --flavor c1.c1r1 \
                                        --network <username>-net \
                                         <username>-server
```

2.3 Associate a floating IP address

Now we will set up a floating IP address and associate it with our server so that we can access it over the Internet,

`openstack floating ip create public-net`

Make note of the IP address in the output from the above command.

`openstack server add floating ip <username>-server <ip-address>`

3 Log in to your VM

Now that your server is up and connected to the Internet, you should be able to ssh into it just as we did last time. Also, you can use the web dashboard to view your VM and the associated resources you just created.

4 Cleaning up

Finally, please delete the items you created so that we don't tie up resources.

```
openstack server delete <username>-server
openstack floating ip delete <io address>
openstack router remove subnet <username>-router <username>subnet
openstack router delete <username>-router
openstack subnet delete <username>-subnet
openstack network delete <username>-net
```

5 Further documentation

See `https://docs.openstack.org/python-openstackclient/latest/cli/command-list.html`