Maintained by: Andrew Pullin, Senior Software Designer, School of Computer Science
andrew.pullin@carleton.ca / x4338 / HP5165

Last Updated: September 11, 2018

SCS OpenStack Cloud

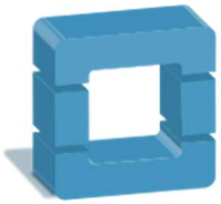

Table of Contents

Maintained by: Andrew Pullin, Senior Software Designer, School of Computer Science
andrew.pullin@carleton.ca / x4338 / HP5165

Last Updated: September 11, 2018

Introduction

The SCS OpenStack provides a virtual machine environment for SCS students, faculty and staff. The environment supports departmental services, course computing, and research computing.

NOTE: In OpenStack terminology, a virtual machine is referred to as an instance.

The SCS OpenStack can be used in a variety of scenarios depending on the clients' use cases, such as

- Instances that need significant amounts of vCPU, memory and disk space for compute-heavy applications
- Multiple instances for parallel computing (MPI, CILK, hadoop)
- Development servers requiring Internet-exposed IP addresses (these IPs are typically limited to the Carleton network)
- Isolated instances that can be used operating system and networking experiments
- General-purpose servers to support Faculty research or provide infrastructure to research labs

FACT: The popular shared linux infrastructure (vmicron) runs exclusively on the SCS OpenStack, with 8 to 16 instances available depending on demand.

Who Can Access the SCS OpenStack?

The SCS OpenStack is available to all faculty members, staff, students whose research would benefit from the resource, and students that require it for course work.

- Faculty and staff can use OpenStack resources to support departmental or research activities
    - Faculty members can leverage the OpenStack to manage their own dedicated research equipment, allowing them to quickly get their research going, without the hassle of system administration
- Graduate students can have their Faculty supervisor to request an OpenStack account if they need it for research.
- Undergraduate students will receive an account if they are in a course that is using OpenStack.

Maintained by: Andrew Pullin, Senior Software Designer, School of Computer Science
andrew.pullin@carleton.ca / x4338 / HP5165

Last Updated: September 11, 2018

SCS OpenStack Statistics

Here some quick facts about the SCS OpenStack cloud:

	Courses	Research	Network Info
Compute Nodes	**21**	**5**	**Admin Network Speed: 10 GB/s**
vCPUs	**464**	**100**	**Project Network Speed: 1 GB/s**
Memory	**2,044 GB**	**519 GB**	**Internet Access Speed: 10 GB/s**
Disk Space	**35.3 TB**	**2.8 TB**	
Volume Disk Space	**24.0 TB**	**-**	
GPUs	**4**	**5**	

Definitions

OpenStack – an open source cloud computing infrastructure that is supported as an industry standard. It provides all the technology to support the computer and networking virtualization required for a cloud environment. It uses a hypervisor, such as KVM (analogous to VirtualBox, or Microsoft Hyper-V), to provide vCPUs, Memory, Disk space, etc. For the SCS OpenStack, we use openvswitch, an open source virtual networking service, to provision virtual networks to the instances.

Compute Node – (or Hardware Node) a minimalist server that runs the OpenStack services, most importantly the hypervisor KVM, so that it an host multiple end-user instances.

Instance – an individual virtual machine running on OpenStack (setup by and end-user). An instance is a fully functional operating system, normally Linux, which runs on the OpenStack. When you launch an instance, you specify how many vCPUs and how much Memory and Disk space to allocate the server. OpenStack decides on which Hardware Node to place it based on current loads.

Image – An image is a ready-to-boot operating system. It is basically a snapshot of the disk of the operating system which can be used to create an instance. It is sort of like a hard drive with an operating system installed, that is not currently plugged into a computer. OpenStack copies the Image to a compute node, and then allocates CPU, Memory and Disk space from the compute node to that image to create an instance – analogous to plugging that hard drive back into your computer.

Cloud Ready Image – This is a pre-built image based operating system distributions, such as Ubuntu, fedora, etc. This is generally a minimalist image with some additional software that allows it to integrate easily into a cloud environment (such as OpenStack). It is typically setup without any usernames or passwords (you use an ssh private key file to connect to it – we talk more about this later).

Maintained by: Andrew Pullin, Senior Software Designer, School of Computer Science
andrew.pullin@carleton.ca / x4338 / HP5165

Last Updated: September 11, 2018

Using the Openstack Web Interface

This How-To section details the various ways in which OpenStack is used and how to use it.

Logging into the OpenStack Dashboard

The OpenStack dashboard provides the user interface for creating, monitoring and ultimately destroying instances. In most cases, once a user has created the instance, they will use other technologies such as SSH or HTTPS web connections to actually interact with the instance, rather than the OpenStack dashboard.

You can access the SCS OpenStack from the Carleton network: https://openstack.scs.carleton.ca/

Connecting to OpenStack from off Campus

Connecting to OpenStack from off campus requires using the Carleton University Virtual Private Networking (VPN) service to temporarily give your device a Carleton University IP address.

For details, see: https://scs.carleton.ca/technical-support/scs-open-stack (under "How to access SCS OpenStack")

Navigating the Interface

The OpenStack dashboard has three components.

1. The Identity Pane (bar at the top) where you can access user profile settings (username drop-down menu) and switch between projects (the project drop-down menu will show any projects to which you are a member).
2. The Navigation Pane on the left, which has a hierarchal navigation menu
3. Details Pane (where you see the details related to the selected Navigation item)

The Details Pane typically shows a list of items. For example, if you click on "Instances" in the Navigation Pane, you will see all of the instances of the currently selected project (or at least those that are visible to you).

In some cases, such as the "Access & Security" Navigation link, the resulting details will have multiple tabs that show different information (Security Groups, Key Pairs, Floating IPs, API Access).

When looking at a list of items – such as "Instances", "Images", or "Networks" – on the very right-hand side of the Details Pane, will be a button (or drop-down button) that shows all the actions you can perform on those particular items. Usually these actions will pop-up a window where you will confirm the action, or specify various details before performing the action. Also, if you want to perform an action on multiple instances (where applicable), you can select the check box on the left-hand side of the Details Pane for all the instances on which you wish to perform the action. You can then click the appropriate button on *above* the list.

For most users, there are two Navigation Menu items that you will use, which are found under the **Project -> Computer** menu:

1. **-> Images** – You will see a list of images (Project owned, Shared, and Public) that you can use to launch an instance.
2. **-> Instances** – You will see a list of currently running instances, and you can access them, modify them, or shut them down.
    - When you click on an instance, you will see a screen with four tabs (Overview, Log, Console, and Action Log). The most important tab is the **Console**, which allows you to see the running console of the server.
    - ***This is VERY useful if you are having trouble connecting to your instance via ssh, VNC, x2go, etc. This console will allow you to login to the instance directly as if you were sitting at a real desktop machine.***

Launching an Instance

The steps to launch an instance are as follows:

1. Click on "**Compute**" -> "**Images**"
2. Find the image you wish to launch as an instance and click the "**Launch Instance**" action button
3. Fill in the fields on the **Details** tab:
    a. **Availability Zone**: will always be **nova**
    b. **Instance Name**: give the instance a name that you will easily identify. OpenStack will also attempt to change the instances hostname to match this name.
    c. **Flavour**: select a flavour with the appropriate number of vCPUs, Memory and Disk space. They have been labelled with names that reflect their specifications. Once you select a flavour, its details will appear in the "**Flavour Details**" section to the right.
    d. **Instance Count**: The number of copies of the instance you wish to launch. If it is more than one (1), then OpenStack will append "-1", "-2", "-3", …, "-n" to the **Instance Name** that you gave above as each instance is created
    e. You can ignore the **Instance Boot Source** and the **Image Name** fields as these are automatically populated when you click on an image's "**Launch Instance**" action button.
4. Fill in the fields on the **Access & Security** tab:
    a. In the **Security Group** section, check the boxes of any services you wish to allow (such as "ssh and ping") **NOTE**: The names may vary, but generally are descriptive of what services you wish to permit.
    b. For most of our courses, you can ignore the **Key Pair** section, the images all have a standard user account. If you are using an image without an existing account, see the **Creating and Using an SSH Key Pair** section for details.
5. In the **Networking** tab, some courses will only have one network, in which case you can ignore the tab. Other courses may have more than one network (as a network is limited to 254 hosts).

    In this case, you should **FIRST** check the **Compute** -> **Access & Security** -> **Floating IPs** tab which will list all of the IPs available. Any that say **Down** are available. If you find an available IP such as 134.117.**216**.XXX, then you need to add the matching network on this **Networking** tab. The matching network for a subnet "**216**" IP would include the "**216**" at the end of its name. Here are some examples:

    Network **comp2406a-f18-net-216** would match any 134.117.**216**.XXX IP

    Network **comp2406a-f18-net-31** would match any 134.117.**31**.XXX IP
6. In the **Post-Creation**, and **Advanced Options** tabs, you should not have to adjust any settings
7. Click the **Launch** button

OpenStack will automatically select a hardware node on which to run the instance. If the image was already run on that node, then the image file should already be "staged", in which case the instance will launch within just a few seconds. If the image has not run on that node before, then the entire image (which may be several GB) will need to be copied to that node from the controller. This typically takes 10-30 seconds per GB of image, depending on the current network load. You can see the size of the Images under "**Compute**" -> "**Images**".

Associate Floating IP

Once an instance is launched, the button on the right of the list will include a "**Associate Floating IP**" action. Use this command to select a Floating IP address that can be used to connect to the instance remotely, as explained in the following sections.

REMEMBER: If you have multiple networks, you need to **find an available IP first**, and then add the **correct network** as indicated in the **Launching an Instance** section above.

Maintained by: Andrew Pullin, Senior Software Designer, School of Computer Science
andrew.pullin@carleton.ca / x4338 / HP5165

Last Updated: September 11, 2018

Openstack Connections and Networking

Connecting to an OpenStack Instance

To connect directly to an instance, you need to be on the SCS or the Carleton networks, just as was required for accessing the OpenStack web interface. See the **Connecting to OpenStack from off Campus** section above for details.

You can easily ssh from any linux system using the ssh commands in the following sections. An easy way to do this (if your computer or laptop is not running linux) would be to install one of our VirtualBox virtual machines on your personal computer, and then you can use it as a means to ssh to other systems (including OpenStack instances).

If you are off campus and you cannot use VPN, you can directly connect to one of the **SCS linux servers** using your **SCS linux account**. These servers are accessible from anywhere on the internet Detailed information on connecting to your **SCS linux account** can be found here: http://scs.carleton.ca/technical-support/linux-network

SSH – With a Username and Password

Most of our images, including our course virtual machines images (COMPXXX…), already have accounts created on them. In this case, you simply ssh to them as you would ssh to any other server on which you have an account (example shows the default student account included on most of our course images):

```
ssh student@134.117.31.20
(password: student)
```

To SSH using an ssh key, see the next section on **Creating and Using an SSH Key Pair**.

> **NOTE**: This is generally for more advanced use of openstack, and is not required for most courses as a username and password are supplied, as indicated above.

Creating and Using an SSH Key Pair

Key Pairs allow connecting to instances via SSH, a secure communications channel that is used for terminal connections, graphical connections, and for software APIs, including things like rsync and MPI. Theoretically, you only need to create one Key Pair to access your instances from any computer. As long as you have the **private key file** that you generate (with the following steps), you can use it to connect to your instance from almost anywhere.

Generating an SSH key through the openstack interface

The steps to create a Key Pair are as follows:

1. Click on **Compute** -> **Access & Security**
2. Click on the **Key Pairs** tab
3. Click the **Create Key Pair** button
4. Enter a suitable name in the **Key Pair Name** field. You should your username or some other convenient identifier in the name of the key pair so that you can easily identify your key pair in the key pair list
5. Click **Create Key Pair** button, which will cause a key pair to be generated. Automatically, a download will begin to download the private key file (.pem file)

Generate an SSH Key on the command line (of an linux system)

You can perform these steps on any linux system to create an ssh key:

1. Create the ssh key: (this will save the public/private keys in the default .ssh folder)
   `ssh-keygen -t rsa` *# Press <ENTER> for all questions, defaults are fine*
2. This key can then be saved and used in the above example
   NOTE: The key file created are **id_rsa.pub** (public file) and **id_rsa** (private file), different from the .pem extension used above. However, the key files are the same, and you can use any name you wish for the key files.
3. If you wish to use this ssh key for openstack, simply select the public key file (eg: id_rsa.pub) at your desired key, rather than having openstack create a key pair, as was done is the **Generating an SSH key through the openstack interface** section above.

Using an SSH Key

You can then use this key to ssh to the system using the following command:

```
ssh -i key-file-name ubuntu@134.117.31.20
```

NOTE: Different cloud ready images, such as Ubuntu, Fedora, Centos, use different usernames as their default for ssh key connections. Here are examples of using the key file with different operating system images:

- Ubuntu: username: **ubuntu**
  `ssh -i key-file-name ubuntu@134.117.31.20`
- Fedora: username: **fedora**
  `ssh -i key-file-name fedora@134.117.31.20`
- Centos (7 and higher): username: **centos**
  `ssh -i key-file-name centos@134.117.31.20`

NOTE: if you are using one of the provided "public" cloud ready images, then you MUST create this key pair. In general, we always name an image with the term **–cloudimg** at the end to indicate it is a cloud image and likely requires the use of a key pair. Here are some example image names:

```
ubuntu-16.04-server-amd64-cloudimg
centos-7-server-x86_64-cloudimg
fedora-20-server-x86_64-cloudimg
```

Connecting to Other Instances On Your Private Network

For many courses, you may need to run multiple instances and connect back and forth between these instances. In most cases, you can SSH with a username / password. However, if you are using applications that need to connect between instances, such as **Zoo, MPI, Hadoop**, etc, then you will need to create an ssh public-private key as you may have done to access the instance externally. More details are given below:

SSH – With a Key Pair's Private Key File

This procedure explains how to setup ssh via private/public keys:

1. Identify the host machine that will be the main server for your application (**Zoo, Hadoop, MPI, etc**), you will be performing these steps on that instance, as you need to grant that instance ssh key access to all the other instances you intend to use. Normally, the instance you use is the one that has a **public floating IP address** (if you have one)
2. Create the ssh key (using one of the steps found in the **Creating and Using an SSH Key** Pair section above
4. Next we copy they public key to the destination instance:
   `ssh-copy-id –i key-file-name username@192.168.111.111` # substitute your key-file-name, username and IP
5. To test, you can ssh to that instance, it should not ask for a password:
   `ssh –i key-file-name username@192.168.111.111`
6. Repeat this process for every instance to which you need to have ssh key access

SSH – With a Username and Password

As indicated, you can also ssh between the instances as you would between any servers, using the assigned private IP addresses:

```
ssh username@192.168.111.111
```