

SCS OpenStack Cloud

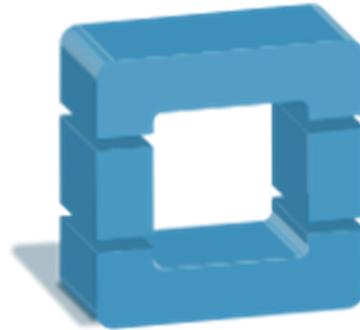


Table of Contents

Introduction	2
Who Can Access the SCS OpenStack?	2
SCS OpenStack Statistics	3
Definitions	3
Using the Openstack Web Interface	4
Logging into the OpenStack Dashboard	4
Connecting to OpenStack from off Campus	4
Navigating the Interface	4
Launching an Instance	5
Creating a Key Pair	5
Openstack Connections and Networking	7
Connecting to an OpenStack Instance	7
SSH – With a Key Pair’s Private Key File	7
SSH – With a Username and Password	7
Connecting to Other Instances On Your Private Network	8
SSH – With a Key Pair’s Private Key File	8
SSH – With a Username and Password	8

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.

SCS OpenStack Statistics

Here some quick facts about the SCS OpenStack cloud:

Compute Nodes: 10

vCPUs: 40

Memory: 480 GB

Node Disk Space: 500 GB / node

Optional Volume Disk Space: 48 TB

Admin Network Speed: 10 GB/s

Project Network Speed: 1 GB/s

Internet Access Speed: 1 GB/s (shared by all projects)

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 can 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).

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”)

For details on how to connect to your instance (virtual machine) – such as by using ssh – see the

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 regularly use:

1. **Compute -> Images** – You will see a list of images (Project owned, Shared, and Public) that you can use to launch an instance.
2. **Compute -> Instances** – You will see a list of the 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. The **Key Pair** section is one of the critical steps. If you are using an instance on which you do not already have an account, you may not be able to access the instance if you do not setup the Key Pair properly! Click the drop-down menu and select your key pair (you may have multiple key pairs to access OpenStack instances from different computers). You must first create a key pair, see the Creating a Key Pair section for details.
 - NOTE: If you *are* connecting to an instance on which you already have an account (ie, you know a valid username or password) or to an instance to which you already have a valid private key, **then you can SKIP the Key Pair field altogether!** This is **LIKELY** the case if you are using an image built from one of our VirtualBox course VMs.
5. In the **Networking, Post-Creation, and Advanced Options** tabs, you should not have to adjust any settings
6. 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 30-60 seconds per GB of image, depending on the current network load. You can see the size of the Images under **“Compute”** -> **“Images”**.

Creating a 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.

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 use 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)

Openstack Connections and Networking

Connecting to an OpenStack Instance

To connect to an instance, you need to be on the SCS or the Carleton networks, just as was required for accessing the OpenStack web interface. 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 machine on your personal computer, and then you can use it as a means to ssh to other systems (including OpenStack instances). Alternatively, you can directly connect to one of the SCS linux servers using your SCS linux account. Detailed information on connecting to your SCS linux account can be found here: <http://scs.carleton.ca/technical-support/linux-network>

NOTE: The gateway to the SCS network is the host **access.scs.carleton.ca**, however, we do not permit connecting to OpenStack instances directly from that machine, as this would put unnecessarily heavy loads on that server. Instead, we ask that you ssh to one of the omicron (**vmicron01 – vmicron08**) servers from **access**. The above link has all the details on how to do that. From those servers, you can connect to any part of the OpenStack network, and have easy access to your SCS linux account files if you wish to transfer them to your OpenStack instance (or copy data back from your instance).

SSH – With a Key Pair’s Private Key File

Most of our cloud ready images are designed to not have an account with a username and password. You instead connect with the default account using an ssh private key file, as you generated earlier. You can always choose to add a regular account with a username and password later, but this will be your initial way of connecting.

In order to use a key pair private key file, you run the following ssh command:

```
ssh -i private_key_file.pem ubuntu@134.117.31.20
```

The username “ubuntu” in the example above may vary depending on the distribution (or if you have setup a specific account with another name). The default username for cloud ready Ubuntu images is “ubuntu”, and – not surprisingly the fedora cloud ready image uses the default username “fedora”. A notable exception is if you are using an amazon EC2 cloud ready image, as they all use the “ec2-user” username.

In general, if you have trouble connecting with a key pair, contact support@scs.carleton.ca and we will help you out!

SSH – With a Username and Password

Many of our images, including our course virtual machines, 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:

```
ssh myusername@134.117.31.20
```

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

You can manually create an ssh key using the following commands on most linux platforms. In general, this has been tested on any images that are provided for courses that expect students to use ssh key authentication:

- 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)
- The first step is to create the ssh key, which will save the public/private keys in the default locations in the .ssh folder:
`ssh-keygen -t rsa # Press <ENTER> for all questions, defaults are fine`
- Next we copy the public key to the destination instance:
`ssh-copy-id username@192.168.111.111 # substitute your username and IP`
- To test, you can try sshing to that instance, it should not ask for a password:
`ssh user@192.168.111.111`
- 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
```