

Chameleon Cloud Tutorial

Chameleon API

## 

## Program Solicitation # NSF 13-602

## CISE Research Infrastructure: Mid-Scale Infrastructure - NSFCloud (CRI: NSFCloud)

**Application Program Interface**

**Objectives**

In this tutorial, you will learn about the Application program Interface ,how it works with open stack.

|  |  |
| --- | --- |
| **Tutorial Action** | **Time Required** |
| **Step 1: Introduction of API on Open Stack** You will learn what is API and how it works with Open Stack. | 5 minutes |
| **Step 2: Nova CLI installation and Configure RC file** To begin, we will learn the process of installing NOVA CLI and how to configure RC file | 10 minutes |
| **Step 3: Key Stone Introduction and installation** We will discuss about Key stone its features and how API works with Key Stone. | 15 minutes |
| **Step 4: Nova Usage** Lastly, we will implement the Nova on API | 10 minutes |

Prerequisites

* Knowledge of API
* A SSH Client(how to use SSH)
* Basic Knowledge of LINUX
* Basic knowledge on Open Stack applications.

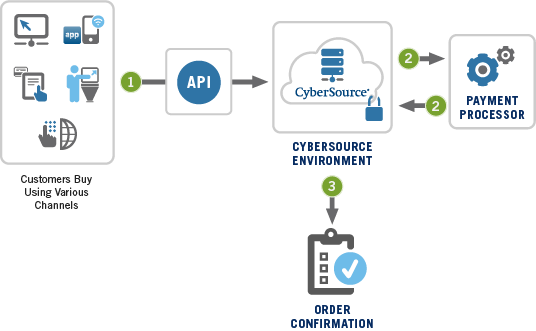
What is Application Program Interface ?

An application program interface (API) is code that allows two software programs to communicate with each other. Typically, APIs are released for third-party development as part of a software development kit (SDK) or as an open API published on the Internet.

What is Open API

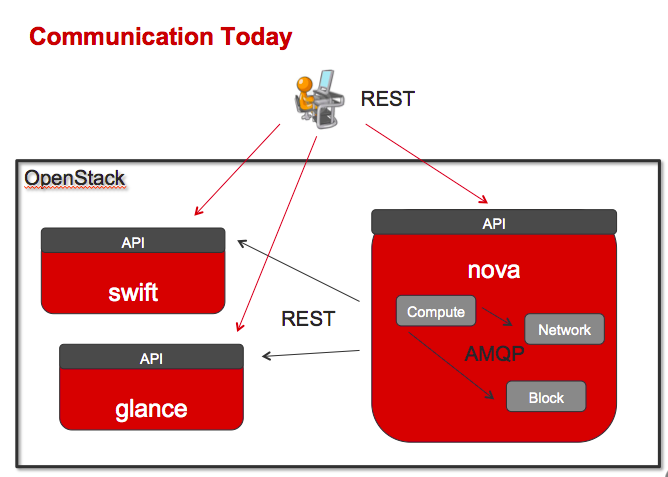
**Open API** (often referred to as OpenAPI new technology) is a term used in the Application Programming Interface (API) process that uses sets of technologies that enable websites to interact with each other by using REST, SOAP, JAVA SCRIPT other web technologies.

How API Works ?



**API for OPEN STACK**

A set of language bindings that provide a language-level API for accessing OpenStack in a manner consistent with language standards.



Each OpenStack project provides a command-line client, which enables you to access the project API through easy-to-use commands

Some of the Open Stack Services and Clients

|  |  |
| --- | --- |
| Service | Client |
| Block Storage | Cinder |
| Compute | Nova |
| Object Storage | Swift |

Nova CLI Installation:

Nova which is a client for Compute service which creates and manages images , instances and flavors.

Installing nova using pip

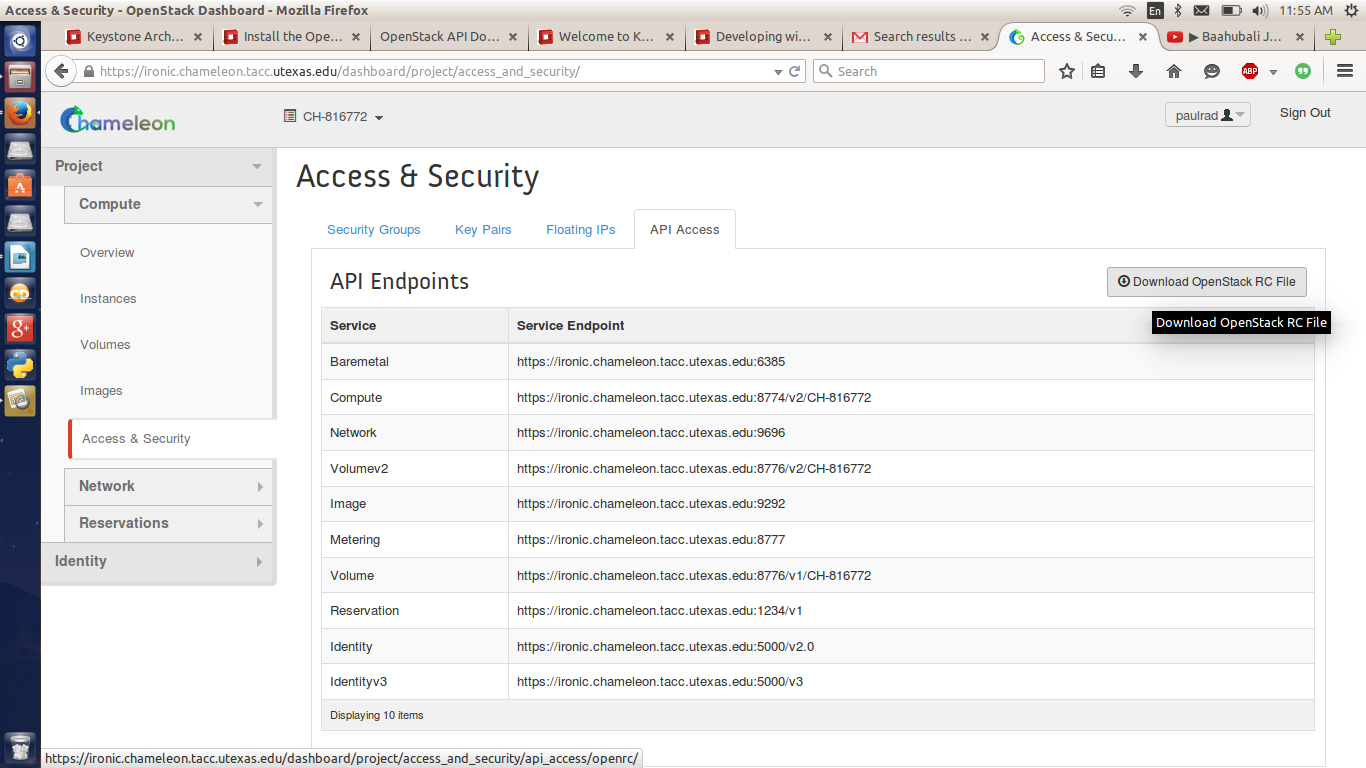
# pip install python-novaclient

Uninistall Nova

# pip uninstall python-novaclient

**Configuration of RC file**

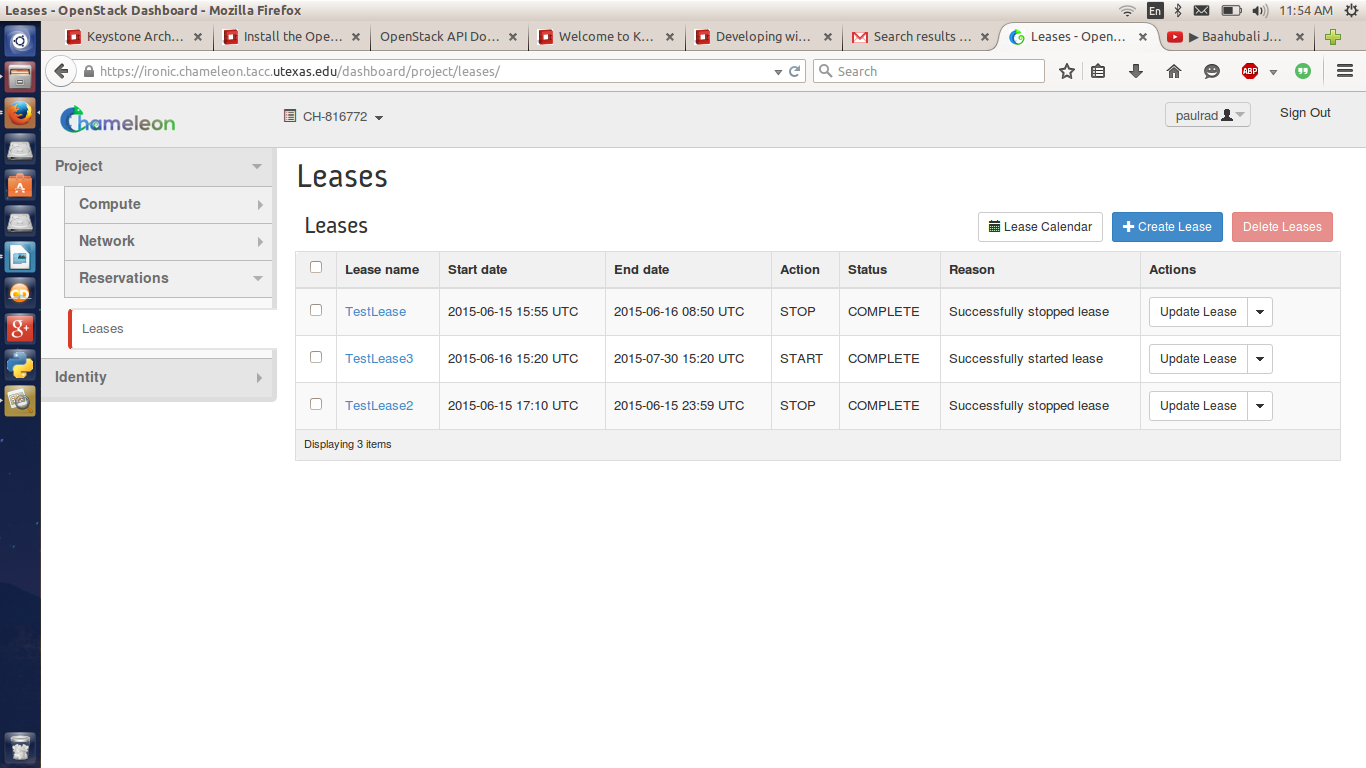
Login to the Chameloen Account → Project → Access & Security → API Access



Click on the Download Open Stack RC file and save it.

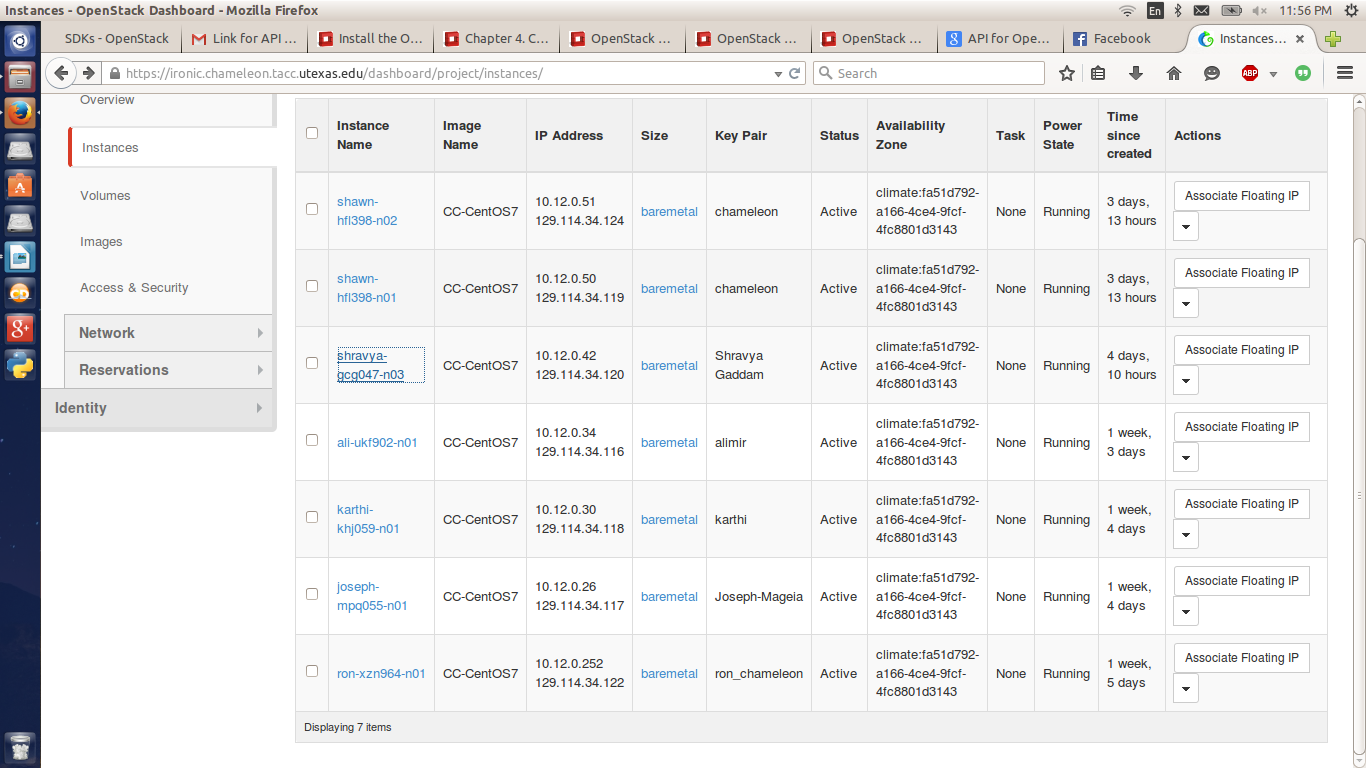
**Advanced Resource Leasing Reservation.**

**Login to the Chameloen account → Project → Reservation → Leases → Create Lease/Delete Lease.**



**Let us check how NOVA works**

* Using Chameleon account Create an instance



* Login to port that is associated with the IP address.
* Use the following Commands.

Open the RC configuration file that we have saved already.

[cc@shravya-gcg047-n03 ~]$ Cat CH-816772-openrc.sh

The file looks in the below format and verify the highlighted fields.

|  |
| --- |
| #!/bin/bash  # To use an Openstack cloud you need to authenticate against keystone, which  # returns a \*\*Token\*\* and \*\*Service Catalog\*\*. The catalog contains the  # endpoint for all services the user/tenant has access to - including nova,  # glance, keystone, swift.  #  # \*NOTE\*: Using the 2.0 \*auth api\* does not mean that compute api is 2.0. We  # will use the 1.1 \*compute api\*  export OS\_AUTH\_URL=https://ironic.chameleon.tacc.utexas.edu:5000/v2.0  # With the addition of Keystone we have standardized on the term \*\*tenant\*\*  # as the entity that owns the resources.  **export OS\_TENANT\_ID=CH-816772**  **export OS\_TENANT\_NAME="CH-816772"**  # In addition to the owning entity (tenant), openstack stores the entity  # performing the action as the \*\*user\*\*.  **export OS\_USERNAME="paulrad"**  # With Keystone you pass the keystone password.  echo "Please enter your OpenStack Password: "  read -sr OS\_PASSWORD\_INPUT  export OS\_PASSWORD=$OS\_PASSWORD\_INPUT  # If your configuration has multiple regions, we set that information here.  # OS\_REGION\_NAME is optional and only valid in certain environments.  export OS\_REGION\_NAME="regionOne"  # Don't leave a blank variable, unset it if it was empty  if [ -z "$OS\_REGION\_NAME" ]; then unset OS\_REGION\_NAME; fi |

[cc@shravya-gcg047-n03 ~]$ source CH-816772-openrc.sh

**KeyStone :** Keystone is an OpenStack project that provides Identity, Token, Catalog and Policy services for use specifically by projects in the OpenStack family. It implements Open Stack's Identity API

Prerequisites for Keystone development

* Git
* Phython latest version
* Virtual env
* pip
* msgfmt(part of gittext package)
* setuptools.

Setup steps for keystone

$ git clone <https://git.openstack.org/openstack/keystone.git>

installing dependencies—for ubuntu

$ sudo apt-get install python-dev python3-dev libxml2-dev libxslt1-dev \ libsasl2-dev libsqlite3-dev libssl-dev libldap2-dev libffi-dev

Installing virtual environment and activating it

$ python tools/install\_venv.py

$ source .venv/bin/activate

Install the dependencies for running keystone

$ pip install -r requirements.txt

Install the dependencies for developing, testing, and running keystone

$ pip install -r test-requirements.txt

Use 'python setup.py' to link Keystone into Python's site-packages

$ python setup.py develop

Verifying keystone is set up perfectly

$ source .venv/bin/activate

you should be able to import keystone with out any issue

$ python -c "import keystone"

Developing with Keystone

Configuring Keystone

* + - * + $ cp etc/keystone.conf.sample etc/keystone.conf

Running Keystone

$ tools/with\_venv.sh bin/keystone-all

Testing Keystone

$ pip install tox

To execute the tests just run

$ tox

Nova Usage

**Now Check for the NOVA USAGE**

[cc@shravya-gcg047-n03 ~]$ nova usage

Usage from 2015-05-27 to 2015-06-25:

+---------+--------------+-----------+---------------+

| Servers | RAM MB-Hours | CPU Hours | Disk GB-Hours |

+---------+--------------+-----------+---------------+

| 22 | 12716184.32 | 9031.38 | 144502.09 |

**+---------+--------------+-----------+---------------+**

**Now checking for the Key Pair.**

[cc@shravya-gcg047-n03 ~]$ nova keypair-list

+---------------+-------------------------------------------------+

| Name | Fingerprint |

+---------------+-------------------------------------------------+

| Paul Rad | 5f:3e:27:7b:df:17:fe:18:15:7c:eb:65:84:14:75:7f |

| Ron\_Laptop | 50:09:57:56:7f:80:53:e6:97:87:df:f0:b0:38:c9:e6 |

| Sam Silvestro | fa:c8:ba:fe:0e:7e:3e:a6:b3:b2:1f:61:25:31:37:0b |

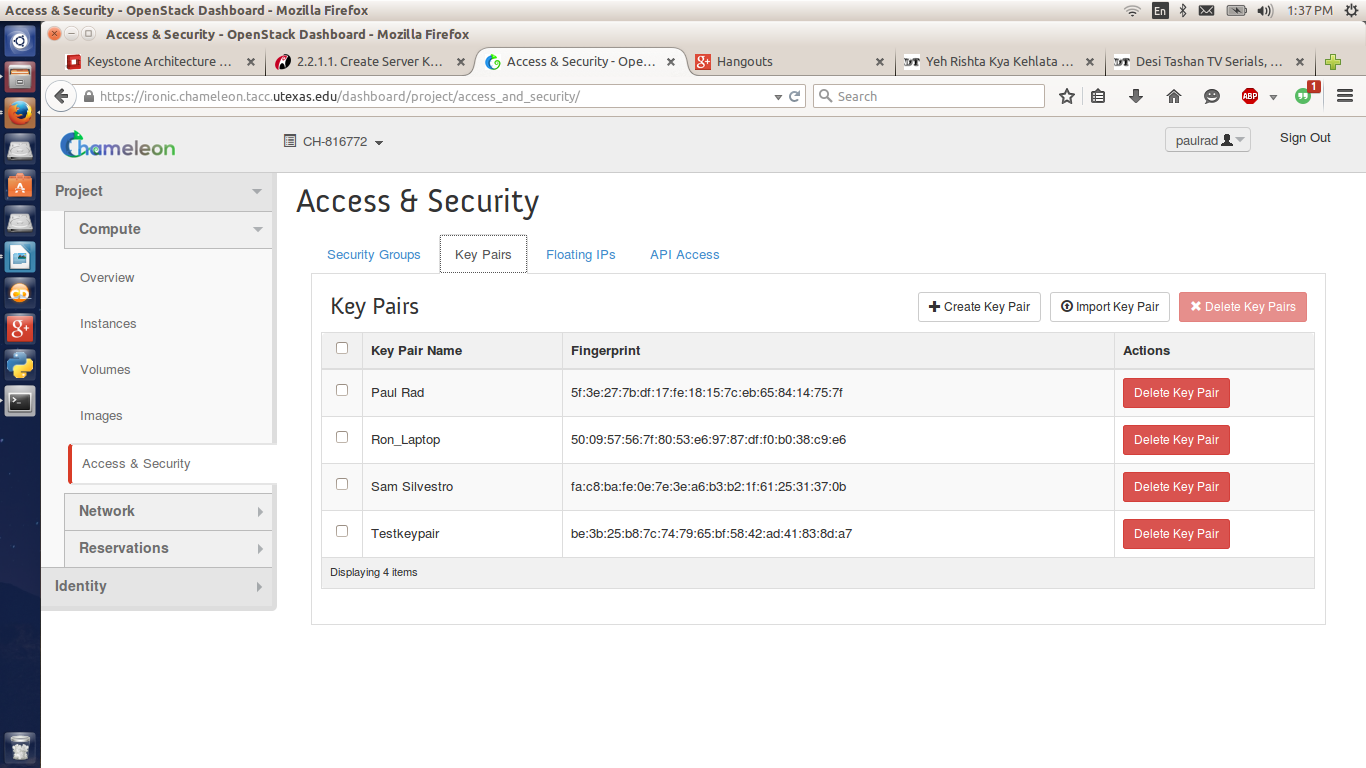
**+---------------+-------------------------------------------------+**

**T**he above command will trigger the API of Nova and display the key pairs that are associated with it.

**Using API and deleting the Keypairs.**

**Using a Command Nova-keypair add**

**[cc@shravya-gcg047-n03 ~]$ nova keypair-add Testkeypair > Test.pem**



**Checking for Key list to ensure the key pair is added**

**[cc@shravya-gcg047-n03 ~]$ nova keypair-list**

**+---------------+-------------------------------------------------+**

**| Name | Fingerprint |**

**+---------------+-------------------------------------------------+**

**| Paul Rad | 5f:3e:27:7b:df:17:fe:18:15:7c:eb:65:84:14:75:7f |**

**| Ron\_Laptop | 50:09:57:56:7f:80:53:e6:97:87:df:f0:b0:38:c9:e6 |**

**| Sam Silvestro | fa:c8:ba:fe:0e:7e:3e:a6:b3:b2:1f:61:25:31:37:0b |**

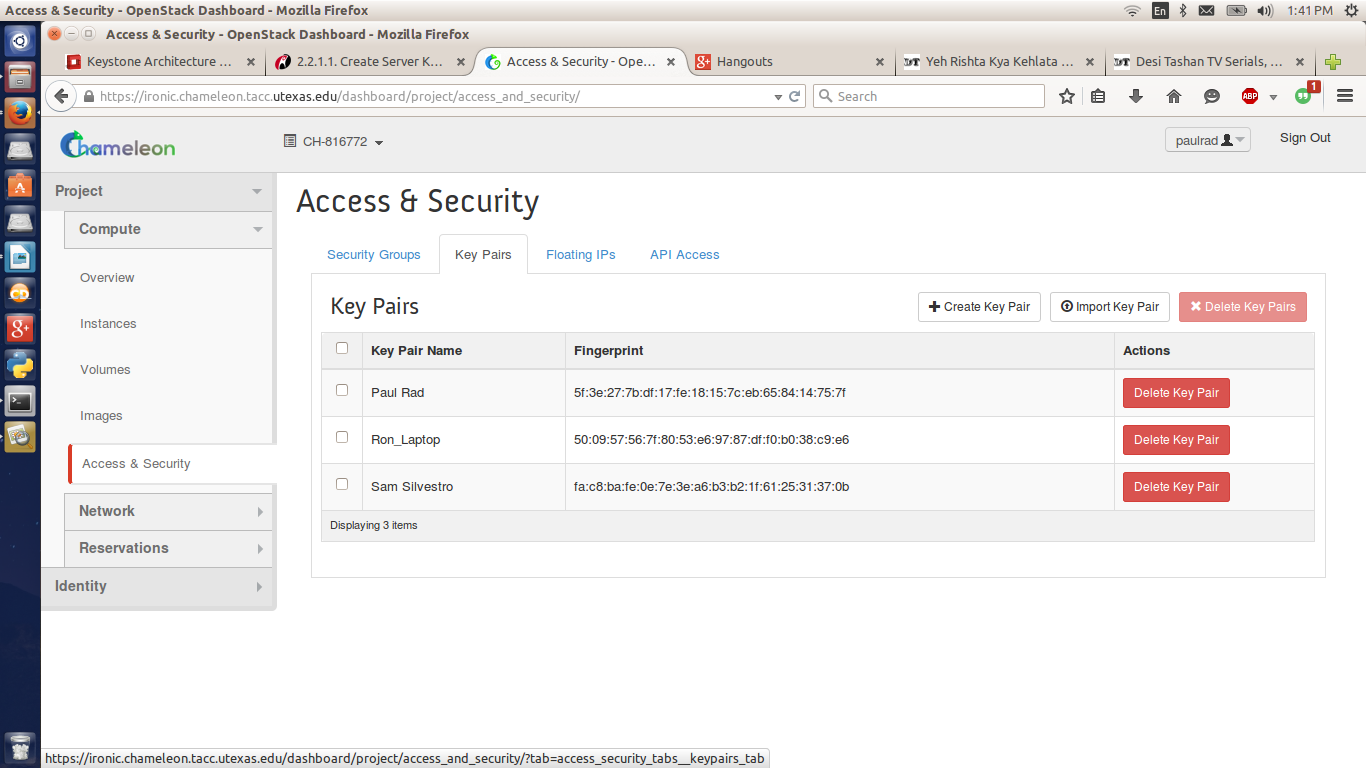
**| Testkeypair | be:3b:25:b8:7c:74:79:65:bf:58:42:ad:41:83:8d:a7 |**

**+---------------+-------------------------------------------------+**

**Usng a command Nova-keypair-delete**

[cc@shravya-gcg047-n03 ~]$ nova keypair-delete "TestKeypair"

This will trigger to the NOVA API and deletes a key pair associated with it



Checking the keypair list to verify whether it is deleted correctty

[cc@shravya-gcg047-n03 ~]$ nova keypair-list

+---------------+-------------------------------------------------+

| Name | Fingerprint |

+---------------+-------------------------------------------------+

| Paul Rad | 5f:3e:27:7b:df:17:fe:18:15:7c:eb:65:84:14:75:7f |

| Ron\_Laptop | 50:09:57:56:7f:80:53:e6:97:87:df:f0:b0:38:c9:e6 |

| Sam Silvestro | fa:c8:ba:fe:0e:7e:3e:a6:b3:b2:1f:61:25:31:37:0b |

+---------------+-------------------------------------------------+

This can be achieved using other Cloud Source API Like

Swift ,glance,neutron,trove,openstack,Cinder etc

**References**

**http://docs.openstack.org/user-guide/cli\_cheat\_sheet.html#block-storage-cinder**

**http://docs.openstack.org/cli-reference/content/install\_clients.html**