Configuring and using the openstack cloud at IBR

Johannes Starosta <j.starosta@tu-bs.de>

September 2012

This document describes how you use and administrate the openstack cloud of the Distributed Systems Group at TU Braunschweig. It also describes some basic concepts of openstack. *Warning: It's not a replace of the official documentation, use with caution!*

Contents

1 Introduction

Openstack consist of five modules, which serves different purposes:

- The Open Stack Compute Infrastructure is provided by the nova-daemon. nova is responsible for:
  - Instance life cycle management
  - Management of compute resources
  - Networking and Authorization

  It also provides a web service API, which can be used to control the instances. It's compatible with the EC2 API of Amazon Web Services and can be used by:
  - The nova-pythonclient, the euca2ools or any other EC2 compatible client, the Horizon Webinterface of Openstack
- The OpenStack Imaging Service ist provided by the glance daemon. It's used for retrival, lookup and uploading of virtual machine images. It can be accessed via the glance CLI client, any EC2/S3 compatible client and Horizon
- OpenStack Object Storage is provided by the Swift daemon. It can be acced by the swift CLI tool, Horizizon oder any Amazon S3 compatible client
- OpenStack Identity Service (Keystone) is used to set up user authentication and authorization. It can be used by using the keystone CLI tool.

2 Configuring users and their permissions with the Openstack Identity Service (Keystone)

2.1 Basic concepts: Users, tenants and roles

There are three main concepts of Identity user management are in Openstack:

- Users: Represents a human user (e.g. Alice,Bob) and has associated information such as usernae, password and email
- Tenants: A tenant can be thought of as a project, group, or organization (e.G: group1,IBR...)
- Roles: A role captures what operations a user is permitted to perform in a given tenant. A user can have different roles in different tenants. Here is an example: Imagine the users Alice and Bob. Both are members of the tenant IBR, as well as

of the tenant group1. Now see their roles:

User	Tenant	Role
Alice	IBR	admin
Bob	IBR	Member
Alice	group1	Member
Bob	group1	Member

Alice has the role „admin" of the tenant „IBR" while Bob has the role „users". In the tenant „group1" Alice and Bob are have the role "users". Please note, that „admin" and „users' are just names, unless you define the permission polices of a certain role. The currenty installed police on the Openstack Cloud at IBR just defines the roles „admin" and „swiftoperator". A user need to have the role „admin" to do some administrative tasks (e.G: Creating users...), which should not be allowed for normal users. If a user should be given access to the swift object storage service, he should get the role „swiftoperator".
Note: A user can have several roles for the same tenant:

User	Tenant	Role
Bob	IBR	Member
Bob	IBR	swiftoperator

For further information see Chapter 6, site 76 of *Compute Administration*.

2.2 Creating and managing users, tenants and roles

Now we want to create an user. Given is following scenario:
Alice and Bob are students of Eve. Alice and Bob participates in Eves cloud computing lab. So Eve wants them to work together on a shared project. They should be able to create custom images, upload them and run their own cloud instances (thus they need access to compute and glance). They also should use Swift Object Storage, so they can upload their documentation to an Object Storage container. Eve decides to create a tenant „cclab-group1" for them. Eve wants to have acces to the tenants instances as well. She also wants to have access to the tenant as well and she wants, administrative permissions, while Alice and Bob shall be just normal users... So she want to create following setup:

User	Tenant	Role
Alice	cclab-group1	Member
Bob	cclab-group1	Member
Alice	cclab-group1	swiftoperator
Bob	cclab-group1	swiftoperator
Eve	cclab-group1	admin

WARNING:

The environment on cloud2 is configured to do administrative tasks. Don't use it to do things, you want to use with a certain user! For example if you create an virtual machine instance as user „admin", you won't be able to access it from your normal user account. In section 3 we'll discuss how to access your user environment without messing up with the admin account.

She connects to `cloud2.ibr.cs.tu-bs.de` and run following commands in the admin console[1]

```
# Every line, which starts with a # is a comment
# First she wants to know, which users, roles and tenants alreday
#exists
eve@uncinus:~$ keystone user-list
+----------------------------------+---------+-------------------------+-----------+
|                id                | enabled |          email          |    name   |
+----------------------------------+---------+-------------------------+-----------+
| 1f53b11a2e0e40239b25961380472c54 |   True  |  admin@ibr.cs.tu-bs.de  |   swift   |
| 552bfa0c2d6847b6a8ea2625bff5531f |   True  |  admin@ibr.cs.tu-bs.de  |   glance  |
| cfb1870fc4c84a56bd1f19a5e577d984 |   True  |  admin@ibr.cs.tu-bs.de  |    nova   |
| dd69242cbdb243dd803b7641c223adb4 |   True  |  admin@ibr.cs.tu-bs.de  |   admin   |
| eveIDint                         |   True  |   eve@ibr.cs.tu-bs.de   |    eve    |
+----------------------------------+---------+-------------------------+-----------+
eve@uncinus:~$
eve@uncinus:~$ keystone tenant-list
+----------------------------------+----------+---------+
|                id                |   name   | enabled |
+----------------------------------+----------+---------+
| 0b6336de185a4f01a59fcb5ef9f40d96 | service  |   True  |
| 9f4b31c709b2431b972666100cf12c79 |  users   |   True  |
| fa188ad427e24bb6b14b4945fc6af6da |  admin   |   True  |
+----------------------------------+----------+---------+
eve@uncinus:~$ keystone role-list
+----------------------------------+---------------+
|                id                |      name     |
+----------------------------------+---------------+
| 2a292fb8580e49c095059a7b258b57f3 | swiftoperator |
| 45649d6a8b044f1d860905a896b3473b |     Member    |
| 47870520359d4f6b9d5d87a175a2f1f1 |     admin     |
+----------------------------------+---------------+
eve@uncinus:~$
# We already have all needed roles, eve still need to create the
```

[1] These are NOT real data ids. Don't use them in the real world!

```
# users, tenant and assign them to the needed roles
eve@uncinus:~$ keystone user-create --name bob --email bob@ibr.cs.tu-bs.de
                                    --pass bobpass
+----------+----------------------------------------------------------------------
| Property |                                                                Value
+----------+----------------------------------------------------------------------
|  email   |                                                    bob@ibr.cs.tu-bs.de
| enabled  |                                                                True
|    id    |                                                           bobsIDint
|   name   |                                                              bob
| password | $6$rounds=40000$0p.sAOcBsq8ms/Sk$B/yUdMQoxAwpM8kB1RwyHuL.mYrLI/BZZxlrDeT1U.
| tenantId |                                                              None
+----------+----------------------------------------------------------------------
eve@uncinus:~$ keystone user-create --name alice --email alice@ibr.cs.tu-bs.de --pass ali
+----------+----------------------------------------------------------------------
| Property |                                                                Value
+----------+----------------------------------------------------------------------
|  email   |                                                  alice@ibr.cs.tu-bs.de
| enabled  |                                                                True
|    id    |                                                          aliceIDint
|   name   |                                                             alice
| password | $6$rounds=40000$0p.sAOcBsq8ms/Sk$B/yUdMQoxAwpM8kB1RwyHuL.mYrLI/BZZxlrDeT1U.
| tenantId |                                                              None
+----------+----------------------------------------------------------------------
#now eve creates the tenant
eve@uncinus:~$ keystone tenant-create  --name cclab-group1
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |               None               |
|   enabled   |               True               |
|      id     | 898daab4fc4d45b98f231872241458e4 |
|     name    |           cclab-group1           |
+-------------+----------------------------------+
eve@uncinus:~$
# at the end we assign the needed roles to every user
#eve got role admin
eve@uncinus:~$ keystone user-role-add --user eveIDint
                       --role 47870520359d4f6b9d5d87a175a2f1f1
                      --tenant_id 898daab4fc4d45b98f231872241458e4
#bob and alice got the role Member
eve@uncinus:~$ keystone user-role-add --user bobIDint
                       --role 45649d6a8b044f1d860905a896b3473b
                      --tenant_id 898daab4fc4d45b98f231872241458e4
```

```
eve@uncinus:~$ keystone user-role-add --user aliceIDint
                       --role 45649d6a8b044f1d860905a896b3473b
                      --tenant_id 898daab4fc4d45b98f231872241458e4
#bob and alice got the role swiftoperator
eve@uncinus:~$ keystone user-role-add --user bobIDint
                      --role 2a292fb8580e49c095059a7b258b57f3
                      --tenant_id 898daab4fc4d45b98f231872241458e4
eve@uncinus:~$ keystone user-role-add --user aliceIDint
                      --role 2a292fb8580e49c095059a7b258b57f3
                      --tenant_id 898daab4fc4d45b98f231872241458e4
```

Now everything ist configured. Of course Eve could have add additional roles or more tenants. More information how to do identify managment can be found in chapter 6 of *Compute Administration*

2.3 Updating user information

You might want to change details of the users. Again we look in our scenario: Bob complains, that he can't remember his password. He prefers to have the password „swordfish". He also would prefered to have another email (bob@tu-bs.de) in the user-details. Again Eve connects to `clou2.ibr.cs.tu-bs.de`:

```
eve@uncinus:~$
eve@uncinus:~$ keystone user-update --email bob@tu-bs.de bobIDint
eve@uncinus:~$ keystone user-password-update --pass swordfish bobIDint
```

You can change the details of names and roles as well.

2.4 Removing everything

Now the term is finished. Alice and Bob have finished their project and passed the examination. Eve now wants to removes their user-account, tenant, since the Cloud will be needed for new students next term:

```
#first eve remove the roles from the tenant
eve@uncinus:~$ keystone user-role-remove --user eveIDint
                      --role 47870520359d4f6b9d5d87a175a2f1f1
                      --tenant_id 898daab4fc4d45b98f231872241458e4
eve@uncinus:~$ keystone user-role-remove --user bobIDint
                      --role 45649d6a8b044f1d860905a896b3473b
                      --tenant_id 898daab4fc4d45b98f231872241458e4
eve@uncinus:~$ keystone user-role-remove --user aliceIDint
                      --role 45649d6a8b044f1d860905a896b3473b
                      --tenant_id 898daab4fc4d45b98f231872241458e4
eve@uncinus:~$ keystone user-role-remove --user bobIDint
                      --role 2a292fb8580e49c095059a7b258b57f3
```

```
                        --tenant_id 898daab4fc4d45b98f231872241458e4
eve@uncinus:~$ keystone user-role-remove --user aliceIDint
                        --role 2a292fb8580e49c095059a7b258b57f3
                        --tenant_id 898daab4fc4d45b98f231872241458e4
# now we remove the tenant:
eve@uncinus:~$ keystone tenant-delete  898daab4fc4d45b98f231872241458e4
# and in the end the accounts of alice and bob
eve@uncinus:~$ keystone user-delete aliceIDint
eve@uncinus:~  keystone user-remove bobIDint
```

3 Using the compute infrastructure

As already mentioned in 2.2 you should not use the environment of the admin user to create new images or instances. Instead you should connect from a client. You can connect via the native openstack client tools(nova-pythonclient,glance,swift etc), any EC2-compatible tool, the HybridFox Firefox Plugin or the Horizon Dashboard. We will describe now how to configure the clients, before describing how to use them.

3.1 Configuring the native tools

Open a web browser and connect to `http://cloud2.ibr.cs.tu-bs.de`. You should see the login page of the Horizon dasboard. Login with your username and password (e.G. Alice, alicepass). Open the subpage „Settings". Open „OpenStack Credentials". Select the project you want to work on and click „Download RC file". The browser will start do download a file, called `openrc.sh`. The native tools are configured by setting environment variables, e.G. „OS_PASSWORD". Thus, you need to source the file „openrc.sh" before you can use the native tools:

```
# source openrc.sh
alice@izcip01:~$ source openrc.sh
Please enter your OpenStack Password:
alice@izcip01:~$
```

If you don't like the idea of always retyping the password, open `openrc.sh` in your favourite text editor and change the last lines to look like this:

```
# With Keystone you pass the keystone password.
#echo "Please enter your OpenStack Password: "
#read -s OS_PASSWORD_INPUT
#export OS_PASSWORD=$OS_PASSWORD_INPUT
export OS_PASSWORD=yourpassword
```

However you still need to source `openrc.sh` everytime you want to access the cloud. If you have enough of it, you can configure your UNIX shell, to source `openrc.sh` every time you login to your UNIX account:

```
# First change openrc.sh like described above, so you not getting asked
# for the password
# Make sure, that openrc.sh can be found by the shell:
alice@izcip01:~$ mv openrc.sh .novarc
# open the file .bashrc in your home directory in your favourite text
# editor and add following line to it
source ~/.novarc
#logout and login again, to activate the changes
#test it
alice@izcip01:~$ nova absolute-limits
+--------------------+-------+
|        Name        | Value |
+--------------------+-------+
|    maxImageMeta    |  128  |
|   maxPersonality   |   5   |
| maxPersonalitySize | 10240 |
|   maxServerMeta    |  128  |
|   maxTotalCores    |   20  |
| maxTotalInstances  |   10  |
|  maxTotalRAMSize   | 51200 |
+--------------------+-------+
```

Now you should be able to use the native tools every time you login to your UNIX-account.

3.2 Creating and running instances

We need to do following steps, to get a cloud instance to work:

1. Create and upload a virtual machine image
2. Create a public keypair
3. Run a virtual machine instance
4. Assign a public IP Address to the instance (optional)

Please note, that although step 4 is optional, in reality the virtual machine is nearly useless without a public IP. Without a public IP, the services provided by the virtual machine, can not be accessed.

3.2.1 Uploading virtual machine images

Since the creation of virtual machine images is quite complex, I put the details in a seperate section 4 on page 14. So let's assume you have already created an image or just downloaded one of the ready Images from `http://cloud-images.ubuntu.com/`. To add the image test.img to the cloud we upload it using the glance command line tool:

```
alice@izcip01:~$ glance add disk_format=qcow2  container_format=ovf name="test" <test.img
====================[100%] 145.932926M/s, ETA  0h  0m  0s
Added new image with ID: 3d183184-5fa7-475e-bacd-6e78c5e8c245
# note: disk_format shall be in the image format of the uploaded
# image, container_format is the format, which openstack will use to
# store. More information is provided in the section about creating images
alice@izcip01:~$
alice@izcip01:~$ glance index
ID                                   Name                           Disk Format
------------------------------------ ------------------------------ --------------------
3d183184-5fa7-475e-bacd-6e78c5e8c245 test                            qcow2
```

3.2.2 Create a public keypair

We need to create a public keypair. It's used to provide ssh access to UNIX images.

```
alice@izcip01:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alice/.ssh/id_rsa):
Your public key has been saved in dd.pub.
The key fingerprint is:
d6:e2:37:41:02:76:4b:9f:53:e6:f9:a7:36:65:8b:e4 jstarosta@uncinus
The key's randomart image is:
+--[ RSA 2048]----+
|      o o   o    |
|     . + o = .   |
|        o = o    |
|         + . .   |
|        S o  .. +|
|       o . .o .=.|
|        . o  E+. |
|         . . . . |
|                 |
+-----------------+
alice@izcip01:~$ nova keypair-add --pub_key .ssh/id_rsa.pub cloudkey
alice@izcip01:~$ nova keypair-list
+----------+-------------------------------------------------+
|   Name   |                   Fingerprint                   |
+----------+-------------------------------------------------+
| cloudkey | c2:df:5d:2d:ce:a6:64:af:08:64:fa:9a:c5:9f:27:19 |
+----------+-------------------------------------------------+
```

3.2.3 Run a virtual machine instance

```
alice@izcip01:~$ nova boot --flavor m1.tiny --image 3d183184-5fa7-475e-bacd-6e78c5e8c245
```

```
+------------------------+--------------------------------------+
|        Property        |                Value                 |
+------------------------+--------------------------------------+
| OS-DCF:diskConfig      | MANUAL                               |
| OS-EXT-STS:power_state | 0                                    |
| OS-EXT-STS:task_state  | scheduling                           |
| OS-EXT-STS:vm_state    | building                             |
| accessIPv4             |                                      |
| accessIPv6             |                                      |
| adminPass              | zS335ooJBMT4                         |
| config_drive           |                                      |
| created                | 2012-10-01T04:12:14Z                 |
| flavor                 | m1.tiny                              |
| hostId                 |                                      |
| id                     | 24d96fdd-55c2-4440-b096-f31aaa148966 |
| image                  | bla                                  |
| key_name               | cloudkey                             |
| metadata               | {}                                   |
| name                   | testinstance                         |
| progress               | 0                                    |
| status                 | BUILD                                |
| tenant_id              | fa188ad427e24bb6b14b4945fc6af6da     |
| updated                | 2012-10-01T04:12:14Z                 |
| user_id                | dd69242cbdb243dd803b7641c223adb4     |
+------------------------+--------------------------------------+
#get a coffe, building the virtual machine take some time
alice@izcip01:~$ nova list
+--------------------------------------+----------------+--------+----------------------
|                  ID                  |       Name     | Status |               Networ
+--------------------------------------+----------------+--------+----------------------
| 24d96fdd-55c2-4440-b096-f31aaa148966 | testinstance   | ACTIVE | private=192.168.4.37
+--------------------------------------+----------------+--------+----------------------
```

3.2.4 Assign a public IP Address to the instance

A new virtual machine instance get's automatically a private IP from the subnet 192.168.4.2/27 It's used for communication with other instances and the hypervisor. However if we want to access a certain instance from other servers, it needs to get a public IP assigned. The IBR network has a subnet 134.169.35.2/27 which provides a pool of public IPs for openstack:

```
# allocate a floating-ip from the pool
alice@izcip01:~$ nova floating-ip-create
+---------------+-------------+----------+------+
|       Ip      | Instance Id | Fixed Ip | Pool |
```

```
+---------------+-------------+----------+------+
| 134.169.35.12 | None        | None     | nova |
+---------------+-------------+----------+------+
#assign floating ip to the virtual machine instance with name "testinstance"
lice@izcip01: ~$ nova add-floating-ip testinstance 134.169.35.12
jstarosta@uncinus:~$ nova list
+--------------------------------------+----------------+--------+----------------------
|                  ID                  |       Name     | Status |               Networ
+--------------------------------------+----------------+--------+----------------------
| 24d96fdd-55c2-4440-b096-f31aaa148966 | testinstance   | ACTIVE | private=192.168.4.37
+--------------------------------------+----------------+--------+----------------------
# allow ssh and pings to the instance
alice@izcip01:~$ nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0
alice@izcip01:~$ nova secgroup-add-rule default tcp 22 22 0.0.0.0/0
```

3.2.5 Terminating instances and cleanup the environment

At some point you'll propably want to deassign public IP, terminate instances or remove images. To remove a public IP from a certain instance (e.G. to use it with another one):

```
alice@izcip01:~$ nova remove-floating-ip testinstance 134.169.35.12
```

To deassociate a public IP (e.G if the users run out of them):

```
alice@izcip01:~$ nova floating-ip-delete  134.169.35.12
```

Terminate and delete an instance:

```
alice@izcip01:~$ nova delete testinstance
# if you have more than one instance with the same name
# use nova delete serverId (as printed from nova list)
```

Remove an obsolete image:

```
alice@izcip01:~$ glance index
ID                                   Name                           Disk Format
------------------------------------ ------------------------------ --------------------
3d183184-5fa7-475e-bacd-6e78c5e8c245 bla                            qcow2
alice@izcip01:~$ glance delete 3d183184-5fa7-475e-bacd-6e78c5e8c245
Delete image 3d183184-5fa7-475e-bacd-6e78c5e8c245? [y/N] y
Deleted image 3d183184-5fa7-475e-bacd-6e78c5e8c245
```

3.3 Configuring the euca2ools

Using the native tools is fine, there is however a problem: In the workstation pool of the Braunschweig Computer Science center (roomz IZG40) they are not installed, only the

euca2ools. It's not a big deal, since Openstack is compatible with the EC2 API. We just need to to some preparing work.

Open a web browser and connect to `http://cloud2.ibr.cs.tu-bs.de`. You should see the login page of the Horizon dasboard. Login with your username and password (e.G. Alice, alicepass). Open the subpage „Settings". Open „OpenStack Credentials". Select the project you want to work on and click „Download RC file". The browser will start do download a ZIP-Archive with a random name. Now open a shell and type following commands:

```
alice@izcip01:~$ mkdir ~/.euca
alice@izcip01:~$ cd ~/.euca
alice@izcip01:~$ unzip pathto_archiv.zip
alice@izcip01:~$ mv ec2rc.sh eucarc
# euca2ools need S3_URL, which is not defined
# (Openstack Bug https://bugs.launchpad.net/horizon/+bug/987678)
alice@izcip01:~$ echo "export S3_URL=http://cloud2.ibr.cs.tu-bs.de:3333" >>eucarc
#source eucarc automatically at the login
alice@izcip01:~$ echo "source ~/.euca/eucarc" >>~/.bashrc
#test the setup
alice@izcip01:~$ euca-describe-regions
REGION  nova    http://cloud2.ibr.cs.tu-bs.de:8773/services/Cloud
```

3.4 Creating and running instances with the euca2ools

Warning: This is just a short description, taken from Johannes Behls and Klaus Stengels slides for the Cloud Computing Lecture in summer ter 2012. It's not complete!

3.4.1 Preparing the virtual machine image

```
# create an image  image.raw as described
# creating the VM-Package (Bundle)                                    "
$ euca - bundle -image -i image.raw -d . --arch i386
# sent VM-Package to OPenstackCloud
# note: The Bucketname doesn't really matter, pick one you like :)
$ euca-upload-bundle -b  <bucket_name> -m image.raw.manifest.xml
# Register the bundle
$ euca-register <bucket_name>/image.raw.manifest.xml
# List own, registred VM bundles
$ euca-describe-images -o <user>
# Modify an image attributes
$ euca - modify - image - attribute -l -r all < vm_id >
# removing an VM bundle
$ euca-deregister <vm_id >
$ euca-delete -bundle -b <bucket_name>
```

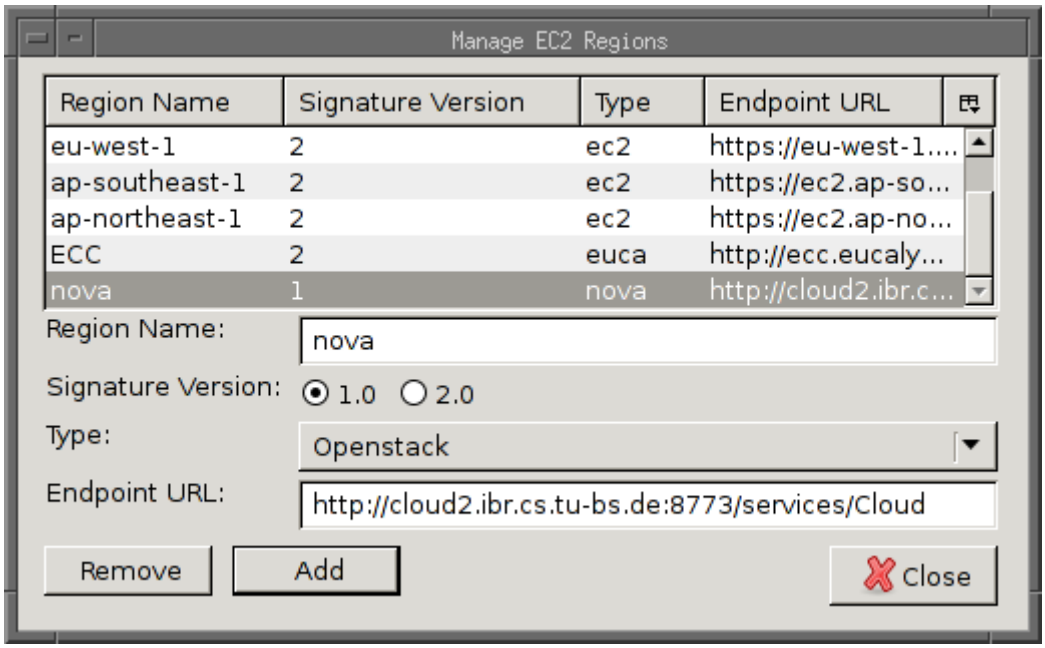
3.4.2 Running the virtual machine

```
$ euca-run-instances [-t <type>] [ -n <numbers_of_instances>] <vm_id>
#output:  ID of instance(s) (instance id)
# Get the status of the instance (state and IP)
$ euca-describe-instances
# terminate instance
$ euca - terminate - instances < instanz_id >
```

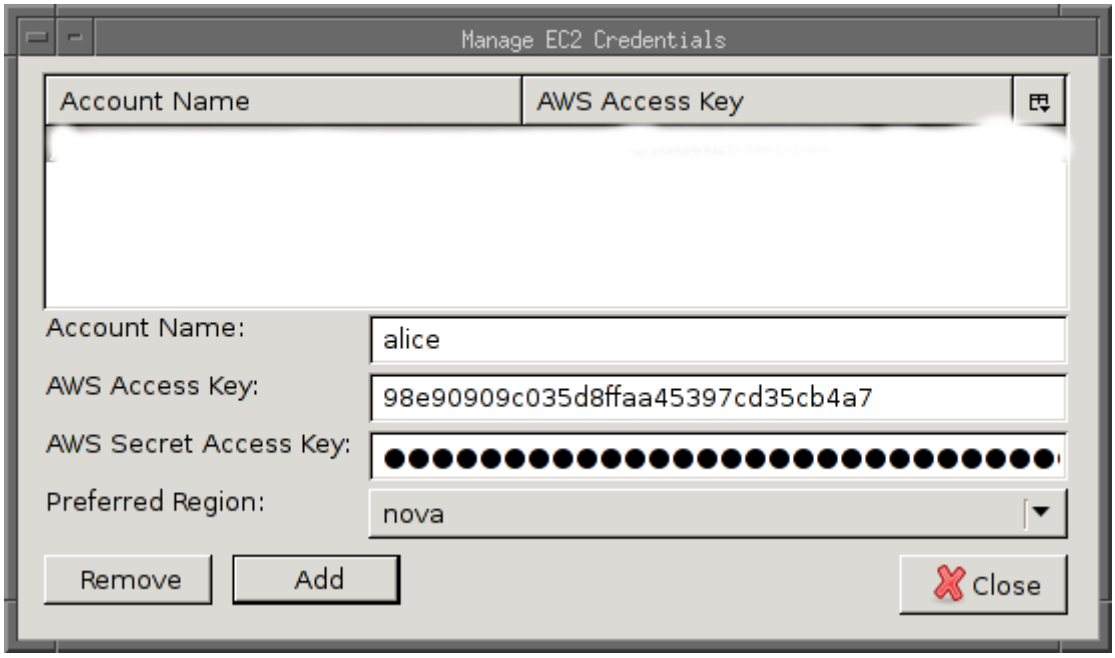
Note: This is NOT a complete guide, please refer to the *Euca2ools User Guide*.

3.5 Configuring and using Hybridfox

Hybridfox is a Firefox Add-On, which provides a GUI for several tasks related with cloud computing. It can be downloaded from `http://code.google.com/p/hybridfox/`. After installation it can be found in the Menu „Extras". We need to add an additional region before we can set up the credentials:



Now we setup the credentials:



Setting up images and instances is quite easy. More information concerning this, can be found in the *Hybridfox User Manual*.

4 Creating images

When we introduced the „glance" command line tool for uploading images, we didn't explain how to create images for use with the virtual machine. The principle is always the same: You create a disk image and use an emulator (e.G. Virtualbox, qemu, kvm etc) to install an operating system. Details how to accomplish this on the different Linux distributions and Windows can be found in chapter 3 „Image Managment" of the *Compute Starter Guide*. However one thing is important: You need to be beware, that you can only use an emulator whose image format is supported by glance. At the moment, glance support raw images, the VHD format used by VMWare, Xen, Microsoft, VirtualBox, vmdk, vdi used by qemu and VirtualBox, qcow2 of qemu and aki, ami and ari of amazon.

References

Euca2ools User Guide. http://open.eucalyptus.com/wiki/EucalyptusUserGuide.

Hybridfox User Manual. http://hybridfox.googlecode.com/files/Hybridfox_User_Manual_v1.pdf. CSS Corp | Confidential. 2011.

OpenStack Compute Administration Manual. http://docs.openstack.org/essex/openstack-compute/admin/content/index.html. OpenStack Foundation. Essex (2012.1).

OpenStack Compute Starter Guide. http://docs.openstack.org/essex/openstack-compute/admin/content/index.html. OpenStack Foundation. Essex (2012.1).