# Eucalyptus Beginner's Guide

## UEC Edition

Ubuntu 10.10 - Maverick

CSS CORP

# Eucalyptus Beginner's Guide - UEC Edition

## (Ubuntu Server 10.10 - Maverick Meerkat)

v2.0, 23 Dec 2010

Johnson D
Kiran Murari
Murthy Raju
Suseendran RB
Yogesh Girikumar

CSS CORP
ICT Services • Technology Adoption

# Eucalyptus Beginner's Guide - UEC Edition

## (Ubuntu Server 10.10 - Maverick Meerkat)

v2.0, 23 Dec 2010

This book is aimed at making it easy/simple for a beginner to build and maintain a private cloud using Eucalyptus. This is a live book updated periodically based on the suggestions, ideas, corrections, etc., from readers.

**Mail Feedback to**: css.ossbooks@csscorp.com

**For the latest version**: http://www.csscorp.com/eucauecbook

# Preface

## Introduction

When we started playing with Eucalyptus in early 2009 and with Ubuntu Enterprise Cloud a little later, we had a tough time before we could get a reasonably good grasp of the software. Eucalyptus documentation, a few white papers on the web sites of Eucalyptus Systems and Canonical Ltd., and discussions on forums were our key sources of information.

After we acquired a certain level of familiarity with the software, we felt the need for a beginner's guide of sorts to help our colleagues at CSS Corp venturing into Eucalyptus/UEC. We started writing a few wiki articles on an internal portal and when these articles were useful to some of our colleagues, we felt it would be a good idea to share the content with a wider audience. The result is this book.

## Target Audience

Our aim has been to provide a guide for a beginner to UEC. Good familiarity with virtualization is assumed, as troubleshooting many Eucalyptus-related problems requires a good knowledge of virtualization. Similarly, familiarity with Cloud Computing concepts and terminology will be of help. Prior exposure to AWS API and/or tools is not mandatory, though such exposure will accelerate the learning process greatly.

## Acknowledgement

Most of the content has been borrowed from web resources like manuals, documentation, white papers etc. from Canonical Limited, Eucalyptus Systems Inc.; numerous posts on Eucalyptus and Ubuntu forums; and many articles on the web including those of our colleagues at CSS Corp. We would like to thank all the authors of these resources.

## License

## Feedback

We would really appreciate your feedback. We will enhance the book on an ongoing basis based on your feedback. Please mail your feedback to css.ossbooks@csscorp.com.

# Contents

# Chapter 1

# Introduction to UEC and its components

## 1.1 Cloud

Cloud computing is a computing model, where resources such as computing power, storage, network and software are abstracted and provided as services on the internet in a remotely accessible fashion. Billing models for these services are generally similar to the ones adopted for public utilities. On-demand availability, ease of provisioning, dynamic and virtually infinite scalability are some of the key attributes of cloud computing.

An infrastructure setup using the cloud computing model is generally referred to as the "cloud". The following are the broad categories of services available on the cloud:

- Infrastructure As A Service (IAAS)
- Platform As A Service (PAAS)
- Software As A Service (SAAS)

This cloud is generally available as a service to anyone on the internet. However, a variant called "private cloud" is increasingly becoming popular for private infrastructure that has some of the attributes of the cloud as mentioned above.

Amazon Web Services (AWS) is one of the major players providing IAAS. They have two popular services – Elastic Compute Cloud (EC2) and Simple Storage Service (S3). These services are available through web services interfaces. The client tools can use EC2 and S3 APIs to communicate with these services. The popularity of these APIs have encouraged other cloud products to provide support for them as well.

## 1.2 Eucalyptus

Eucalyptus is a software available under GPL that helps in creating and managing a private or even a publicly accessible cloud. It provides an EC2-compatible cloud computing platform and S3-compatible cloud storage platform. Eucalyptus has become very popular and is seen as one of the key open source cloud platforms. Since Eucalyptus makes its services available through EC2/S3 compatible APIs, the client tools written for AWS can be used with Eucalyptus as well.

## 1.3 Ubuntu Enterprise Cloud (UEC)

Ubuntu Enterprise Cloud, UEC for short, is a stack of applications from Canonical included with Ubuntu Server Edition. UEC includes Eucalyptus along with a number of other open source software. UEC makes it very easy to install and configure the cloud. Canonical also provides commercial technical support for UEC.

## 1.4 Components of a Eucalyptus (UEC) based Cloud

### 1.4.1 Node Controller (NC)

A UEC node is a VT enabled server capable of running KVM as the hypervisor. UEC automatically installs KVM when the user chooses to install the UEC node. The VMs running on the hypervisor and controlled by UEC are called “instances”. Eucalyptus supports other hypervisors like Xen apart from KVM, but Canonical has chosen KVM as the preferred hypervisor for UEC.

Node Controller runs on each node and controls the life cycle of instances running on the node. The NC interacts with the OS and the hypervisor running on the node on one side and the Cluster Controller (CC) on the other side.

NC queries the Operating System running on the node to discover the node’s physical resources – the number of cores, the size of memory, the available disk space and also to learn about the state of VM instances running on the node and propagates this data up to the CC.

**Functions:**

- Collection of data related to the resource availability and utilization on the node and reporting the data to CC
- Instance life cycle management

### 1.4.2 Cluster Controller (CC)

CC manages one or more Node Controllers and deploys/manages instances on them. CC also manages the networking for the instances running on the Nodes under certain types of networking modes of Eucalyptus. More on this in later chapters.

CC communicates with Cloud Controller (CLC) on one side and NCs on the other side.

**Functions:**

- To receive requests from CLC to deploy instances
- To decide which NCs to use for deploying the instances on
- To control the virtual network available to the instances
- To collect information about the NCs registered with it and report it to the CLC

### 1.4.3 Walrus Storage Controller (WS3)

WS3 provides a persistent simple storage service using REST and SOAP APIs compatible with S3 APIs.

**Functions:**

- Storing the machine images
- Storing snapshots
- Storing and serving files using S3 API

WS3 should be considered as a simple file storage system.

### 1.4.4 Storage Controller (SC)

SC provides persistent block storage for use by the instances. This is similar to the Elastic Block Storage (EBS) service from AWS.

**Functions:**

- Creation of persistent EBS devices
- Providing the block storage over AoE or iSCSI protocol to the instances
- Allowing creation of snapshots of volumes

### 1.4.5 Cloud Controller (CLC)

The Cloud Controller (CLC) is the front end to the entire cloud infrastructure. CLC provides an EC2/S3 compliant web services interface to the client tools on one side and interacts with the rest of the components of the Eucalyptus infrastructure on the other side. CLC also provides a web interface to users for managing certain aspects of the UEC infrastructure.

**Functions:**

- Monitor the availability of resources on various components of the cloud infrastructure, including hypervisor nodes that are used to actually provision the instances and the cluster controllers that manage the hypervisor nodes
- Resource arbitration – Deciding which clusters will be used for provisioning the instances
- Monitoring the running instances

In short, CLC has a comprehensive knowledge of the availability and usage of resources in the cloud and the state of the cloud.

## 1.5 Architecture

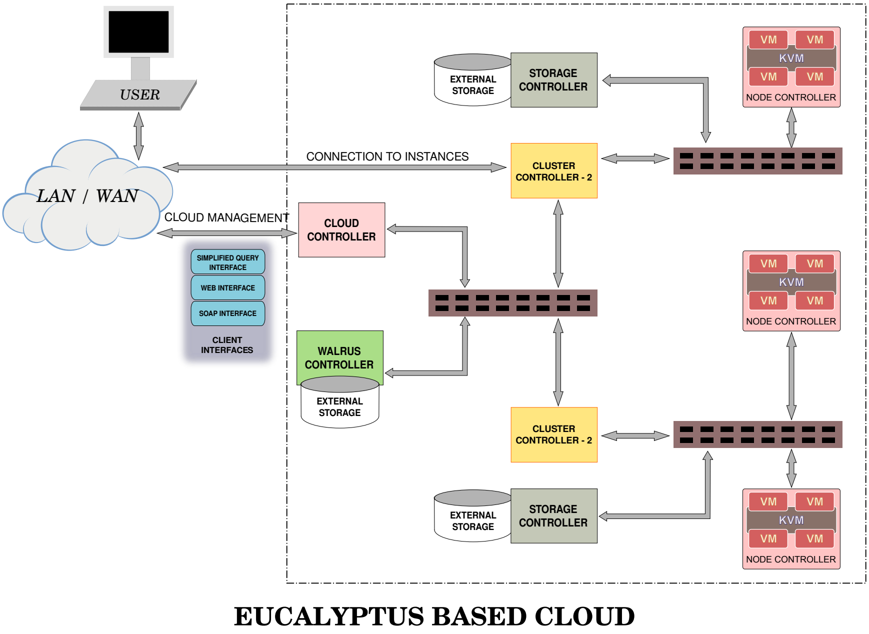


Figure 1.1: Components of UEC

# Chapter 2

# Installation & Configuration

The following sections describe how to install and configure a basic UEC setup spanning three servers. Two Servers (Server1 and Server2) will run a Maverick 64-bit server version and the third server will run a Maverick Desktop 64-bit version (Client1). We have decided on installing the Desktop version on Client1 so that we can use Firefox or other browsers to access the web interface of UEC and also to use RDP/VNC clients during the image creation process.

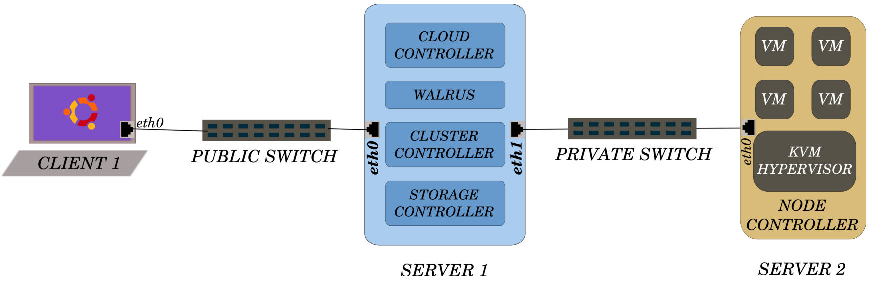


Figure 2.1: UEC basic setup with three machines

What we have used as a sample setup is a typical minimal Eucalyptus setup. Client1 above need not be a dedicated machine. If you need to setup Eucalyptus on a single server, you can refer to UEC: CC and NC on a single machine . However, such a setup will not support Managed and Managed-NOVLAN modes of networking.

| Hardware | Server 1 | | Server 2 | | Client 1 | |
|---|---|---|---|---|---|---|
| | Minimum | Suggested | Minimum | Suggested | Minimum | Suggested |
| **CPU** | 1GHz | 2 x 2GHz | VT extensions | VT, 64Bit, Multicore | VT extensions | VT, 64Bit, Multicore |
| **Memory** | 1 GiB | 2 GiB | 1 GiB | 4 GiB | 1 GiB | 2 GiB |
| **Disk** | 5400rpm IDE | 7200rpm SATA | 5400rpm IDE | 7200rpm SATA or SCSI | 5400rpm IDE | 7200rpm SATA or SCSI |
| **Disk Space** | 40 GiB | 200 GiB | 40 GiB | 100 GiB | 40 GiB | 100 GiB |
| **Networking** | 100 Mbps | 1000 Mbps | 100 Mbps | 1000 Mbps | 100 Mbps | 1000 Mbps |

Table 2.1: Suggested Hardware for a Eucalyptus Cloud Infrastructure

Note: UEC installation is very similar to the installation of Ubuntu server, the only difference is the additional configuration screens for the UEC components. The following instructions only cover the UEC-specific options as it is assumed that you are familiar with the installation of Ubuntu server.

For ease of understanding, the instructions here in this chapter and the rest of the book assume the following sample setup. Please make necessary modifications to suit your setup.

| | Server 1 | Server 2 | Client |
|---|---|---|---|
| | **Common username to login: ”uecadmin”** | | |
| **Functionality** | CLC, CC and Walrus | NC | Bundling and Web UI Client |
| **No. of NICs** | eth0 - (Enterprise N/W)<br>eth1 - (Eucalyptus N/W) | eth0 - (Eucalyptus N/W) | eth0 - (Enterprise N/W)<br>eth1 - (Eucalyptus N/W) |
| **IP Addresses** | eth0 - 192.168.10.121 (set during install)<br>eth1 - 192.168.20.1 (set after install) | eth0 - 192.168.20.2<br>(set during install) | eth0 - ???<br>eth1 - ??? |
| **Hostname** | server1.example.com | server2.example.com | client1.example.com |
| **Enterprise Name Servers** | 192.168.10.2<br>192.168.10.3 | 192.168.10.2<br>192.168.10.3 | 192.168.10.2<br>192.168.10.3 |
| **Gateway IP** | 192.168.10.100 | 192.168.20.1 | 192.168.10.2<br>192.168.10.3 |

Table 2.2: Comparison of Server1, Server2 and Client

- The gateway for Server2 is set to the IP of the CC – IP 192.168.20.1. This will enable the Server2 to connect to the enterprise network through Server1 (CC).
- Server1 is a 64-bit server and Server2 is a 64-bit VT-enabled server.
- The enterprise network runs on a class “C” private network – 192.168.10.0/255.255.255.0
- IP addresses allocated for the cloud instances as public IP addresses: 192.168.10.200-192.168.10.220 (Enterprise range)
- The UEC components will use a dedicated class “C” private network for their interconnections – 192.168.20.0/255.255.255.0
- Cluster Name – “myueccluster”

## 2.1 Server1

### 2.1.1 Installation

- Boot the server off the Ubuntu Server 10.10 64 bit CD. At the graphical boot menu, select “Install Ubuntu Enterprise Cloud” and proceed with the basic installation steps.

- Installation only lets you set up the IP address details for one interface. Please do that for eth0.

- You will need to choose certain configuration options for your UEC, during the course of the install:

- Cloud Controller Address – Leave this blank as Server1 is the Cloud Controller in this setup.

- Cloud Installation Mode – Select “Cloud controller”, “Walrus storage service”, “Cluster controller” and “Storage controller”.

- Network interface for communication with nodes – eth1

- Eucalyptus cluster name – myueccluster

- Eucalyptus IP range – 192.168.10.200-192.168.10.220

### 2.1.2 Post Install Steps & Configuration

- Set up static IP for the second network interface eth1 by adding the following to /etc/network/interfaces:

```
auto eth1
iface eth1 inet static
address 192.168.20.1
netmask 255.255.255.0
network 192.168.20.0
broadcast 192.168.20.255
```

- After making the above changes, run the following command to restart the networking:

```
uecadmin@server1:~$sudo /etc/init.d/networking restart
```

- Make sure that you have the latest version of Eucalyptus:

```
uecadmin@server1:~$sudo apt−get update
uecadmin@server1:~$sudo apt−get upgrade eucalyptus
```

- Install the NTP package. This machine is going to act as an NTP server for the nodes. The time on all components of UEC will have to be in sync. Since CLC is connected to internet, we can run NTP server on it and have other components sync to it:

```
uecadmin@server1:~$sudo apt−get install ntp
```

- Open the file /etc/ntp.conf and add the following two lines to make sure the server serves time even when its connectivity to the internet is down. The following settings makes sure that the NTP server uses its own clock as the clock source:

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

- Restart NTP service to make the changes effective:

```
uecadmin@server1:~$sudo /etc/init.d/ntp restart
```

- Save the changes and restart CC:

```
uecadmin@server1:~$sudo restart eucalyptus−cc CLEAN=1
```

## 2.2 Server2

### 2.2.1 Installation

- Boot the server off the Ubuntu Server 10.10 64 bit CD. At the graphical boot menu, select “Install Ubuntu Enterprise Cloud” and proceed with the basic installation steps.

- Installation only lets you set up the IP address for one interface. Please do that for eth0 by setting up the private IP – 192.168.20.2

- You will need to choose certain configuration options for your UEC, during the course of the install. You can ignore all the settings, except the following:

- Cloud Controller Address – 192.168.20.1
- Cloud Installation Mode – Select “Node Controller”
- Gateway – 192.168.20.1 (IP of the CC)

### 2.2.2 Post Install Steps & Configuration

- Add your DNS server details to /etc/resolv.conf:

```
nameserver 192.168.10.2
nameserver 192.168.10.3
```

- Make sure that you have the latest version of Eucalyptus:

```
uecadmin@server2:~$sudo apt−get update
uecadmin@server2:~$sudo apt−get upgrade eucalyptus
```

- Install the NTP package to synchronize with the NTP server:

```
uecadmin@server2:~$sudo apt−get install ntp
```

- Open the file /etc/ntp.conf and append the following line:

```
server 192.168.20.1
```

- Restart NTP service to make the changes effective:

```
uecadmin@server2:~$sudo /etc/init.d/ntp restart
```

- Open the file /etc/eucalyptus/eucalyptus.conf and make the following changes:

```
VNET_PUBINTERFACE=”br0”
VNET_PRIVINTERFACE=”br0”
VNET_BRIDGE=”br0”
VNET_DHCPDAEMON=”/usr/sbin/dhcpd3”
VNET_DHCPUSER=”dhcpd”
VNET_MODE=”MANAGED−NOVLAN”
```

- After making the above changes, run the following commands:

```
uecadmin@server2:~$sudo restart eucalyptus−nc
```

### 2.2.3 Install CC"s ssh public key to NC

- On the Node Controller, temporarily set a password for the "eucalyptus" user:

```
uecadmin@server2:~$sudo passwd eucalyptus
```

- On the Cluster Controller:

```
uecadmin@server2:~$sudo −u eucalyptus ssh−copy−id −i
~eucalyptus/.ssh/id_rsa.pub eucalyptus@192.168.20.2
```

- You can now remove the password of the "eucalyptus" account on the Node:

```
uecadmin@server2:~$sudo passwd −d eucalyptus
```

## 2.3 Client1

The purpose of Client1 machine is to interact with the cloud setup, for bundling and registering new Eucalyptus Machine Images (EMI)

### 2.3.1 Installation

- Boot the Desktop off the Ubuntu Desktop 10.10 64 bit CD and install. The Desktop will be on the enterprise network and will obtain an IP address through DHCP.
- Install KVM to help us install images on KVM platform and bundle them:

```
uecadmin@client1:~$apt−get install qemu−kvm
```

### 2.3.2 Post Install Configuration

- Install euca2ools to be able to manage the cloud from it:

```
uecadmin@client1:~$sudo apt−get install euca2ools
```

- Login to the web interface of CLC by using the following link https://192.168.10.121:8443. The default username is “admin” and the default password is “admin”.
- Note that the installation of UEC installs a self signed certificate for the web server. The browser will warn you about the certificate not having been signed by a trusted certifying authority. You can authorize the browser to access the server with the self signed certificate. Instruction on how to install a valid certificate signed by a trusted certifying authority is beyond the scope of this guide.
- When you login for the first time, the web interface prompts you to change the password and provide the email ID of the admin. After completing this mandatory step, download the credentials archive from https://192.168.10.121:8443/#credentials and save it in the ~/.euca directory.
- Extract the credentials archive:

```
$ cd .euca
$ unzip mycreds.zip
```

- Source eucarc script to make sure that the environmental variables used by euca2ools are set properly.
- To verify that euca2ools are able to communicate with the UEC, try fetching the local cluster availability details:

```
$ . ~/.euca/eucarc
$ euca—describe—availability—zones verbose
```

You should see something like the following if it works properly:

```
$ euca—describe—availability—zones verbose
 AVAILABILITYZONE myueccluster 192.168.10.121
AVAILABILITYZONE |— vm types free / max cpu ram disk
AVAILABILITYZONE |— m1.small 0002 / 0002 1 192 2
AVAILABILITYZONE |— c1.medium 0002 / 0002 1 256 5
AVAILABILITYZONE |— m1.large 0001 / 0001 2 512 10
AVAILABILITYZONE |— m1.xlarge 0001 / 0001 2 1024 20
AVAILABILITYZONE |— c1.xlarge 0000 / 0000 4 2048 20
```

- If you see the free/max VCPUs as 0 in the above list, it means the node did not get registered automatically. Please use the following on Server1 and approve when prompted to add 192.168.20.2 as the Node Controller:

```
uecadmin@client1:~$sudo euca_conf --discover-nodes
```

## 2.4 Virtio

Eucalyptus 2.0 bundled with UEC on Ubuntu 10.10 supports virtio. virtio is a framework which allows the instances to interact directly with the native hardware using paravirtualiztion for enhanced performance. In this case, the instance is aware of the virtualized environment and the guest OS needs to use suitable drivers to interact with the paravitualized devices. Drivers are currently available for most of the recent versions of Linux, Windows 7 and Windows 2008. If you want to run OSes that do not have virtio drivers, disable virtio in the configuration.

### 2.4.1 Configuration

Virtio can be configured by modifying the /etc/eucalyptus/eucalyptus.conf on the Node Controller as below.

- To enable Virtio for the EBS volumes attached from the SC:

```
USE_VIRTIO_DISK=”1”
```

- To enable Virtio for the root hard drive:

```
USE_VIRTIO_ROOT=”1”
```

- To enable Virtio for the Network Interface:

```
USE_VIRTIO_NET=”1”
```

# Chapter 3

# Web interface

Ubuntu Enterprise Cloud provides a minimal web interface to manage user credentials, manage images, administer user accounts, etc.

In our sample setup, the web interface is accessible at:

```
https://192.168.10.121:8443/
```

You have already finished setting up the password for the admin user of the web interface while working on Desktop1 in the chapter "Installation & Configuration".

A user trying to access the interface for the first time, will need to sign up and would be able to use the interface after the account is approved by the administrator. The approval of the account by the administrator results in an email to the user.

## 3.1 Using the Web Interface

After logging into the web interface, you will see seven tabs as given below:

1. Credentials
2. Images
3. Store
4. Users
5. Configuration

6. Services
7. Extras

## 3.2 Credentials

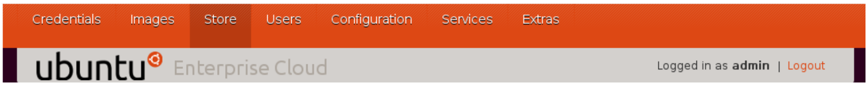


Figure 3.1: Web UI Tabs

You can change your account password, download the credentials (to be used while bringing up instances etc.) and view the query interface credentials (to be used with tools like euca2ools to query CC for information like getting the list of images, instances etc.) here. Later chapters have more detailed discussion about use of these credentials.

## 3.3 Images

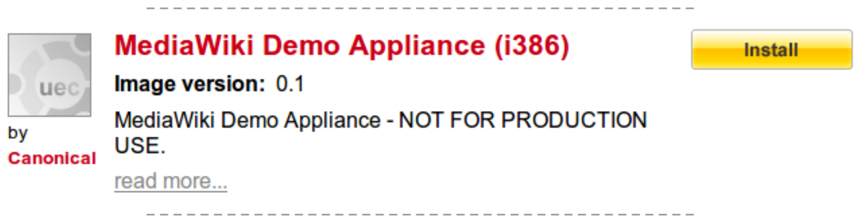
You can see here a list of images that have been uploaded to the CLC. Each entry contains information about the image such as its ID, meta information like the Ids of associated kernel and ramdisk images etc. This information comes in handy when you try to bring up instances using these images. This is mostly a read-only listing, except that you can disable/enable each of the images.

## 3.4 Store

The store tab lists new images available on the UEC image store of Canonical. These are provided by Canonical and third-party providers such as M/Gateway. You can install any of these images. Installation here refers to the task of downloading and making it available for use on the local installation of UEC. After installation, the image should now show up under the Images tab along with its kernel and ramdisk images.

## 3.5 Users

As an administrator, you can create as well as view and manage user accounts here including enabling/disabling accounts and changing the passwords.



MediaWiki Demo Appliance (i386)

**Image version:** 0.1

MediaWiki Demo Appliance - NOT FOR PRODUCTION USE.

read more...

Figure 3.2: Image Store

## 3.6 Configuration

You can view and manage information related to various components of your UEC setup, such as CLC, CC, Walrus etc. You can also specify the resource limits for different VM types such as m1.small, c1.medium etc. More on this the chapter on “Instance management”

## 3.7 Services

A few pieces of commercial information about services from Canonical and their partners for UEC.

## 3.8 Extras

You can find a list of Eucalyptus-certified Images and Eucalyptus-compatible Tools like euca2ools.

# Chapter 4

# Image management

## 4.1 Eucalyptus Machine Image (EMI)

A Eucalyptus Machine Image (EMI) is a combination of a virtual disk image(s), kernel and ramdisk images as well as an xml file containing metadata about the image. These images reside on WS3 and are used as templates for creating instances on UEC. Each Linux EMI is a combination of the following:

- An XML file with a name like "jaunty.img.manifest.xml" with information about one or more hard disk images, a kernel image and a ramdisk image (id – emi-65440E7E)
- An XML file with a name like "vmlinuz-2.6.28-11-server.manifest.xml" with information about the corresponding kernel image (id – eki-39FC1244)
- An XML file with a name like "initrd.img-2.6.28-11-server.manifest.xml" with information about the corresponding ramdisk image (id – eri-71ED1322)

Each of these images has its own ID that can be used while running the instances. More on this in the chapter on "Managing Instances"

From the web interface of Eucalyptus, you can view a list of EMIs in the "Store" tab of the web UI. These are the EMIs listed from Canonical or partners. You can choose to download and install any of these images directly from Canonical's site.

Since most enterprise/individual users of Eucalyptus have a need for bringing up instances based on custom images, image management plays a key role in Eucalyptus administration. Such images could be based on a preferred version/variant of a preferred OS distribution with a set of required applications pre-installed.

Bundling an EMI is a multi-step process involving the following:

- creating a virtual disk image
- installing the OS
- installing required applications
- making the OS ready to run under UEC
- registering the images with UEC
- testing the image etc.

Bundling process differs between Linux and Windows images and we will discuss the process in both cases in some detail below.

## 4.2 Bundling Linux Image

We will use Client1 for the purpose of working on creation of the image. Please note that we had installed KVM on Client1.

### 4.2.1 Creating a new disk Image

This will represent the main HDD of the virtual machine, so make sure to give it as much space as you will need.

```
uecadmin@client1:~$ kvm-img create -f raw image.img 1.5G
```

### 4.2.2 OS Installation

Download the iso file of the Linux distribution you want installed in the image. The example below refers to creating a Maverick Meerkat 64-bit server image:

```
uecadmin@client1:~$ wget
http://releases.ubuntu.com/10.10/ubuntu-10.10-server-amd64.iso
```

And start the installation process:

```
uecadmin@client1:~$ sudo kvm -m 256 -cdrom ubuntu-10.10-server-amd64.iso
-drive file=image.img,if=scsi,index=0 -boot d -net nic -net user
-nographic -vnc :0
```

If your installation process requires more than 256MB of RAM change the -m option, and if you need more processors available, you can use the “-c” option.

The command above will boot a new KVM instance, with the disk image you’ve created as the primary HDD and the iso as the first bootable device. Also the “-nographic” option will not display any graphical output. You can connect to the instance through VNC (use display number :0) and finish the installation. For example:

```
vncviewer A.B.C.D :0, where A.B.C.D is the IP address of the Client1.
```

After finishing the installation, relaunch the VM by executing the following command:

```
uecadmin@client1:~$ sudo kvm -m 256 -drive
file=image.img,if=scsi,index=0,boot=on -boot c -net nic -net user
-nographic -vnc :0
```

At this point you can add all the packages you want to have installed, update the installation, add users and any settings that need to be present in your new UEC instances.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install mediawiki
```

### 4.2.3 Integrating with Eucalyptus

An instance running under Eucalyptus needs to know what IP it has and also, it needs to have the public key of the user allowed to do a passwordless access through SSH. This is accomplished by using a restful interface provided by the cloud. The interface is available under this URL: http://169.254.169.254/latest/meta-data and is accessible from within the instance.

Eucalyptus node controller is set up to prevent automatic key injection if the system is in MANAGED or MANAGED-NOVLAN mode. Instead, it is assumed that the instance will use the above metadata service to retrieve the public keys when running in these modes. You will need to facilitate this by installing curl and adding a script that will run on each boot.

Install curl on the VM:

```
$ sudo apt-get install curl
```

Now add the following lines to /etc/rc.local of the image:

```
depmod -a
modprobe acpiphp
# simple attempt to get the user ssh key using the meta-data service
# assuming ?user? is the username of an account that has been created
mkdir -p /home/<username>/.ssh
echo >> /home/<username>/.ssh/authorized_keys
curl -m 10 -s
http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key |
grep 'ssh-rsa' >> \
                          /home/<username>/.ssh/authorized_keys
echo "AUTHORIZED_KEYS:"
echo "************************"
cat /home/<username>/.ssh/authorized_keys
echo "************************"
```

Add the above lines before the “exit 0” in /etc/rc.local

Copy the kernel and the initrd image from the VM image to some place outside. These will be used later for creating and uploading a complete virtual image to Eucalyptus:

```
$ scp /boot/initrd.img-2.6.35-22-server user@A.B.C.D:
$ scp /boot/vmlinuz-2.6.35-22-server user@A.B.C.D:
```

Also remove the network persistent rules from /etc/udev/rules.d, so that the instance always comes up with eth0 as the interface name as expected by Eucalyptus:

```
$ sudo rm -rf /etc/udev/rules.d/70-persistent-net.rules
```

Shut down the VM.

This completes the process of customizing the OS installed as a reference image.

### 4.2.4 Registering with Eucalyptus

The last step is to upload the images to Eucalyptus. The files that need to be uploaded for the above sample setup are: vmlinuz-2.6.28-11-server, initrd.img-2.6.28-11-server, image.img.

**Registering kernel image** Execute the following commands to bundle and register the kernel image (vmlinuz-2.6.35-22-server):

```
uecadmin@client1:~$ euca-bundle-image -i vmlinuz-2.6.35-22-server
--kernel true
uecadmin@client1:~$ euca-upload-bundle -b mybucket -m
/tmp/vmlinuz-2.6.35-22-server.manifest.xml
uecadmin@client1:~$ euca-register
mybucket/vmlinuz-2.6.35-22-server.manifest.xml
```

Save the output produced by the last command above (eki-XXXXXXXX), which will be needed while registering the disk image.

**Registering ramdisk image** Execute the following commands to bundle and register the ramdisk image (initrd.img-2.6.35-22-server):

```
uecadmin@client1:~$ euca-bundle-image -i initrd.img-2.6.35-22-server
--ramdisk true
uecadmin@client1:~$ euca-upload-bundle -b mybucket -m
/tmp/initrd.img-2.6.35-22-server.manifest.xml
uecadmin@client1:~$ euca-register
mybucket/initrd.img-2.6.35-22-server.manifest.xml
```

Save the output produced by the last command above (eri-XXXXXXXX), which will be needed while registering the disk image.

**Registering disk image** Execute the following commands to bundle and register the ramdisk image (image.img):

```
uecadmin@client1:~$ euca-bundle-image -i image.img --kernel eki-XXXXXXXX
--ramdisk eri-XXXXXXXX
uecadmin@client1:~$ euca-upload-bundle -b mybucket -m
/tmp/image.img.manifest.xml
uecadmin@client1:~$ euca-register mybucket/image.img.manifest.xml
```

Replace eki-XXXXXXXX and eri-XXXXXXXX with the exact values you have saved earlier.

### 4.2.5 Image Listing

The new images that have been uploaded can be seen by using euca-describe-images command. For example:

```
uecadmin@client1:~$ euca-describe-images
IMAGE emi-70B70EC0 mybucket/image.img.manifest.xml
```

```
   admin available public x86_64 machine
IMAGE eri−A2BE13EC
mybucket/initrd.img−2.6.35−22−server.manifest.xml admin available
     public x86_64 ramdisk
IMAGE eki−685F1306 mybucket/vmlinuz−2.6.35−22−server.manifest.xml
admin available public x86_64 kernel
```

More details on managing the instances in later chapters.

## 4.3 Bundling Windows Image

Bundling Windows images for UEC is not straight forward. Please refer to Appendix A for a hack to get Windows images running on UEC on Maverick Meerkat.

### 4.3.1 Creating new disk image

This will represent the main HDD of the virtual machine, so make sure to give it as much space as you will need.

```
uecadmin@client1:˜$ kvm−img create −f raw win−2k3.img 9G
```

### 4.3.2 OS Installation

Create an ISO image of the relevant Windows installation CD. And start the installation process:

```
uecadmin@client1:˜$ sudo kvm −m 1024 −cdrom Win2003ServerR2x86_cd1.iso
−drive file=win−2k3.img,if=scsi,boot=on −nographic −vnc :0
```

After finishing the installation and in case you need to install from the second CD as well, reboot the VM and launch the VM by the following command:

```
uecadmin@client1:˜$ sudo kvm −m 1024 −boot c −cdrom
Win2003ServerR2x86_cd2.iso −drive file=win−2k3.img,if=scsi,boot=on
−nographic −vnc :0
```

Once installation is complete, we need to create the boot disk needed for Windows 2003 server.

### 4.3.3 For Windows XP bundling

While bundling Windows XP, the OS doesn't recognize SCSI driver during installation process. The following workaround is suggested:

1. Attach the disk as IDE disk:

```
$ sudo kvm −m 1024 −cdrom winxpcd.iso −drive file=winxp.img,boot=on
−nographic −vnc :0
```

2. Finish the installation of Windows XP.

3. Now create another disk image using kemu-img:

```
$ kvm−img create −f raw newdisk.img 2G
```

4. Start the instance by attaching the second disk as SCSI disk.

```
$ sudo kvm −m 1024 −drive file=winxp.img,boot=on −drive file=newdisk.img
,if=scsi −nographic −vnc :0
```

5. After Windows XP boots, it detects and installs the driver for the SCSI disk (sys_hi.sys)

The remaining procedure is same as that of bundling Windows 2003 Server.

### 4.3.4 Copying the boot files

Re-launch the VM using the following command, so that we can access the first CD, in order to copy some files from it:

```
uecadmin@client1:˜$ sudo kvm −m 1024 −boot c −cdrom
Win2003ServerR2x86_cd1.iso −drive file=win−2k3.img,if=scsi,boot=on
−nographic −vnc :0
```

Copy the following files to a specific location to create the boot disk, say C:\bootfiles

- ntldr
- ntdetect.com
- sym_hi.sys

Rename sym_hi.sys to Ntbootdd.sys (SCSI driver needed to boot the hard disk).

In case, sym_hi.sys is not present in the CD, copy it from the following location of the installation:

```
C:\WINDOWS\system32\drivers
```

Write the following lines to a new file and save it as Boot.ini in the same location, i.e, C:\bootfiles:

```
[boot loader]
timeout=30
Default= multi(0)disk(0)rdisk(0)partition(1)\Windows
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS=”Windows Server 2003”
```

Shut down the VM and proceed for the boot disk creation.

### 4.3.5 Boot disk creation

Execute the following command to create the boot disk:

```
uecadmin@client1:~$ dd bs=512 count=2880 if=/dev/zero of=win-boot.img
```

Now attach this disk as a floppy disk and start the Windows VM:

```
uecadmin@client1:~$ sudo kvm -m 1024 -boot c -drive
file=win-2k3.img,if=scsi,boot=on,index=1 -fda win-boot.img -nographic
-vnc :0
```

On the Windows VM, format the floppy disk and copy all the files from C:\bootfiles to the floppy. Shutdown the VM and test the boot disk created with the following command:

```
uecadmin@client1:~$ sudo kvm -m 1024 -boot a -drive
file=win-2k3.img,if=scsi,boot=on,index=1 -fda win-boot.img -nographic
-vnc :0
```

This should boot you into Windows, if you have created the boot floppy right.

### 4.3.6 Network Configuration

Now you need to download the e1000 drivers, since e1000 is the default interface that is supported by Eucalyptus. You can download the drivers from the following location:

http://downloadcenter.intel.com/Detail_Desc.aspx?agr=Y&DwnldId=8659&lang=ENG

```
For 32−bit VM, download PRO2KXP.exe
For 64−bit Vm, download PROEM64T .ex
```

Once the download is finished, shutdown the VM and restart the VM with the following command:

```
uecadmin@client1:˜$ sudo kvm −m 1024 −boot a −drive
file=win−2k3.img,if=scsi,boot=on,index=1 −fda win−boot.img −net
nic,model=e1000 −net tap −nographic −vnc :0
```

The above command tells the VM to make use of e1000 model as its NIC. After executing the above command, you will get this warning:

```
Warning: vlan 0 is not connected to host network
```

You can safely ignore this warning message.

After the VM boots, install the network drivers.

Enable remote desktop for all users on the VM. This is very important. This is the only way, without installing a third party tool, you can allow access to a Windows instance running on UEC. You can enable remote desktop by navigating to Start → Settings → Control Panel → System → Remote and selecting Remote Desktop option for all users. This completes the Windows disk setup. Shut down the VM.

### 4.3.7 Memdisk

The kernel memdisk is needed to load Windows. The default syslinux memdisk can be used for this purpose. If you cannot locate memdisk on CC, you need to install syslinux:

```
uecadmin@client1:˜$ sudo apt−get install syslinux
```

After installing, memdisk will be found in /usr/lib/syslinux:

```
uecadmin@client1:~$ cd /usr/lib/syslinux
uecadmin@client1:/usr/lib/syslinux$ ls -l memdisk
-rw-r--r-- 1 root root 20068 2008-07-15 20:25 memdisk
```

Copy it to the location where you have all the Windows files.

### 4.3.8 Registering with Eucalyptus

The last step is to upload the images to UEC. The files that need to be uploaded are:

- memdisk
- win-boot.img
- win-2k3.img

**Registering memdisk** Execute the following commands to bundle and register the memdisk:

```
uecadmin@client1:~$ euca-bundle-image -i memdisk --kernel true
uecadmin@client1:~$ euca-upload-bundle -b win2k3_bucket -m
/tmp/memdisk.manifest.xml
uecadmin@client1:~$ euca-register win2k3_bucket/memdisk.manifest.xml
```

Save the output produced by the last command above (eki-XXXXXXXX), which will be needed while registering the disk image.

**Registering boot disk** Execute the following commands to bundle and register the boot disk (win-boot.img):

```
uecadmin@client1:~$ euca-bundle-image -i win-boot.img --ramdisk true
uecadmin@client1:~$ euca-upload-bundle -b win2k3_bucket -m
/tmp/win-boot.img.manifest.xml
uecadmin@client1:~$ euca-register
win2k3_bucket/win-boot.img.manifest.xml
```

Save the output produced by the last command above (eri-XXXXXXXX), which will be needed while registering the disk image.

**Registering disk image** Execute the following commands to bundle and register the disk image (win-2k3.img):

```
uecadmin@client1:~$ euca-bundle-image -i win-2k3.img --kernel
eki-XXXXXXXX --ramdisk eri-XXXXXXXX
uecadmin@client1:~$ euca-upload-bundle -b win2k3_bucket -m
/tmp/win-2k3.img.manifest.xml
uecadmin@client1:~$ euca-register win2k3_bucket/win-2k3.img.manifest.xml
```

Replace eki-XXXXXXXX and eri-XXXXXXXX with the exact values noted down earlier.

### 4.3.9 Image Listing

The new images that have been uploaded can be seen by using euca-describe-images command. For example:

```
uecadmin@client1:~$ euca-describe-images
IMAGE emi-55470DE2 win-2k3/win-2k3.img.manifest.xml admin
available public x86_64 machine
IMAGE eki-2BC30D1B win-2k3/memdisk.manifest.xml admin
available public x86_64 kernel
IMAGE eri-76C20ED5 win-2k3/win-boot.img.manifest.xml admin
available public x86_64 ramdisk
```

### 4.3.10 Running instance of custom image

To launch a new instance of the custom image, execute euca-run-instances command:

```
uecadmin@client1:~$ cd .euca
uecadmin@client1:~/.euca$ euca-run-instances emi-55470DE2 -k mykey -t
c1.xlarge
```

Since this is a Windows instance the VM type is specified as c1.xlarge, which has 20GB of hard disk space.

# Chapter 5

# Instance management

There are broadly three types of tools that you can use to manage Eucalyptus instances.

- Firefox Plugins – Elasticfox and Hybridfox
- Command Line tools like euca2ools
- Custom applications developed using EC2 APIs. (Will not be covered in this beginner's guide)

## 5.1 Elasticfox

Elasticfox is an open source Mozilla Firefox extension that works on Firefox Version 2.0 or later to help you with managing your Amazon EC2 account: Launch new instances, mount Elastic Block Storage volumes, map Elastic IP addresses, and more. This was originally written for EC2, but, since version 1.7, Elasticfox added Eucalyptus support as well, because API of Eucalyptus is compatible with that of EC2.

### 5.1.1 Features

- List available Machine images (AMIs in the case of AWS and EMIs in the case of Eucalyptus)
- List running instances
- Launch new instances of an AMI/EMI
- Manage security groups and rules
- Manage Snapshots/EBS volumes

### 5.1.2 Installation on Firefox

Install the extension from http://s3.amazonaws.com/ec2-downloads/elasticfox.xpi

You will need to restart Firefox before you can start using Elasticfox. You can launch Elasticfox from "Tools" menu of Firefox.

### 5.1.3 Configuration

#### Define a Region

- Click on Regions button
- Enter a logical name for the region (Example: "Eucalyptus" , "MyEucalyptus" etc.)
- Enter the value of EC2_URL from your eucarc file as the Endpoint URL (http://192.168.10.121:8773/services/Eucalyptus in our sample setup)

#### Define Credentials

- Click on Credentials button
- Enter a logical name for the credential set (Example: "EucaAcc1 etc.)
- Enter EC2_ACCESS_KEY and EC2_SECRET_KEY from your eucarc file as the AWS Access Key and AWS Secret Access Key respectively

#### Define Key Pairs

- Click on KeyPairs tab
- Click on "Create a new keypair" icon
- Enter a name for the key pair (Example: "eucakey" etc.)
- Choose location on the client machine to save the ID file to (You will need this to use with putty etc. to make an SSH connection)

### Define Security Groups

- Click on SecurityGroups tab
- Enter a name for the group (Example: “Eucalyptus”, “EucaGroup” etc.)
- Enter the description and click on ”Create” button
- You will have an option to specify some basic ports like SSH/RDP to be opened up while creating the group itself. After the group is created, you can add rules any time by choosing the security group in the left pane titled “Your groups” and adding rules in the right pane titled “Group Permissions”

For more information,

http://ec2-downloads.s3.amazonaws.com/elasticfox-owners-manual.pdf

## 5.2 Hybridfox

Hybridfox was forked from Elasticfox to make it usable with Eucalyptus, when Elasticfox worked only with AWS. Hybridfox can be used as the single interface to AWS and Eucalyptus.

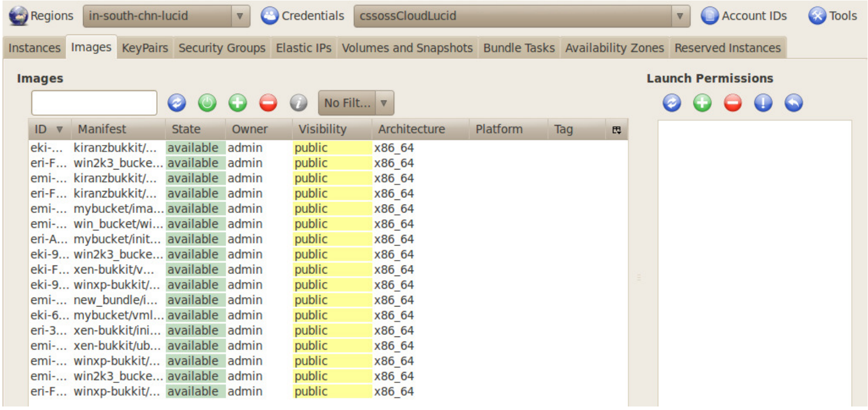
The interface of Hybridfox is similar to Elasticfox and all the above steps in configuring Elasticfox apply to Hybridfox as well.

Even after the recent release of Elasticfox (version 1.7) with support for Eucalyptus, Hybridfox has the following additional features that make it attractive for users of Eucalyptus:

- Raising instances with Private IP
- Some quirks related to hard coded mapping of instance types and architecture types addressed
- Support for Eucalyptus 1.5.x as well as 1.6.x
- Other usability enhancements

You can install the extension from http://code.google.com/p/hybridfox/downloads/list. The latest version (at the time of writing) is hybridfox-1.6.000030.xpi.

You will need to restart Firefox before you can start using Hybridfox. You can access Hybridfox from “Tools” menu of Firefox.



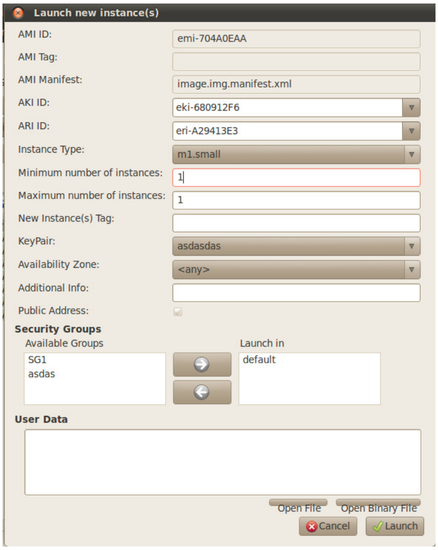
Regions: in-south-chn-lucid

Credentials: cssossCloudLucid

Images

| ID | Manifest | State | Owner | Visibility | Architecture | Platform | Tag |
|---|---|---|---|---|---|---|---|
| eki-... | kiranzbukkit/... | available | admin | public | x86_64 | | |
| eri-F... | win2k3_bucke... | available | admin | public | x86_64 | | |
| emi-... | kiranzbukkit/... | available | admin | public | x86_64 | | |
| eri-F... | kiranzbukkit/... | available | admin | public | x86_64 | | |
| emi-... | mybucket/ima... | available | admin | public | x86_64 | | |
| emi-... | win_bucket/wi... | available | admin | public | x86_64 | | |
| eri-A... | mybucket/init... | available | admin | public | x86_64 | | |
| eki-9... | win2k3_bucke... | available | admin | public | x86_64 | | |
| eki-F... | xen-bukkit/v... | available | admin | public | x86_64 | | |
| eki-9... | winxp-bukkit/... | available | admin | public | x86_64 | | |
| emi-... | new_bundle/i... | available | admin | public | x86_64 | | |
| eki-6... | mybucket/vml... | available | admin | public | x86_64 | | |
| eri-3... | xen-bukkit/ini... | available | admin | public | x86_64 | | |
| emi-... | xen-bukkit/ub... | available | admin | public | x86_64 | | |
| emi-... | winxp-bukkit/... | available | admin | public | x86_64 | | |
| emi-... | win2k3_bucke... | available | admin | public | x86_64 | | |
| eri-F... | winxp-bukkit/... | available | admin | public | x86_64 | | |

Figure 5.1: Images in HybridFox



| | |
|---|---|
| AMI ID: | emi-704A0EAA |
| AMI Tag: | |
| AMI Manifest: | image.img.manifest.xml |
| AKI ID: | eki-680912F6 |
| ARI ID: | eri-A29413E3 |
| Instance Type: | m1.small |
| Minimum number of instances: | 1 |
| Maximum number of instances: | 1 |
| New Instance(s) Tag: | |
| KeyPair: | asdasdas |
| Availability Zone: | <any> |
| Additional Info: | |
| Public Address: | ☑ |

Security Groups

Available Groups: SG1, asdas

Launch in: default

User Data

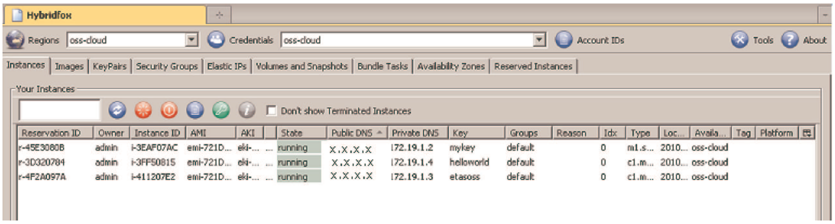
Figure 5.2: Launching an instance in HybridFox

### 5.2.1 Screenshots

## 5.3 Command Line Tools (Euca2ools)

Euca2ools from Eucalyptus provide a bunch of command line tools to manage the eucalyptus setup. These commands help you manage images, instances, storage, networking etc. A few commands related to managing the instances are given below.

For a complete list of commands, see Appendix.



Regions: oss-cloud

Credentials: oss-cloud

□ Don't show Terminated Instances

| Reservation ID | Owner | Instance ID | AMI | AKI | | State | Public DNS | Private DNS | Key | Groups | Reason | Idx | Type | Loc... | Availa... | Tag | Platform |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| r-45E3080B | admin | i-3EAF07AC | emi-721D... | eki-... | ... | running | X.X.X.X | 172.19.1.2 | mykey | default | | 0 | m1.s... | 2010... | oss-cloud | | |
| r-3D320784 | admin | i-3FF50815 | emi-721D... | eki-... | ... | running | X.X.X.X | 172.19.1.4 | helloworld | default | | 0 | c1.m... | 2010... | oss-cloud | | |
| r-4F2A097A | admin | i-411207E2 | emi-721D... | eki-... | ... | running | X.X.X.X | 172.19.1.3 | etasoss | default | | 0 | c1.m... | 2010... | oss-cloud | | |

Figure 5.3: Instances in HybridFox

To install eucac2ools execute the following command:

```
$ sudo apt-get install euca2ools
```

### 5.3.1 Creation of Key Pairs

Eucalyptus expects the client tools to use two kinds of credentials. One set of credentials are called Access Key and Secret Key that all clients need to use to make any requests to the Cloud Controller. Each user registered on the web interface has this set created for him. You can download it from the web interface as mentioned in the chapter on “Web interface”.

You will also need to generate a keypair consisting of private key/public key to be able to launch instances on Eucalyptus. These keys are injected into the instance and used to make passwordless SSH access to the instance possible.

Keypairs can also be generated using the following commands:

```
uecadmin@client1:~$ cd ~/.euca
uecadmin@client1:~/.euca$ source eucarc
uecadmin@client1:~$ euca-add-keypair mykey > mykey.priv
uecadmin@client1:~$ chmod 600 mykey.priv
```

This creates a new keypair called mykey. The private key mykey.priv is saved locally which can be used to connect to an instance launched with mykey as the keypair. To list the available keypairs:

```
uecadmin@client1:~$ euca-describe-keypairs
KEYPAIR mykey
f7:ac:8e:f5:05:19:2b:31:28:8c:9b:d7:b8:07:0c:3c:b6:34:8f:79
KEYPAIR helloworld
12:96:b3:21:34:8d:6a:3f:92:2e:2b:70:23:ff:7f:51:b5:b7:ad:37
KEYPAIR ubuntu
f6:af:9a:59:65:35:32:c4:3a:c4:62:0e:e1:44:0f:71:29:03:2d:91
KEYPAIR lucid
```

```
74:04:70:33:ed:57:7a:30:36:1f:ca:c6:ec:d5:4f:10:34:1a:52:51
KEYPAIR karmic
01:f9:aa:5f:4d:20:e2:53:d1:29:d0:0f:e2:f3:8c:21:91:70:7e:c8
```

To delete an existing keypair:

```
uecadmin@client1:~$ euca-delete-keypair helloworld
```

The above tasks can be achieved using Hybridfox from the “Keypairs” tab.

### 5.3.2 Launch and manage instances

The following commands can be used to launch an instance from the above generated keypair, check its status, connect to the instance etc.:

```
$ euca-run-instances emi-721D0EBA -k mykey -t c1.medium
RESERVATION r-55560977 admin admin-default
INSTANCE i-50630A2A emi-721D0EBA 0.0.0.0 0.0.0.0 pending
mykey 2010-05-07T07:17:48.23Z eki-675412F5 eri-A1E113E0
$ euca-describe-instances
RESERVATION r-55560977 admin default
INSTANCE i-50630A2A emi-721D0EBA 192.168.10.200
172.19.1.2 running mykey 0 c1.medium
2010-05-07T07:17:48.23Z myueccluster eki-675412F5
eri-A1E113E0
$ euca-get-console-output i-50630A2A
i-50630A2A
2010-05-07T07:22:40.795Z
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Linux version 2.6.32-21-server (buildd@yellow) (gcc
version 4.
4.3 (Ubuntu 4.4.3-4ubuntu5) ) #32-Ubuntu SMP Fri Apr 16 09:17:34 UTC
2010 (Ub
untu 2.6.32-21.32-server 2.6.32.11+drm33.2)
[ 0.000000] Command line: root=/dev/sda1 console=ttyS0
[ 0.000000] KERNEL supported cpus:
[ 0.000000] Intel GenuineIntel
[ 0.000000] AMD AuthenticAMD
[ 0.000000] Centaur CentaurHauls
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: 0000000000000000 - 000000000009f000 (usable)
[ 0.000000] BIOS-e820: 000000000009f000 - 00000000000a0000
(reserved)
............
............
............
uecadmin@client1:~$ ssh -i mykey.priv user@192.168.10.200
uecadmin@client1:~$ euca-reboot-instances i-50630A2A
uecadmin@client1:~$ euca-terminate-instances i-50630A2A
```

Make sure that you launch the instance with the correct VM type. If it is launched with a smaller VM type, then the following error is encountered:

```
error: insufficient disk capacity remaining
```

By default, m1.small is the VM type that is used which comes with 2GiB of disk space. So if you have a disk image of size 5GB, instead of

```
uecadmin@client1:˜$ euca—run—instances emi—XXXXXXXX —k mykey
```

use the following command to run the instance as a better suited VM type:

```
uecadmin@client1:˜$ euca—run—instances emi—XXXXXXXX —k mykey —t
c1.medium
```

VM type also has implications for amount of RAM and number of CPUs allocated to the instance. To know the configured VM types, please see the Configuration page of the web interface.

## 5.4 Life cycle of an instance

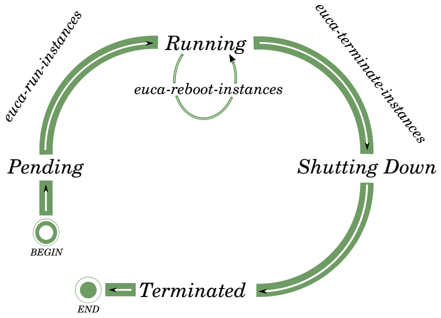


Figure 5.4: Instance Lifecycle

When you invoke “euca-run-instances” command (or when you choose to run an instance from Hybridfox/Elasticfox), here are a few things that happen on various components of UEC:

- Authentication/Authorization of the user request to ensure you have permission to launch the instance
- Identification of CC to take responsibility for deploying the instance and identification of the NC for running the instance.
- Downloading the image from WS3 to NC (images are cached so that starting multiple instances of the same machine image downloads that image only once)

- Creating the requested virtual network interface
- Starting the instance of the machine image running as a virtual machine using KVM

Starting and stopping of an instance is a complex process involving multiple components of UEC and a more detailed discussion is beyond the scope of a beginner's guide.

# Chapter 6

# Storage management

## 6.1 Storage Controller

Storage Controller is the component of Eucalyptus that provides persistent block storage compatible with Amazon's Elastic Block Store. euca2ools or EC2 commands can be used to manage these volumes. Here are a few euca commands for managing the volumes

## 6.2 Interacting with Storage Controller

### 6.2.1 Linux Instances

Create a 10 GB volume:

```
uecadmin@client1:˜$ euca−create−volume −s 10 −z mycloud
VOLUME vol−333C04B8 10 creating
2010−03−26T05:20:56.383Z
```

List the volumes:

```
uecadmin@client1:˜$ euca−describe−volumes
VOLUME vol−333C04B8 10 mycloud available
2010−03−26T05:20:56.383Z
```

Attach a volume to a running instance:

```
uecadmin@client1:˜$ euca−attach−volume −i i−41620887 −d /dev/sdb
vol−333C04B8
```

The volumes status changes from “available” to “in-use” if it is attached properly as shown in the output of “euca-describe-volumes” below:

```
uecadmin@client1:~$ euca−describe−volumes
VOLUME vol−333C04B8 10 mycloud in−use 2010−03−26T05:41:28.019Z
ATTACHMENT vol−331F04B2 i−41620887
unknown,requested:/dev/sdb 2010−03−26T05:43:51.343Z
```

You can login to the instance through SSH and see the new volume as a newly attached block storage, which shows up in the example below as /dev/sdb:

```
$ sudo fdisk −l
Disk /dev/sda: 2195 MB, 2195718144 bytes
4 heads, 32 sectors/track, 33504 cylinders
Units = cylinders of 128 ∗ 512 = 65536 bytes
Disk identifier: 0x000adc1c
 Device Boot Start End Blocks Id System
/dev/sda1 1 32769 2097152+ 83 Linux
/dev/sda3 32769 33504 47071 82 Linux swap /
Solaris
Disk /dev/sdb: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders
Units = cylinders of 2048 ∗ 512 = 1048576 bytes
Disk identifier: 0x00000000
Disk /dev/sdb doesn't contain a valid partition table
```

You can format /dev/sdb, mount it and use it as a persistent storage. It is better to unmount the volume before bringing down the instance, to avoid any data corruption.

Detach a volume:

```
uecadmin@client1:~$ euca−detach−volume vol−333C04B8
```

### 6.2.2 Windows Instances

For Windows instances, the procedure for creating the volume and attaching the volume is the same.

Note: Even while attaching to Windows instances, the device has to be specified as /dev/sdb.

The only change is the way this disk is accessed. Once the disk is attached to a running Windows instance, perform the following steps:

- Connect to Windows instance via RDP

- Go to Start -> Run and type diskmgmt.msc without the quotes. Select OK. This opens the Disk Management console
- Right click on the new disk at the bottom of the list and choose Initialize Disk Confirm initialization
- Right click on the volume and select "New Partition"
- Accept the defaults or modify to suit your setup and proceed with formatting
- After completion the disk management tool shows a healthy partition on the new drive.

To detach a volume from an instance use euca-detach-command command:

```
uecadmin@client1:~$ euca-detach-volume vol-333C04B8
```

Note that the volume has to unmounted before detaching it from the instance. Volume can be deleted using euca-delete-volume command:

```
uecadmin@client1:~$ euca-delete-volume vol-333C04B8
```

## 6.3 WS3 Storage Controller

Walrus is a storage service in Eucalyptus which is compatible with Amazon's S3. Using Walrus users can store persistent data, which is organized as buckets and objects. WS3 is a file level storage system, as compared to the block level storage system of Storage Controller.

See Amazon's S3 Getting Started Guide for more information http://docs.amazonwebservices.com/AmazonS3/2006-03-01/gsg/).

Walrus Controller options can be modified from the Web UI, on the "Configuration" page under "Walrus Configuration" section.

For using Walrus to manage Eucalyptus VM images, you can use Amazon's tools to store/register/delete them from Walrus. Other third party tools can also be used to interact with Walrus directly.

### 6.3.1 Third party tools for interacting with Walrus

- s3curl S3 Curl is a command line tool that is a wrapper around curl. http://open.eucalyptus.com/wiki/s3curl

- s3cmd is a tool that allows command line access to storage that supports the S3 API. http://open.eucalyptus.com/wiki/s3cmd

- s3fs is a tool that allows users to access S3 buckets as local directories. http://open.eucalyptus.com/wiki/s3fs

## 6.4 Snapshots

The volumes that have been created on the Block Storage can be used to create point-in-time snapshots of volumes, which are stored on WS3. These snapshots can be used as the starting point for new Block Storage volumes and protect data for long-term durability. The same snapshot can be used to create volumes when needed.

Snapshots can be created by using euca-create-snapshot command:

```
uecadmin@client1:~$ euca−create−snapshot vol−333C04B8
```

euca-describe-snapshots lists the available snapshots:

```
uecadmin@client1:~$ euca−describe−snapshots
SNAPSHOT snap−32A804A2 vol−333C04B8 completed
2010−04−15T13:48:32.01Z 100%
```

Volumes from these snapshots can be created by using the snapshot ID:

```
uecadmin@client1:~$ euca−create−volume −s 10 −−snapshot snap−32A804A2
−−zone mycloud
```

Please note that the volume can only be created after the snapshot status is completed. Similarly, snapshots from a volume can only be created after the volume status is created, but not during the creating stage. To delete a snapshot:

```
uecadmin@client1:~$ euca−delete−snapshot snap−32A804A2
```

# Chapter 7

# Eucalyptus Networking

## 7.1 System

In System mode, CC generates and assigns a random MAC address to the VM instance while requesting NC to bring up the instance. NC attaches the VM instance's virtual NIC to the physical NIC on the node through a bridge. This mode requires that the Nodes are connected to the enterprise network directly. Instances obtain an IP address using DHCP, just as physical machines on the network do.

This mode is very easy to setup as it does not have any additional prerequisites in terms of networking, except for a running DHCP server on enterprise network. It is a good way to get started with Eucalyptus, particularly if you want to set it up on your laptop/desktop to get a basic understanding.

This mode of networking is similar to "Bridged Networking" that hypervisors like VMware, VirtualBox etc. offer or like "tap" networking offered by KVM/Qemu.

## 7.2 Static

Static mode offers the Eucalyptus administrator more control over VM IP address assignment than System mode does. In this mode, the administrator configures Eucalyptus with a "map" of MAC address/IP address pairs on CC.

Before requesting NC to raise an instance, CC sets up a static entry within a Eucalyptus controlled DHCP server, takes the next free MAC/IP pair, and passes on to NC, which attaches the virtual NIC of the instance to the physical NIC of the Node through a bridge similar to how it is handled in System mode.

This mode of networking is similar to “Bridged Networking” that hypervisors like VMware, VirtualBox etc. offer or like “tap” networking offered by KVM/Qemu.

This mode is useful for administrators who have a pool of MAC/IP addresses that they wish to always assign to their instances without relying on the DHCP server running in the enterprise network.

Note – Running Eucalyptus in System or Static mode disables some of the following key functionalities that would make an enterprise deployment more manageable:

- Ingress filtering for the instances (Security Groups)
- User Controlled dynamic assignment of IPs to instances (Elastic IPs)
- Isolation of network traffic between instances VMs
- Availability of the metadata service (use of the http://169.254.169.254/ URL to obtain instance specific information)

## 7.3 Managed

Managed mode is the most feature-rich mode offered by Eucalyptus. In this mode, the Eucalyptus administrator defines a large network (usually private and unroutable) from which VM instances will draw their IP addresses. As with Static mode, CC will maintain a DHCP server with static mappings for each instance that is raised and allocate the right IPs at the time of requesting an NC to raise the instance.

Managed mode implements “security groups” for ingress filtering and isolation of instances. The user specifies a security group to which the new instance should be associated with, at the time of requesting a new instance. CC allocates a subset of the entire range of IPs to each security group in such a way that all the instances raised to be a part of the same security group use IPs from the same subset.

The user can define ingress filtering rules at the “security group” level. More on this in the chapter on Security. In addition, the administrator can specify a pool of public IP addresses that users may allocate, either while raising the instances or later at run-time. This functionality is similar to “elastic IPs” of AWS.

Eucalyptus administrators who need to implement require security groups, elastic IPs, and VM network isolation must use this mode.

## 7.4 Managed NOVLAN

This mode is identical to MANAGED mode in terms of features (dynamic IPs and security groups), but does not provide VM network isolation. Eucalyptus administrators who want dynamic assignable IPs and the security groups, but are not in a position to run on a network that allows VLAN tagged packets or those who do not have a need for VM network isolation can use this mode.

## 7.5 Comparison of Eucalyptus Networking Modes

| Sl. No. | Networking Type | DHCP Server running on the N/W? | CC runs own DHCP server? | Instance isolation | Private IPs | Ingress filtering |
|---|---|---|---|---|---|---|
| 1. | System | Required | No | No | No | No |
| 2. | Static | No | Yes | No | No | No |
| 3. | Managed | No | Yes | Yes | Yes | Yes |
| 4. | Managed - NOVLAN | No | Yes | No | Yes | Yes |

Table 7.1: Networking Types - Characteristics

## 7.6 Configurations

For the sake of convenience, the following network setup is assumed.

Managed and Managed-NOVLAN:

- CC – two interfaces eth0 and eth1. eth1 is connected to internet and eth0 is connected to NC.
- NC – one interface eth0, which is part of bridge br0 and this is connected to CC.

System and Static Mode

- CC – one interfaces eth0 connected to the enterprise network.
- NC – One interface eth0, which is part of bridge br0 connected to the enterprise network.

The following settings have to be made in /etc/eucalyptus/eucalyptus.conf file on CC and NC to configure the corresponding networking mode.

| PARAMETER | SYSTEM | | STATIC | | MANAGED-NOVLAN | | MANAGED | |
|---|---|---|---|---|---|---|---|---|
| | CLUSTER | NODE | CLUSTER | NODE | CLUSTER | NODE | CLUSTER | NODE |
| **VNET_PUBINTERFACE** | eth0 | br0 | eth0 | br0 | eth0 | br0 | eth0 | eth1 |
| **VNET_PRIVINTERFACE** | eth0 | br0 | eth0 | br0 | eth1 | br0 | eth1 | eth1 |
| **VNET_BRIDGE** | - | br0 | - | br0 | - | br0 | - | - |
| **VNET_DHCPDAEMON** | - | - | - | - | /usr/sbin/dhcpd | - | /usr/sbin/dhcpd | - |
| **VNET_DHCPUSER** | - | - | - | - | dhcpd | - | dhcpd | - |
| **VNET_MODE** | SYSTEM | SYSTEM | STATIC | STATIC | MANAGED-NOVLAN | MANAGED-NOVLAN | MANAGED | MANAGED |
| **VNET_SUBNET** | - | - | 192.168.10.0 | - | 172.1.0.0 | - | 172.1.0.0 | - |
| **VNET_NETMASK** | - | - | 255.255.255.0 | - | 255.255.255.0 | - | 255.255.255.0 | - |
| **VNET_BROADCAST** | - | - | 192.168.10.255 | - | - | - | - | - |
| **VNET_ROUTER** | - | - | 192.168.10.1 | - | - | - | - | - |
| **VNET_DNS** | - | - | - | - | 192.168.10.2 | - | 192.168.10.2 | - |
| **VNET_MACMAP** | - | - | AA:BB:CC:DD:DE:EE=192.168.10.200<br>AA:BB:CC:DD:DE:EF=192.168.10.201 | - | - | - | - | - |
| **VNET_ADDRSPERNET** | - | - | 32 | - | 32 | - | 32 | - |
| **VNET_PUBLICIPS** | - | - | - | - | 192.168.10.200<br>192.168.10.220 | - | 192.168.10.200<br>192.168.10.220 | - |
| **VNET_LOCALIP** | - | - | - | - | - | - | - | - |
| **VNET_CLOUDIP** | - | - | - | - | - | - | - | - |

Table 7.2: eucalyptus.conf - Settings

## 7.7 Commands

To list the range of public IP addresses allocated for instances:

```
uecadmin@client1:~$ euca-describe-addressess
ADDRESS 192.168.10.200 nobody
ADDRESS 192.168.10.201 nobody
ADDRESS 192.168.10.202 nobody
ADDRESS 192.168.10.203 nobody
ADDRESS 192.168.10.204 nobody
ADDRESS 192.168.10.205 nobody
ADDRESS 192.168.10.206 nobody
ADDRESS 192.168.10.207 nobody
ADDRESS 192.168.10.208 nobody
ADDRESS 192.168.10.209 nobody
ADDRESS 192.168.10.210 nobody
ADDRESS 192.168.10.211 nobody
ADDRESS 192.168.10.212 nobody
ADDRESS 192.168.10.213 nobody
ADDRESS 192.168.10.214 nobody
ADDRESS 192.168.10.215 nobody
ADDRESS 192.168.10.216 nobody
ADDRESS 192.168.10.217 nobody
ADDRESS 192.168.10.218 nobody
ADDRESS 192.168.10.219 nobody
ADDRESS 192.168.10.220 nobody
```

To allocate a public IP address for a specify user:

```
uecadmin@client1:~$ euca-allocate-address
ADDRESS 192.168.10.200
```

To associate a public IP address to a running instance:

```
uecadmin@client1:~$ euca-associate-address -i i-4799086D 192.168.10.200
```

To disassociate a public IP address from a running instance:

```
uecadmin@client1:~$ euca-disassociate-address 192.168.10.200
```

To release a public IP address from a specify user:

```
uecadmin@client1:~$ euca-release-address 192.168.10.200
```

# Chapter 8

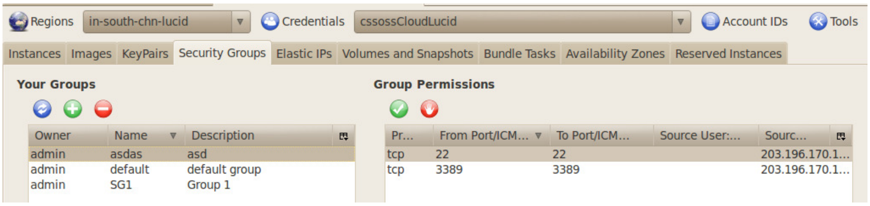
# Security

Eucalyptus provides ingress filtering for the instances based on the concept of security groups. A Security Group is a named set of rules that Eucalyptus applies to the incoming packets for the instances in Managed and Managed-NOVLAN modes. You can specify a security group while launching an instance. Each security group can have multiple rules associated with it. Each rule specifies the source IP/network, protocol type, destination ports etc. Any packet matching these parameters specified in a rule is allowed in. Rest of the packets are blocked.

A security group that does not have any rules associated with it causes blocking of all incoming traffic.

The mechanism only provides ingress filtering and does not provide any egress filtering. As a result all outbound traffic is allowed. If you need to implement egress filtering, you will need to implement that inside the instance using a firewall.

Tools like Hybridfox let you manage security groups and also let you specify a security group while launching an instance.



Your Groups

| Owner | Name | Description |
|---|---|---|
| admin | asdas | asd |
| admin | default | default group |
| admin | SG1 | Group 1 |

Group Permissions

| Pr... | From Port/ICM... | To Port/ICM... | Source User:... | Sourc... |
|---|---|---|---|---|
| tcp | 22 | 22 | | 203.196.170.1... |
| tcp | 3389 | 3389 | | 203.196.170.1... |

Figure 8.1: Security Groups in HybridFox

Here are a few euca commands to manage security groups:

Create a security group named “webservers”:

```
uecadmin@client1:~$ euca—add—group —d ”Web Servers” webservers
```

Add a rule to the security group “webservers” allowing icmp and tcp traffic from a.b.c.d:

```
uecadmin@client1:~$ euca—authorize —P tcp —s a.b.c.d webservers
uecadmin@client1:~$ euca—authorize —P icmp —s a.b.c.d webservers
```

The added rules can be viewed with euca-describe-groups command:

```
uecadmin@client1:~$ euca—describe—groups
GROUP admin default default group
GROUP admin webservers Web Servers
PERMISSION admin webservers ALLOWS icmp —1 —1 FROMCIDR 0.0.0.0/0
PERMISSION admin webservers ALLOWS tcp 22 22 FROMCIDR 0.0.0.0/0
```

Launch an instance associated with the security group “webservers”:

```
uecadmin@client1:~$ euca—run—instances emi—XXXXXXXX —k mykey —g
webservers
```

List the details of the existing security groups:

```
uecadmin@client1:~$ euca—describe—groups
```

Remove the rule for icmp traffic from the source ip a.b.c.d from the security group “webservers”:

```
uecadmin@client1:~$ euca—revoke —P icmp —s a.b.c.d webservers
```

Delete the security group “webservers”:

```
uecadmin@client1:~$ euca—delete—group webservers
```

# Chapter 9

# Troubleshooting

## 9.1 Eucalyptus Log Files

The following log files help in debugging issues encountered while working with Eucalyptus:

| **Eucalyptus Component** | **Log File** |
|---|---|
| CC | cc.log, httpd-cc_error_log, registration.log |
| NC | nc.log, httpd-nc_error_log, euca_test_nc.log |
| CLC | cloud-debug.log, cloud-error.log, cloud-output.log, axis2c.log |
| SC | sc-stats.log, registration.log |
| WS3 | walrus-stats.log, registration.log |

Table 9.1: Component Log Files

The logging level is controlled by the LOGLEVEL macro in eucalyptus.conf of the respective component. The log levels are DEBUG, INFO, WARN, ERROR, and FATAL (in descending order of verbosity). The default level is DEBUG (everything).

“tail -f ” on the log files is a good way to understand what is happening with the components of Eucalyptus.

## 9.2 Linux commands

Linux comamnds like ps, top etc. run on CC and NC are helpful to understand the state of the processes related to Eucalyptus.

## 9.3 Libvirt tools

NC launches the instances by making libvirt calls to launch a VM. You can use libvirt tools like virsh for troubleshooting any hypervisor related issues.

## 9.4 Launching an instance with VNC attached (Hack)

If the output of euca-describe-instances shows the status of an instance as “running”, but if you are not able to SSH into it or RDP into it, it is tough to find out where the failure is. In such cases, a VNC interface to the instance is generally of great help to troubleshoot further. But Eucalyptus does not provide a way of connecting to the instance through VNC. The following hack can help in setting up a VNC interface to the instance.

### 9.4.1 Windows instances

On the NC, edit /usr/share/eucalyptus/gen_kvm_libvirt_xml (if KVM is the hypervisor) to change

```
print <<EOF;
        <serial type=”file”>
            <source path='BASEPATH/console.log'/>
            <target port='1'/>
        </serial>
    </devices>
</domain>
EOF
```

to:

```
print <<EOF;
        <serial type=”file”>
            <source path='BASEPATH/console.log'/>
            <target port='1'/>
        </serial>
    <graphics type='vnc' listen='A.B.C.D' port='5904'/>
    </devices>
</domain>
EOF
```

where A.B.C.D is the IP address of the interface on NC that is reachable from the client machine where you want to run the VNC client. We assume that your NC has two interfaces and you are able to use the second interface to reach it without going through CC. Again, this is only for troubleshooting. In a production setup, you may have the NC connected to the outside world only through CC.

With these changes, KVM adds the "-vnc" option to the command with the specified NC's IP address on display number 4.

To connect to the instance through VNC client from a different machine, use the NC's IP address and display number 4.
Ex: A.B.C.D :4

In the case of Xen, you would modify the relevant sections of /usr/share/eucalyptus/gen_libvirt_xml. We have not tested with Xen, though.

### 9.4.2 Linux instances

You can use the same approach for Linux instances as well, with the caveat that you will not see any output with euca-get-console-output as console output does not get written to /var/lib/eucalyptus/instances/<user>/i-xxxxxxxx/console.log. The output will, of course, be displayed though VNC.

Note: This works only if we are debugging one instance at a time. If you attempt to start another instance while one instance is already running, KVM will try to attach the same VNC display 4 to the second instance and will fail with the following error in /var/log/libvirtd/qemu/i-XXXXXXXX.log:

```
inet_listen_opts: bind(ipv4,A.B.C.D,5904): Address already in use
inet_listen_opts:FAILED
```

# Appendix A

# Hacks

## A.1 Bringing up a Windows instance without using kernel and ramdisk images

Edit /usr/share/eucalyptus/gen_kvm_libvirt_xml to remove the lines that add “-kernel” and “-initrd” options in the libvirt.xml.

Here is the patch for /usr/share/eucalyptus/gen_kvm_libvirt_xml that ships with Ubuntu 10.10 (Maverick Meerkat):

```
DISCLAIMER: This patch works with the Windows instances on Maverick Meerkat.
This would work as long as all your disk images are bootable and you
don't need the kernel and ramdisk images. If you need the flexibility
of launching instances using different kernel/ramdisk combinations,
this hack is not for you!
```

```
91c91
< GetOptions('ramdisk' => sub { $use_ramdisk = 1; },
−−−
> GetOptions('ramdisk' => sub { $use_ramdisk = 0; },
117d116
< print ” <kernel>BASEPATH/kernel</kernel>\n”;
119c118,119
< print ” <initrd>BASEPATH/ramdisk</initrd>\n”;
−−−
> print ” <kernel>BASEPATH/kernel</kernel>\n”;
> print ” <initrd>BASEPATH/ramdisk</initrd>\n”;
125c125
< print ” <cmdline>root=/dev/sda1 console=ttyS0</cmdline>\n”
−−−
> # print ” <cmdline>root=/dev/sda1 console=ttyS0</cmdline>\n”
187a188
```

# Appendix B

# Eucalyptus Commands

## B.1 Image management

### B.1.1 euca-bundle-image

```
Bundles an image for use with Eucalyptus or Amazon EC2.
euca−bundle−image −i, −−image image_path −u, −−user user [−c, −−cert cert_path] ↩
      [−k, −−privatekey pri
vate_key_path] [−p, −−prefix prefix] [−−kernel kernel_id] [−−ramdisk ramdisk_id] [−B,↩
       −−block−device−mapping map
ping] [−d, −−destination destination_path] [−−ec2cert ec2cert_path] [−r, −−arch ↩
      target_architecture] [−−batch]
[−h, −−help] [−−version] [−−debug]
REQUIRED PARAMETERS
−i, −−image Path to the image file to bundle.
−u, −−user User ID (12−digit) of the user who is bundling the image.
OPTIONAL PARAMETERS
−c, −−cert Path to the user's PEM encoded certificate.
−k, −−privatekey Path to the user's PEM encoded private key.
−p, −−prefix The prefix for the bundle image files. (default: image name).
−−kernel The kernel to be associated with the bundled image.
−−ramdisk The ramdisk to be associated with the bundled image.
−B, −−block−device−mapping Default block device mapping for the image (comma−↩
      separated list of key=value
pairs).
−d, −−destination Directory to store the bundled image in (default: ”/tmp”). ↩
      Recommended.
−−ec2cert The path to the Cloud's X509 public key certificate.
−r, −−arch Target architecture for the image ('x86_64' or 'i386' default: 'x86_64').
−−batch Run in batch mode (compatibility only. has no effect).
−a, −−access−key User's Access Key ID.
−s, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or /etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.1.2 euca-bundle-vol

```
Bundle the local filesystem of a running instance as a bundled image.
euca—bundle—vol —u, ——user user —s, ——size size_in_MB
[—c, ——cert cert_path] [—k, ——privatekey private_key_path]
[—a, ——all] [—e, ——exclude dir1, dir2,...dirN] [—p, ——prefix prefix]
[——[no—]inherit] [—v, ——volume volume_path] [——fstab fstab_path]
[——generate—fstab] [——kernel kernel_id] [——ramdisk ramdisk_id] [—B,
——block—device—mapping mapping]
[—d, ——destination destination_path] [——ec2cert ec2cert_path] [—r,
——arch target_architecture] [——batch] [——version]
REQUIRED PARAMETERS
—u, ——user User ID (12—digit) of the user who is bundling the image.
—s, ——size Size for the image in MB (default: 10GB or 10240MB).
OPTIONAL PARAMETERS
—c, ——cert Path to the user's PEM encoded certificate.
—k, ——privatekey Path to the user's PEM encoded private key.
—a, ——all Bundle all directories (including mounted filesystems).
—p, ——prefix The prefix for the bundle image files. (default: image
name).
——[no—]inherit Add (or do not add) instance metadata to the bundled
image. Inherit is set by default.
—e, ——exclude comma—separated list of directories to exclude.
——kernel The kernel to be associated with the bundled image.
——ramdisk The ramdisk to be associated with the bundled image.
—B, ——block—device—mapping Default block device mapping for the image
(comma—separated list of key=value pairs).
—d, ——destination Directory to store the bundled image in (default:
”/tmp”). Recommended.
——ec2cert The path to the Cloud's X509 public key certificate.
—r, ——arch Target architecture for the image ('x86_64' or 'i386'
default: 'x86_64').
—v, ——volume Path to mounted volume to create the bundle from (default:
”/”).
——fstab Path to the fstab to be bundled into the image.
——generate—fstab Generate fstab to bundle into the image.
——batch Run in batch mode (compatibility only. has no effect).
```

### B.1.3 euca-delete-bundle

```
Delete a previously uploaded bundle.
euca—delete—bundle —b, ——bucket bucket [—m, ——manifest manifest_path]
[—p, ——prefix prefix] [——clear] [—h, ——help] [——version] [——debug]
REQUIRED PARAMETERS
bucket The name of the bucket to delete from.
OPTIONAL PARAMETERS
—m, ——manifest The path to the manifest file.
—b, ——bucket The bucket to delete the bundle from (if no
manifest is specified, all images from this
bucket will be deleted).
—p, ——prefix The filename prefix for bundled files. Defaults
to image name.
——clear Delete the bucket containing the image.
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
```

```
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.1.4 euca-download-bundle

```
Downloads a bundled image from a bucket.
euca—download—bundle —b, ——bucket bucket [—m, ——manifest manifest_path]
[—d, ——directory directory]
[—h, ——help] [——version] [——debug]
REQUIRED PARAMETERS
—b, ——bucket The name of the bucket to download from.
OPTIONAL PARAMETERS
—m, ——manifest The path to the manifest file.
—d, ——directory The name of the directory to download the bundled parts
to.
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.1.5 euca-unbundle

```
Unbundles a previously bundled image.
euca—unbundle—image —m, ——manifest manifest_path [—k, ——privatekey
private_key_path]
[—s, ——source source_directory] [—d, ——destination
destination_directory]
[—h, ——help] [——version] [——debug]
REQUIRED PARAMETERS
—m, ——manifest Path to the manifest file for the bundled image.
OPTIONAL PARAMETERS
—d, ——destination Directory to store the image to (defaults to the
current directory).
—s, ——source Source directory for the bundled image parts (defaults to
manifest directory).
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.1.6 euca-upload-bundle

```
Upload a previously bundled image to the cloud.
euca−upload−bundle −b, −−bucket bucket −m, −−manifest manifest_path [−a,
−−access−key access_key_id]
[−s, −−secret−key secret_key] [−−acl canned_acl] [−−ec2cert
ec2cert_path] [−d, −−directory bundle_path]
[−−part part] [−−url url] [−−skipmanifest] [−−debug] [−h, −−help]
[−−version]
REQUIRED PARAMETERS
−b, −−bucket The name of the bucket to upload to. Bucket will be created
if it does not exist.
−m, −−manifest The path to the manifest file for the bundled image.
OPTIONAL PARAMETERS
−−acl Canned access control policy (defaults to ”aws−exec−read”).
−−ec2cert The path to the Cloud's X509 public key certificate.
−d, −−directory The directory containing the bundled image to upload
(defaults to the manifest directory).
−−part Uploads specified part and all subsequent parts.
−−skipmanifest Do not upload the manifest file.
−a, −−access−key User's Access Key ID.
−s, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.1.7 euca-describe-images

```
Shows information about machine images.
euca−describe−images [−a] [−o owner] [−x user] [−h, −−help] [−−version]
[−−debug] [image1 image2 ... imageN]
OPTIONAL PARAMETERS
image1 image2 ... imageN Images to describe.
−a Show all images that the user has access to.
−o Show only images owned by the owner specified are displayed.
−x Show only images that the specified user is permitted to launch.
−A, −−access−key User's Access Key ID.
−S, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.1.8 euca-describe-image-attribute

```
Show image attributes.
euca−describe−image−attribute [−l, −−launch−permission] [−p,
−−product−code]
[−B, −−block−device−mapping] [−−kernel] [−−ramdisk]
[−h, −−help] [−−version] [−−debug] image_id
```

```
REQUIRED PARAMETERS
image_id unique identifier for the image that you want to retrieve the
attributes for.
OPTIONAL PARAMETERS
—B, ——block—device—mapping show block device mapping.
—l, ——launch—permission show launch permissions.
—p, ——product—code show the product codes associated with the image
——kernel show the kernel id associated with the image.
——ramdisk show the ramdisk id associated with the image.
—a, ——access—key user's Access Key ID.
—s, ——secret—key user's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or /etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

## B.1.9 euca-register

```
Registers a manifest for use with the cloud.
euca—register [—h, ——help] [——version] [——debug] image_location
REQUIRED PARAMETERS
image_location — path to the uploaded image (bucket/manifest).
OPTIONAL PARAMETERS
—a, ——access—key user's Access Key ID.
—s, ——secret—key user's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or /etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

## B.1.10 euca-deregister

```
Deregisters an image from the cloud.
euca—deregister [—h, ——help] [——version] [——debug] image_id
REQUIRED PARAMETER
image_id unique identifier for the image to be deregistered.
OPTIONAL PARAMETER
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

# B.2 Instance management

## B.2.1 euca-describe-instances

```
Shows information about instances.
euca-describe-instances [-h, --help] [--version] [--debug]
[instance1... instanceN]
OPTIONAL PARAMETERS
instance1... instanceN instances to describe.
-a, --access-key User's Access Key ID.
-s, --secret-key User's Secret Key.
-U, --url URL of the Cloud to connect to.
--config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
-h, --help Display this help message.
--version Display the version of this tool.
--debug Turn on debugging.
```

## B.2.2 euca-reboot-instances

```
Reboots specified instances.
euca-reboot-instances [-h, --help] [--version] [--debug]
[instance1... instanceN]
OPTIONAL PARAMETERS
instance1... instanceN instances to reboot.
-a, --access-key User's Access Key ID.
-s, --secret-key User's Secret Key.
-U, --url URL of the Cloud to connect to.
--config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
-h, --help Display this help message.
--version Display the version of this tool.
--debug Turn on debugging.
```

## B.2.3 euca-run-instances

```
Starts instances.
euca-run-instances [-n, --instance-count count] [-g, --group group_name]
[-k, --key keyname]
[--addressing addressing] [-t, --instance-type instance_type] [-z,
--availability-zone zone]
[--kernel kernel_id] [--ramdisk ramdisk_id] [-h, --help] [--version]
[--debug] image_id
REQUIRED PARAMETERS
image_id identifier for the image to run.
OPTIONAL PARAMETERS
-n, --instance-count Number of instances to run.
-g, --group Security group to run the instance under.
-k, --key Name of a (previously created) keypair to associate with this
reservation.
-d, --user-data User data for instances read from the command line.
```

```
−f, −−user−data−file User data for instances as a filename.
−−addressing Addressing mode (e.g., private).
−t, −−instance−type VM Image type to run the instance(s) as (default:
m1.small).
−z, −−availability−zone Availability zone to run the instance(s) in.
−−kernel Id of the kernel to be used to launch instance(s).
−−ramdisk Id of the ramdisk to be used to launch instance(s).
−a, −−access−key User's Access Key ID.
−s, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.2.4 euca-terminate-instances

```
Stops specified instances.
euca−terminate−instances [−h, −−help] [−−version] [−−debug]
[instance1… instanceN]
REQUIRED PARAMETERS
instance1… instanceN instances to terminate.
OPTIONAL PARAMETERS
−a, −−access−key User's Access Key ID.
−s, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.2.5 euca-confirm-product-instance

```
Confirm if instance is running with product code attached.
Euca−confirm−product−instance i, −−instance instance_id [−h, −−help]
[−version] [−−debug] product_code
REQUIRED PARAMETERS
−i, −−instance Unique identifier for a running instance.
product_code The product code to verify.
OPTIONAL PARAMETERS
−a, −−access−key User's Access Key ID.
−s, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

# B.3 Storage management

## B.3.1 euca-attach-volume

```
Attaches a volume to an instance.
euca—attach—volume —i, ——instance instance_id —d, ——device device
[—h, ——help] [——version] [——debug] volume_id
REQUIRED PARAMETERS
—i, ——instance unique id of a running instance to attach the volume to.
—d, ——device local device name (inside the guest VM) to use.
volume_id unique id for a volume to attach.
OPTIONAL PARAMETERS
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

## B.3.2 euca-create-volume

```
Creates a volume in a specified availability zone.
euca—create—volume —S, ——size size | ——snapshot snapshot_id —z zone
[—h, ——help] [——version] [——debug]
REQUIRED PARAMETERS
—s, ——size size of the volume (in GiB).
——snapshot snapshot id to create the volume from. Either size or
snapshot can be specified (not both).
—z, ——zone availability zone to create the volume in
OPTIONAL PARAMETERS
—A, ——access—key User's Access Key ID.
—S, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

## B.3.3 euca-delete-volume

```
Deletes a volume.
euca—delete—volume [—h, ——help] [——version] [——debug] volume_id
REQUIRED PARAMETERS
volume_id unique identifier for the volume to be deleted.
OPTIONAL PARAMETERS
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
```

```
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.3.4 euca-describe-volumes

```
Shows information about volumes.
euca−describe−volumes [−h, −−help] [−−version] [−−debug]
[volume1 volume2...volumeN]
OPTIONAL PARAMETERS
volume1 volume2...volumeN volumes to describe.
−a, −−access−key User's Access Key ID.
−s, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.3.5 euca-detach-volume

```
Detaches a volume from an instance.
euca−detach−volume [−h, −−help] [−−version] [−−debug] volume_id
REQUIRED PARAMETERS
volume_id unique id for a volume to detach.
OPTIONAL PARAMETERS
−a, −−access−key User's Access Key ID.
−s, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.3.6 euca-create-snapshot

```
Creates a snapshot from an existing volume.
euca−create−snapshot [−h, −−help] [−−version] [−−debug] volume_id
REQUIRED PARAMETERS
volume_id unique id for a volume to snapshot.
OPTIONAL PARAMETERS
−a, −−access−key User's Access Key ID.
−s, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
```

```
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.3.7 euca-delete-snapshot

```
Deletes a snapshot.
euca—delete—snapshot [—h, ——help] [——version] [——debug] snapshot_id
REQUIRED PARAMETERS
snapshot_id unique identifier for the snapshot to be deleted.
OPTIONAL PARAMETERS
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.3.8 euca-describe-snapshots

```
Shows information about snapshots.
euca—describe—snapshots [—h, ——help] [——version] [——debug]
[snapshot1 snapshot2...snapshotN]
OPTIONAL PARAMETERS
snapshot1 snapshot2...snapshotN — snapshots to describe.
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

## B.4 Network management

### B.4.1 euca-allocate-address

```
Allocate a public IP address.
euca—allocate—address [—h, ——help] [——version] [——debug]
OPTIONAL PARAMETERS
—a, ——access—key User's Access Key ID.
```

```
−s, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.4.2 euca-associate-address

```
Associate an instance with a public IP address.
euca−associate−address −i, −−instance instance_id
[−h, −−help] [−−version] [−−debug] ip
REQUIRED PARAMETERS
−i, −−instance unique identifier for a running instance to associate the
address with.
ip IP address to associate.
OPTIONAL PARAMETERS
−a, −−access−key user's Access Key ID.
−s, −−secret−key user's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or /etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.4.3 euca-describe-addresses

```
Shows information about addresses.
euca−describe−addresses [−h, −−help] [−−version] [−−debug] [ip1
ip2...ipN]
OPTIONAL PARAMETERS
ip1 ip2...ipN IP addresses to display information about.
−a, −−access−key User's Access Key ID.
−s, −−secret−key User's Secret Key.
−U, −−url URL of the Cloud to connect to.
−−config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
−h, −−help Display this help message.
−−version Display the version of this tool.
−−debug Turn on debugging.
```

### B.4.4 euca-disassociate-address

```
Disassociate a public IP address from an instance.
euca−disassociate−address [−h, −−help] [−−version] [−−debug] ip
REQUIRED PARAMETERS
ip IP address to disassociate.
```

```
OPTIONAL PARAMETERS
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.4.5 euca-release-address

```
Releases a public IP address.
euca—release—address [—h, ——help] [——version] [——debug] ip
REQUIRED PARAMETERS
ip The public IP address to release.
OPTIONAL PARAMETERS
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

## B.5 Security

### B.5.1 euca-add-group

```
Creates a new security group.
euca—add—group —d, ——description group_description
[—h, ——help] [——version] [——debug] group_name
REQUIRED PARAMETERS
—d, ——description Description for the group to be created.
OPTIONAL PARAMETERS
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.5.2 euca-delete-group

```
Deletes a security group.
euca—delete—group [—h, ——help] [——version] [——debug] group_name
REQUIRED PARAMETER
group_name unique name for a group to delete.
OPTIONAL PARAMETER
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.5.3 euca-describe-groups

```
Shows information about groups.
euca—describe—groups [—h, ——help] [——version] [——debug] [group1
group2...groupN]
OPTIONAL PARAMETERS
group1 group2...groupN groups to describe.
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.5.4 euca-add-keypair

```
Creates a new key pair for use with instances.
euca—add—keypair [—h, ——help] [——version] [——debug] keypair_name
REQUIRED PARAMETERS
keypair_name — unique name for a keypair to create.
OPTIONAL PARAMETERS
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.5.5 euca-delete-keypair

```
Delete a key pair.
euca—delete—keypair [—h, ——help] [——version] [——debug] keypair_name
REQUIRED PARAMETERS
keypair_name unique name for the keypair to delete.
OPTIONAL PARAMETERS
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.5.6 euca-describe-keypairs

```
Shows information about keypairs.
euca—describe—keypairs [—h, ——help] [——version] [——debug]
[keypair1 keypair2...keypairN]
OPTIONAL PARAMETERS
keypair1 keypair2...keypairN keypairs to describe.
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.5.7 euca-describe-availability-zones

```
Shows information about availability zones.
euca—describe—availability—zones [——region region] [—h, ——help]
[——version] [——debug] [zone1 zone2...zoneN] [verbose]
OPTIONAL PARAMETERS
——region region to describe availability zones for
zone1 zone2...zoneN zones to describe.
verbose show verbose resource information (Eucalyptus administrator—only
functionality).
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.5.8 euca-describe-regions

```
Shows information about regions.
euca—describe—regions [—h, ——help] [——version] [——debug]
[region1 region2...regionN]
OPTIONAL PARAMETERS
region1 region2...regionN regions to describe.
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.5.9 euca-get-console-output

```
Prints console output from a running instance.
euca—get—console—output [—h, ——help] [——version] [——debug] instance_id
REQUIRED PARAMETERS
instance_id unique identifier for the instance show the console output
for.
OPTIONAL PARAMETERS
—a, ——access—key User's Access Key ID.
—s, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

### B.5.10 euca-modify-image-attribute

```
Modify attributes for an image.
euca—modify—image—attribute [—l, ——launch—permission] [—p,
——product—code]
[—a, ——add entity ] [—r, ——remove entity]
[—h, ——help] [——version] [——debug] image_id
REQUIRED PARAMETERS
image_id Unique identifier for the image that you want to modify the
attributes for.
OPTIONAL PARAMETERS
—l, ——launch—permission Show launch permissions.
—p, ——product—code Show the product codes associated with the image
—a, —add Entity (typically, user id) to add.
—r, ——remove Entity (typically, user id) to remove.
—A, ——access—key User's Access Key ID.
—S, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
```

```
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
-h, --help Display this help message.
--version Display the version of this tool.
--debug Turn on debugging.
```

### B.5.11 euca-reset-image-attribute

```
Reset attributes for an image.
euca-reset-image-attribute -l, --launch-permission
[-h, --help] [--version] [--debug] image_id
REQUIRED PARAMETERS
-l, --launch-permission show launch permissions.
image_id - unique identifier for the image that you want to reset the
attributes for.
OPTIONAL PARAMETERS
-a, --access-key User's Access Key ID.
-s, --secret-key User's Secret Key.
-U, --url URL of the Cloud to connect to.
--config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
-h, --help Display this help message.
--version Display the version of this tool.
--debug Turn on debugging.
```

### B.5.12 euca-revoke

```
Revoke a rule for a security group.
euca-revoke [-P | --protocol protocol] [-p | --port-range port_range]
[-t | --icmp-type-code type:code] [-o | --source-group source_group]
[-u | --source-group-user source_group_user] [-s | --source-subnet
source_subnet]
[-h, --help] [--version] [--debug] group_name
REQUIRED PARAMETERS
group_name Name of the group to add the rule to.
OPTIONAL PARAMETERS
-P, --protocol Protocol (”tcp” ”udp” or ”icmp”).
-p, --port-range Range of ports for the rule (specified as ”from-to”).
-t, --icmp-type-code ICMP type and code specified as ”type:code”
-o, --source-group Group from which traffic is authorized by the rule.
-u, --source-group-user User ID for the source group.
-s, --source-subnet The source subnet for the rule.
-A, --access-key User's Access Key ID.
-S, --secret-key User's Secret Key.
-U, --url URL of the Cloud to connect to.
--config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
-h, --help Display this help message.
--version Display the version of this tool.
--debug Turn on debugging.
```

### B.5.13 euca-version

```
Shows the Euca2ools version
euca—version [—h, ——help] [——debug]
```

### B.5.14 euca-authorize

```
Add a new rule to a security group.
euca—authorize [—P, ——protocol protocol] [—p, ——port—range port_range]
[—t, ——icmp—type—code type:code] [—o, ——source—group source_group]
[—u, ——source—group—user source_group_user] [—s, ——source—subnet
source_subnet]
[—h, ——help] [——version] [——debug] group_name
REQUIRED PARAMETERS
group_name Name of the group to add the rule to.
OPTIONAL PARAMETERS
—P, ——protocol Protocol (”tcp” ”udp” or ”icmp”).
—p, ——port—range Range of ports for the rule (specified as ”from—to”).
—t, ——icmp—type—code ICMP type and code specified as ”type:code”
—o, ——source—group Group from which traffic is authorized by the rule.
—u, ——source—group—user User ID for the source group.
—s, ——source—subnet The source subnet for the rule.
—A, ——access—key User's Access Key ID.
—S, ——secret—key User's Secret Key.
—U, ——url URL of the Cloud to connect to.
——config Read credentials and cloud settings from the
specified config file (defaults to $HOME/.eucarc or
/etc/euca2ools/eucarc).
—h, ——help Display this help message.
——version Display the version of this tool.
——debug Turn on debugging.
```

## B.6 Commands in UEC and not available in Eucalyptus installations

### B.6.1 uec-publish-tarball

```
Usage: uec—publish—tarball [ options ] tarfile bucket [arch]
register a UEC tarball (ie, per http://uec—images.ubuntu.com)
if arch is not provided, a name—based attempt is made to guess
options:
   —k | ——kernel k use previously registered kernel with id 'k'
                           specify 'none' for no kernel
   —K | ——kernel—file f bundle, upload, use file 'f' as kernel
   —r | ——ramdisk r use previously registered ramdisk with id 'r'
                           specify 'none' for no ramdisk
   —R | ——ramdisk—file f bundle, upload, use file 'f' as ramdisk
——resize s resize the image to 's' bundling uploading
                        's' must be valid input to uec—resize—image
—q | ——quiet be quiet, only output published ids
—l | ——use—loader use the loader kernel rather than linux kernel
Example: uec—publish—tarball lucid—uec—i386.tar.gz my—lucid—bucket i386
```

### B.6.2 uec-registration

```
uec-registration [--manual|--auto|--help]
Options:
   --manual Disable automatic component registration
   --auto Enable automatic component registration
   -h, --help This message
```

### B.6.3 uec-run-instances

```
uec-run-instances - wrapper for euca-run-instances that supports an
option for injecting public ssh keys retrievable from Launchpad.net
uec-run-instances [-l|--launchpad-id lp_id_1,lp_id_2,lp_id_3] [euca-run-instances ←↩
     options]
This program is a wrapper script for euca-run-instances(1) that takes one additional ←↩
     option,
-l|--launchpad-id.With this option, a user can specify a comma-separated list of
Launchpad.net usernames.Once the instance is booted, the cloud-init boot script will
retrieve the public ssh keys of the specified users from Launchpad.net using
ssh-import-lp-id(1).All other options besides -l|--launchpad-id are simply passed
on to euca-run-instances(1).
```

### B.6.4 uec-publish-image

```
uec-publish-image - publish a cloud image
uec-publish-image [OPTIONS] ARCH IMAGE BUCKET
-l|--add-launch <user_id>user_id can be ”all”, or ”none”
--dry-run only report what would be done
--allow-existing if a image is already registered simply report as if work was done
-o|--output <file> write registered id and manifest to file
--rename <publish_path> publish to bucket/<publish_path>, default: bucket/<←↩
     basename(image)>
-t|--type <type> type is one of kernel/ramdisk/image. if type is 'image', then:
    -k | --kernel k : use previously registered kernel with id 'k'
                      specify 'none' for no kernel
    -K | --kernel-file f : bundle, upload, use file 'f' as kernel
    -r | --ramdisk r : use previously registered ramdisk with id 'r'
                      specify 'none' for no ramdisk
    -R | --ramdisk-file f : bundle, upload, use file 'f' as ramdisk
    -B | --block-device-mapping m : specify block device mapping in bundle
-v|--verbose increase verbosity
ARCH Target architecture, one of i386 or x86_64
IMAGE Target image to upload and register
BUCKET Target bucket to publish the image to
```

### B.6.5 uec-query-builds

```
uec-query-builds - request information about available cloud images builds
uec-query-builds mode [options]
```

```
latest list information about the latest build available
       is-update-available
              indicate if there is a newer build available
       latest-ec2
              list information for the latest builds on ec2
```

### B.6.6 uec-resize-image

```
uec-resize-image - resize a cloud image
uec-resize-image [ options ] IMAGE SIZE [OUTPUT]
-v | --verbose
              show output of resize and fsck commands

       IMAGE Target file for resizing

       SIZE New size is specified per resize2fs(8), e.g. "1G" for 1 gigabyte

       [OUTPUT]
              If OUTPUT filname is given, do not modify 'IMAGE', but create new file '↩
                   OUTPUT'
```