Eucalyptus®

Cloud Computing Platform

User's Guide

Enterprise Edition 2.0

Eucalyptus Systems, Inc.

Eucalyptus Cloud Computing Platform
User's Guide
Enterprise Edition 2.0

Table of Contents

Introduction

Eucalyptus Enterprise Edition (EE) 2.0 is a Linux-based software architecture that implements scalable, efficiency-enhancing private and hybrid clouds within an enterprise's existing IT infrastructure. Eucalyptus provides Infrastructure as a Service (IaaS). This means that users can provision their own collections of resources (hardware, storage, and network) via Eucalyptus' self-service interface on an as-needed basis. A Eucalyptus cloud is deployed across an enterprise's "on-premise" data center and is accessed by users over enterprise intranet. Thus sensitive data remains secure from external intrusion behind the enterprise firewall.

Eucalyptus was designed from the ground up to be easy to install and as non-intrusive as possible. The software framework is highly modular, with industry-standard, language-agnostic communication. Eucalyptus is also unique by providing a virtual network overlay that both isolates network traffic of different users and allows two or more clusters to appear to belong to the same Local Area Network (LAN). Eucalyptus also interoperates seamlessly with Amazon's EC2 and S3 public cloud services and thus offers the enterprise a hybrid cloud capability.

Initially developed to support the high performance computing (HPC) research of Professor Rich Wolski's research group at the University of California, Santa Barbara, Eucalyptus is engineered based upon design principles that ensure compatibility with existing Linux-based data center installations. Eucalyptus can be deployed without modification on all major Linux OS distributions, including Ubuntu, RHEL/CentOS, openSUSE, and Debian. Ubuntu distributions now include the Eucalyptus software core as the key component of the *Ubuntu Enterprise Cloud.*

Who Should Read this Guide?

This guide is for Eucalyptus users who wish to run and manage Linux- and Windows-based virtual machines (VMs) within a Eucalyptus cloud.

What's in this Guide?

This guide contains instructions for users of the Eucalyptus EE 2.0 cloud platform. While these instructions apply generally to all client tools capable of interacting with Eucalyptus, the primary focus is on the use of Euca2ools (Eucalyptus command line tools). The following is an overview of the contents of this guide.

Section 1: Getting Started

This section shows you how to get started using your Eucalyptus cloud. Included is an overview of the features available to end users, followed by step-by-step instructions for creating an account, obtaining credentials, and setting up the environment variables required for client tools to interact with the Eucalyptus cloud.

Section 2: Euca2ools Quickstart

This section shows you how use Euca2ools (Eucalyptus' command-line tools) to quickly spin up a VM in the Eucalyptus cloud. Many frequently used commands and concepts are introduced in this section, including creating keypairs, running VM instances, and logging into VM instances.

Section 3: Managing VM Images

This section shows you how to manage the VM images from which you generate VM instances in the cloud. Subjects covered include preparing images for use in the cloud (bundle/upload/register), associating kernels and ramdisk with instances, and unbundling images. Both Linux- and Windows-based VM images are discussed.

Section 4: Working with VM Instances

This section shows you how to run and control VM instances. Subjects covered include querying the system, running VM instances, shutting down instances, rebooting instances, and logging into Windows VMs.

Section 5: VM Networking and Security

This section introduces the VM networking and security features of Eucalyptus, including *elastic IPs* and *security groups*. Subjects covered include allocating and associating IP addresses, creating security groups, and authorizing security group rules.

Section 6: Dynamic Block Volume and Snapshots

This section gets you started using Eucalyptus' dynamic block storage volumes with your VM instances. Subjects covered include creating volumes, creating snapshots, creating volumes from snapshots, and attaching volumes to VM instances.

Appendices

Appendix A: Installing and Using ElasticFox shows you how to manage your Eucalyptus user account using the ElasticFox plug-in for the FireFox web browser.

Appendix B: An Overview of Cloud Computing presents a broad overview of the general cloud-computing landscape and is useful for those who wish to gain a better understanding of how cloud computing can benefit the IT organization.

For detailed instructions on installing, configuring, managing and troubleshooting Eucalyptus EE 2.0, see the *Eucalyptus EE Administrator's Guide.*

What's New for Users in Eucalyptus EE 2.0?

Eucalyptus EE version 2.0 offers support for Windows VMs. This means that users can now run instances of Windows VMs in the Eucalyptus cloud. Eucalyptus EE currently supports Windows Server 2003/2008 and Windows 7 VMs.

Conventions Used in this Guide

Command-line instructions presented in this guide apply to any Unix-flavored system (e.g., Linux, OSX) with a bash shell, not just the cluster where Eucalyptus is installed. Command line examples in this guide use the following format to designate the user and the specific machine on which the user performs a given function:

Client machine = **`[bob@client]$`**

Left/right arrows **< >** are used to indicate variables. For example, within a code sample, a specific IP Address might be represented as **<IPAddress>** (where the actual input might be 192.168.7.82). Arguments accepting more than one variable inputs use the variable "N" to indicate multiple possible variable inputs. For example a command argument that accepts multiple node names may appear as <**nodeName1…nodeNameN**>**.**

Command line input samples use a forward slash (\) to denote that a single line of input is continued on the next line due to space limitations. Output samples use the carriage return (↵) to denote that a single line of output is continued on the next line due to space limitations.

Fonts: General instructional text is presented in Cambria 12 pt font. Command line input/output, as well as directory locations are printed in Courier 10 pt. font. For example: **`/etc/eucalyptus`**.

Bolded text is used within text discussions and command line samples to draw attention to the specific command or portion of command line input/output under discussion.

Italics are used throughout the text to introduce and emphasize important Eucalyptus and cloud computing terms and phrases.

Contacting Eucalyptus

Please send any questions, corrections, comments, or suggestions for this User's Guide to documentation@eucalyptus.com.

Section 1: Getting Started

This section helps you get started using your Eucalyptus cloud. First, we present an overview of the features available to end-users. Then we show you how to sign up for an account, obtain credentials, and set the *environment variables* that enable you to interact with Eucalyptus via client tools, such as Euca2ools—Eucalyptus command-line tools. Note that Eucalyptus is compatible with Amazon EC2, thus EC2 users can continue using ec-2 tools with Eucalyptus.

1.1 Overview of User Features

Eucalyptus users interacting via client tools with the Eucalyptus cloud have a variety of features at their disposal for implementing, managing, and maintaining their own collections of virtual resources (machines, network, and storage). The following is an overview of these features.

SSH Key Management **-** Eucalyptus employs public and private keypairs to validate user's identity when logging into VMs via SSH. Eucalyptus users can add, describe and delete keypairs.

Image Management **-** Before running instances, VM images must be prepared for use in the cloud. Eucalyptus users can bundle, upload, register, describe, download, unbundle, and deregister VM images.

Linux-based VM Management **-** Eucalyptus lets users run their own VM instances in the cloud. Users can run, describe, terminate, and reboot a wide variety of Linux-based VM instances that were prepared using Eucalyptus' Image Management functions.

Windows-based VM Management **-** Eucalyptus EE version 2.0 users can run, describe, terminate, reboot, and bundle instances of Windows VMs (Windows Server 2003/2008 and Windows 7 supported).

IP Address Management **-** Depending on the networking mode, users may have access to elastic IPs — public IP addresses that users can reserve and dynamically associate with VM instances. Eucalyptus users can allocate, associate, disassociate, describe, and release IP addresses.

Security Group Management - *Security groups* are sets of firewall rules applied to VM instances associated with the group. Eucalyptus lets users create, describe, delete, authorize, and revoke security groups.

Volume and Snapshot Management **-** Eucalyptus lets users create *dynamic block volumes*, which are analogous to raw block storage devices that can be used with VM instances. Users can create, attach, detach, describe, bundle, and delete volumes. Users can also create and delete *snapshots* of volumes and create new volumes from snapshots.

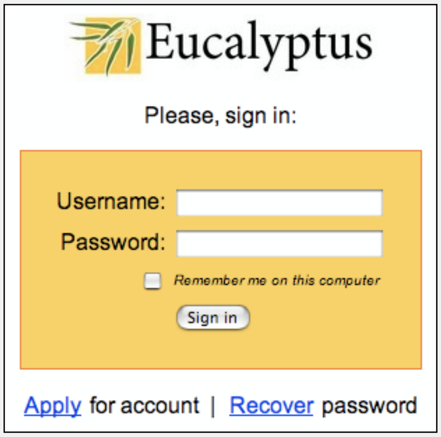
1.2 Signing up for an Account

Before you can interact with your Eucalyptus cloud, you must first sign up for an account via the Eucalyptus Web interface.

To sign up for an account:

1. Open your browser to the Web page of your Eucalyptus cloud installation. Ask your system administrator for the precise URL if you don't know it. The URL is of this form: **https://<YourCloudServer>:8443/**.

2. Click "Apply" to access the application form (**Figure 1.1**).



Eucalyptus

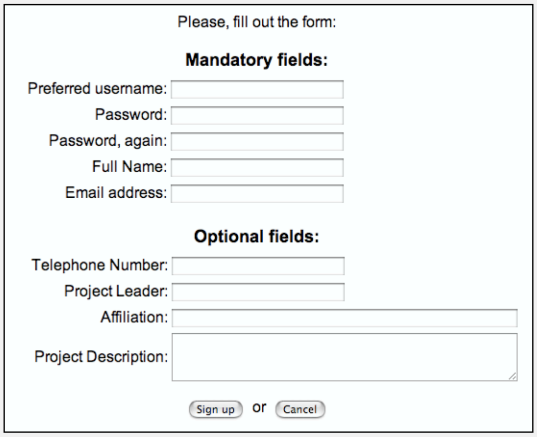
Please, sign in:

Username:
Password:
□ *Remember me on this computer*
Sign in

Apply for account | Recover password

Figure 1.1. Eucalyptus sign in window. Click "Apply" to open the account application form.

3. Fill out the Eucalyptus account application form (**Figure 1.2**). An approval email is sent to your mailbox. Note that you won't be able to use Eucalyptus until your administrator has approved and enabled your account. The more complete the information you provide on your account application the easier for the administrator to verify your identity.



Please, fill out the form:

Mandatory fields:

Preferred username:
Password:
Password, again:
Full Name:
Email address:

Optional fields:

Telephone Number:
Project Leader:
Affiliation:
Project Description:

Sign up or Cancel

Figure 1.2. Eucalyptus Account Application form. Fill out this form and click "Sign up" to create a new Eucalyptus account. An approval email containing a confirmation URL is sent to your inbox by the cloud administrator.

4. Load the confirmation URL that you received in the approval email message from the cloud administrator.

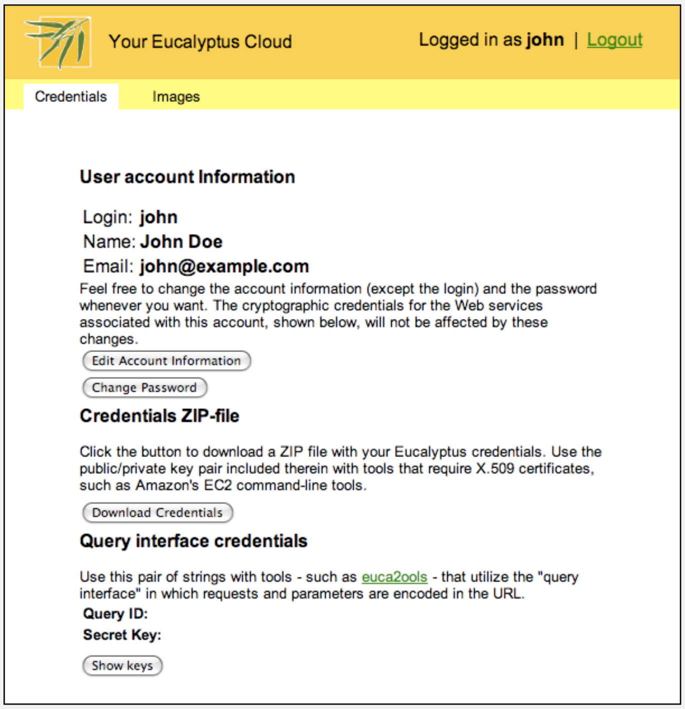
5. Log in to the system with the username and password that you chose when filling out the application form.

1.3 Obtaining User Credentials

Users must have proper credentials to use client tools (such as Euca2ools) to interact with the Eucalyptus cloud. After you have signed up for an account and have received approval from the administrator, you can obtain your credentials via the Eucalyptus Web UI. First, log into the Eucalyptus Web interface using your username and password. Next, on the *Your Eucalyptus Cloud* page, click on the *Credentials* tab, and proceed with the following steps:

To obtain user credentials:

1. Click the Download Credentials button to download your credentials zip-file **(Figure 1.3**) and save it to a secure location. The zip-file contains your public/private key pair, a bash script, and several other required files.



User account Information

Login: john
Name: John Doe
Email: john@example.com

Feel free to change the account information (except the login) and the password whenever you want. The cryptographic credentials for the Web services associated with this account, shown below, will not be affected by these changes.

Credentials ZIP-file

Click the button to download a ZIP file with your Eucalyptus credentials. Use the public/private key pair included therein with tools that require X.509 certificates, such as Amazon's EC2 command-line tools.

Query interface credentials

Use this pair of strings with tools - such as euca2ools - that utilize the "query interface" in which requests and parameters are encoded in the URL.

Query ID:
Secret Key:

Figure 1.3. Credentials window of the Eucalyptus Web UI. Click Download Credentials to obtain your credentials.

2. Unzip your credentials zip-file to a directory of your choice. In the following example we download the credentials zip file to **~/.euca**, then change access permissions, as shown:

```
[bob@client]$ mkdir ~/.euca
[bob@client]$ cd ~/.euca
[bob@client]$ unzip <securelocation>/euca2-<user>-x509.zip
[bob@client]$ chmod 0700 ~/.euca
[bob@client]$ chmod 0600 *
```

1.4 Interacting with Eucalyptus via Client Tools

Eucalyptus users have a variety of *client tools* at their disposal for interacting with the Eucalyptus cloud. Some of these client tools are provided as a hosted service (e.g., a SaaS offering) and others are deployed on premise. Each of these tools utilizes the user's credentials to make requests on behalf of the user. Eucalyptus provides a set of command-line tools, called Euca2ools, while there are many graphical Web UI tools available in the Eucalyptus/Amazon AWS eco-system to allow users to interact with their on-premise private Eucalyptus cloud. The remainder of this guide uses Euca2ools as an example of a command-line tool to demonstrate the different functionalities of Eucalyptus. These tools conform to the command-line tools that Amazon distributes as part of their EC2 offering (EC2 tools). If you wish to use a free Web-based graphical tool to interact with your Eucalyptus cloud, we describe how to use one such tool in *Appendix A: Installing and Using ElasticFox.*

1.4.1 What is Euca2ools?

Euca2ools is a set of command-line tools for interacting with Web services that export a REST/Query-based API[1] compatible with Amazon's EC2 and S3 services. These tools can be used with both the Eucalyptus cloud-computing platform and Amazon Elastic Computing Cloud (EC2). Euca2ools emulate the command-line tools distributed by Amazon (api-tools and ami-tools) and generally accept the same command-line options and honor the same environmental variables. Euca2ools, however, were designed from scratch in Python, relying on the Boto library and M2Crypto toolkit.

1.4.2 Installing Euca2ools

[1] **REST/Query-based API –** Web service interface using HTTP-based representational messaging architecture to conduct client-server (request and response) interactions.

Euca2ools are included with installation packages of Eucalyptus EE 2.0. Please check with your administrator to confirm that Euca2ools is installed properly on your client machine.

1.4.3 Setting Environment Variables

Euca2ools uses encrypted credentials to authenticate user identity. Two types of credentials are issued by EC2- and S3-compatible services (such as Eucalyptus): x.509 *PEM-encoded certificates* and *keys*. While some commands only require key authentication, it is best to always specify both types of credentials. In addition, unless the Web services reside on 'localhost', the URLs of the EC2- and S3-compatible Web service endpoints must also be specified. You must either define a set of *environment variables* in advance or use command-line options to allow Euca2ools to communicate with the cloud and verify user identity (**Figure 1.4**).

Variable	Option	Explanation
EC2_URL	-U or—url <url>	http://host:8773/services/Eucalyptus
S3_URL	-U or –url <url>	http://host:8773/service/Walrus
EUCALYPTUS_CERT	--ec2cert_path <file>	Path to cloud cert
EC2_CERT	-c or –cert <file>	User's PEM-encoded certificate
EC2_PRIVATE_KEY	-k or –privatekey <file>	User's PEM-encoded private key
EC2_ACCESS_KEY	-a or –access-key <key>	Access Key ID / Query ID
EC2_SECRET_KEY	-s or –secret-key <key>	Secret Access Key / Secret Key

Figure 1.4 Eucalyptus environment variables. These environment variables must be set to the correct values to allow the client tools (such as Euca2ools and ec2 tools) to communicate with the cloud.

You can specify each value individually via the command line or set all of them collectively by sourcing the **`eucarc`** file. The `eucarc` is a resource configuration file that defines the correct values for each associated environment variable. Sourcing `eucarc` sets these environment variables to the correct values.

To source the `eucarc` file, enter the following:

```
[bob@client]$ source ~/.euca/eucarc
```

The appropriate environment variables are now set and you are ready to use Euca2ools to interact with your Eucalyptus cloud.

TIP 1: What is the “environment variable must be set” warning?

When entering a Euca2ools command, you may receive an “`environment variable must be set`” warning, as follows:

```
[bob@client]$ euca-describe-instances
EC2_ACCESS_KEY environment variable must be set.
```

This message indicates that you have not defined the value of the EC2_ACCESS_KEY. You can address this problem by either sourcing the eucarc file or use the –access_key command-line option.

1.5 Getting Help

You can use either *help pages* or *man pages* to view information about Euca2ools commands and associated options.

- To see a help page, enter `<commandName>` **`--help`**
- To see a man page, enter **`man`** `<commandName>`

For example, to get help for the `euca-bundle-images` command, enter either of the following commands:

```
[bob@client]$ euca-bundle-image --help #(shows a help page)
[bob@client]$ man euca-bundle-image    #(shows a man page)
```

Section 2: Euca2ools Quickstart

Now that you have created a user account, obtained credentials, and set your environment variables by sourcing the **`eucarc`** configuration file, you are ready to begin uploading and running VM instances in your Eucalyptus cloud. This section provides a quickstart guide to help familiarize you with the most frequently used Euca2ools commands and to get you quickly up and running. We begin with introducing Eucalyptus query commands that enable you to view important information about the status of resources, images, and instances. Then we show you how to create necessary "keypairs" for user authentication. Finally, we show you how to run (instantiate), log into, and terminate VM instances.

2.1 Querying the System

Euca2ools *query commands* let you view information about various elements of the Eucalyptus cloud. Using query commands is an essential part of working with VM instances, thus we recommend you familiarize yourself with these commands before proceeding. Query commands begin with the prefix **`euca-describe-.`** The following is a list of definitions of some frequently used query commands for your reference. These commands are demonstrated throughout this guide.

- **`euca-describe-images`** `[-a] [-o owner] [-x user][image1 image2... imageN]`

  ```
  -a             Show all images that the user has access to.

  -o             Show only images owned by the specified owner.

  -x             Show only images that the specified user is permitted
                 to launch.

  image1 image2...
  imageN         images to describe.
  ```

- **`euca-describe-instances`** `[instance1...instanceN]`

  ```
  instance1...
  instanceN      instances to describe.
  ```

- **`euca-describe-keypairs`** `[keypair1...keypairN]`

  ```
  keypair1...
  keypairN       keypairs to describe.
  ```

- **`euca-describe-availability-zones`** `[--region region][zone1...zoneN]`

  ```
  --region       region to describe availability zones for

  zone1...zoneN     availability zones to be described.
  ```

2.2 Creating Keypairs

Eucalyptus uses *cryptographic keypairs* to verify access to VM instances. Before you can run a VM instance, you must create a keypair using the **euca-add-keypair** command. Creating a keypair generates two keys: a *public key* (saved within Eucalyptus) and a corresponding *private key* (output to the user as a character string). To enable this private key you must save it to a file and set appropriate access permissions (using the **chmod** command), as shown in the example below.

When you create a VM instance, the public key is then injected into the VM. Later, when attempting to login to the VM instance via SSH, the public key is checked against your private key to verify access. Note that the private key becomes obsolete when the public key is deleted.

- **euca-add-keypair** keypair_name

  keypair_name   unique name for a keypair to create

To create a keypair:

1. Enter **euca-add-keypair** and a keypair name. In this example, we use the name "mykey," but you may use any string you wish. Note that the private key is saved to the local directory in a file called "mykey.private":

```
[bob@client]$ euca-add-keypair mykey >mykey.private
```

2. Change access permissions to enable the private key in the local directory:

```
[bob@client]$ chmod 0600 mykey.private
```

3. Query the system with **euca-describe-keypairs** to view the public key "mykey":

```
[bob@client]$ euca-describe-keypairs
KEYPAIR mykey 7e:ee:fc:c7:4d:bf:12:96:73:18:c6:22:3a:c8:2a:08
```

2.3 Running a VM Instance

Running a VM instance in Eucalyptus requires first identifying the specific *VM image* you wish to run, then *instantiating* that image using the **euca-run-instances** command. Eucalyptus EE lets you run instances from both Linux-based images (e.g., Ubuntu, RHEL/CentOS, openSUSE, Debian, Fedora, etc.) and Windows-based images (i.e., Windows Server 2003/2008 and Windows 7).

- **`euca-run-instances`** `[-k] [-n] image id`

  ```
  image id       Identifier of the image to run.
  -k             Name of keypair to associate with instance.
  -n             Number of instances to run.
  ```

To run a VM instance:

1. You can query the system with **`euca-describe-images`** to view available VM images and identify the specific VM image you wish to run. Note that VM image names begin with the prefix *`emi-`* followed by an eight-character string (e.g., `emi-EC1410C1`).

   ```
   [bob@client]$ euca-describe-images
    IMAGE  emi-EC1410C1   centos-32/centos.5-3.x86.img.manifest.xml ↲
           admin  available      public  x86_64  machine
    IMAGE  eki-822C1344   kernel-i386/vmlinuz-2.6.28-11-server.manifest.xml ↲
           admin  available      public  x86_64  kernel
    IMAGE  eri-A98C13E4  initrd-64/initrd.img-2.6.28-11-generic.manifest.xml ↲
           admin  available      public  x86_64  ramdisk
   ```

2. Instantiate the VM using the **`euca-run-instances`** command:

   ```
   [bob@client]$ euca-run-instances -k mykey -n 1 emi-EC1410C1
    RESERVATION   r-460007BE     steven.roback   steven.roback-default
    INSTANCE      i-2F930625     emi-EC1410C1    0.0.0.0 0.0.0.0 pending mykey
   ```

3. Enter **`euca-describe-instances`** to verify your VM instance is running and to view information, such as instance id, key name, VM Type, and IP addresses.

   ```
   [bob@client]$ euca-describe-instances
    RESERVATIONr-338206B5  steven.roback  default
    INSTANCE      i-4DCF092C  emi-EC1410C1 192.168.7.24  10.17.0.130 ↲
      running     mykey   0   m1.small      2010-03-15T21:57:45.134Z ↲
      wind        eki-822C1344   eri-BFA91429
   ```

2.4 Authorizing a Security Group

Security Groups are sets of networking rules (in effect a firewall) applied to VMs associated with a group. When you first create a VM instance, it is assigned to a "default" security group that denies incoming network traffic from all sources.

Hence, to allow login and usage of a new VM instance, you must authorize network access to the “default” security group with the **`euca-authorize`** command.

- **`euca-authorize`** `[-P] [-p] [-s] group_name`

```
group_name     Name of the group to add the rule to.
-P             Protocol (“tcp” “udp” or “icmp”)
-p             Range of ports for the rule (“from-to”)
-s             The source subnet for the rule.
```

To authorize a security group:

- Enter **`euca-authorize`**, followed by the name of the security group, and the options of the network rules you wish applied (to the specified security group). In this example, the security group “default” is authorized unlimited network access via `ssh` (TCP, port 22):

```
[bob@client]$ euca-authorize –P tcp –p 22 –s 0.0.0.0/0 default
```

For more information on network rules and security groups, see *Section 5: VM Networking and Security.*

2.5 Logging into a VM Instance

When you create a VM instance, Eucalyptus assigns the instance two IP addresses: a *public IP address* and a *private IP address*. The public IP address provides access to the instance from external network sources; the private IP address provides access to the instance from within the Eucalyptus cloud environment. Note that the two IP addresses may be the same depending on the current networking mode set by the administrator. For more information on Eucalyptus networking modes, see the *Eucalyptus EE Administrator’s Guide*.

To use a VM instance you must log into it via `ssh` using one of the IP addresses assigned to it. You can obtain the instance’s IP addresses using the **`euca-describe-instances`** query as shown in the following example.

To log into a VM instance:

1. Enter **`euca-describe-instances`** to view the IP addresses of your VM instance. Note that the public IP address appears after the VM image name, with the private address immediately following. In the example that follows, the external IP address is 192.168.7.24, and the internal IP address is 10.17.0.130.

```
[bob@client]$ euca-describe-instances
 RESERVATION r-338206B5  steven.roback  default
 INSTANCE      i-4DCF092C  emi-EC1410C1  192.168.7.24  10.17.0.130 ↵
running        mykey   0      m1.small  2010-03-15T21:57:45.134Z ↵
wind   eki-822C1344   eri-BFA91429
```

2. Log into the VM instance via **ssh** using your private key and the external IP address. In this example, we log in via **ssh** using the private key "mykey.private" and the external IP address 192.168.7.24. Note that the prompt "bash-3.2" appears on the next line and it indicates that you have successfully logged into the VM instance.

```
[bob@client]$ ssh -i mykey.private root@192.168.7.24
bash-3.2#
```

For instructions on logging into a Windows VM instance, see *Section 4.5: Logging into a Windows VM instance* (Eucalyptus EE only).

TIP 2: Deleting RSA Host Key Changes

When attempting to log into a VM via `ssh` you may receive a warning message stating that your “Remote Host Identification Has Changed” as shown in the following example:

```
[bob@client]$ ssh -i mykey root@192.168.7.23
  @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  @    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
  @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
  Someone could be eavesdropping on you right now (man-in-the-middle attack)!
  It is also possible that the RSA host key has just been changed.
  The fingerprint for the RSA key sent by the remote host is
  17:91:22:94:7b:13:5c:dd:80:ee:eb:cd:25:73:dc:48.
  Please contact your system administrator.
  Add correct host key in /home/bob/.ssh/known_hosts:11
  RSA host key for 192.168.7.23 has changed and you have requested strict
checking.
 Host key verification failed.
```

This type of message appears when a new instance assumes a known IP address (that is, an IP address previously used by a now-terminated instance). While in general this could be indicative of a “man-in-the-middle attack,” in the cloud setting, this is harmless because public IPs are frequently reused.

You can work around this warning by deleting the line containing the offending key. In the above example, the key is located at line 11 in the file `/home/bob/.ssh/know_hosts.` You can delete this line using the **sed** (stream editor) as shown:

```
[bob@client]$ sed ‘11’ d /home/bob/.ssh/known_hosts
```

2.6 Terminating VM Instances

The **`euca-terminate-instances`** command lets you cancel running VM instances. When you terminate instances you must specify the ID string of the instance(s) you wish to terminate. Note that you can obtain the ID strings of your instances using the **`euca-describe-instances`** query command, as shown in the example:

- **`euca-terminate-instances`** `instance1…instanceN`

  `instance1…instanceN     ID string of instance(s) to terminate.`

To terminate VM instances:

1. Enter **`euca-describe instances`** to obtain the ID of the instances you wish to terminate. Note that an instance ID strings begin with the prefix `i-` followed by an 8-character string:

```
[bob@client]$ euca-describe-instances
 RESERVATION   r-338206B5  steven.roback  default
 INSTANCE      i-4DCF092C  emi-EC1410C1      192.168.7.24   10.17.0.130 ↵
running        mykey   0      m1.small       2010-03-15T21:57:45.134Z ↵
wind  eki-822C1344   eri-BFA91429
```

2. Enter **`euca-terminate-instances`** and the ID string(s) of the instance(s) you wish to terminate:

```
[bob@client]$ euca-terminate-instances i-4DCF092C
 INSTANCE      i-3ED007C8
```

Warning: *Terminating an instance can cause the instance and all items associated with the instance (data, packages installed, etc.) to be lost. Be sure to save any important work or data to Walrus or EBS before terminating an instance.*

Note that depending on the Eucalyptus networking mode currently in use some Euca2ools operations may not be supported (e.g., security groups/elastic IPs are not supported in Static nor System mode). Consult your administrator for details. For more information on these command-line tools, see *Amazon's EC2 Getting Started Guide*[2].

[2] **Amazon's EC2 Getting Started Guide** is available at http://docs.amazonwebservices.com/AWSEC2/2008-12-01/GettingStartedGuide/

Section 3: Managing VM Images

Before you can run instances from VM images that you created or downloaded, you must first prepare these images for use in the cloud by *bundling*[3] them with your credentials, uploading them, and registering them with Eucalyptus. This section introduces Euca2ools commands for performing these and other important image management tasks. For specific instructions on managing Windows VM images, see *Section: 3.7: Managing Windows VM Images* (Eucalyptus EE only).

3.1 Preparing a VM Image (bundle/upload/register)

To enable a VM image as an executable entity, the administrator or user must first prepare the image by bundling a root disk image and kernel/ramdisk pair[4] (in some cases, a ramdisk is optional), uploading the bundled image to Walrus *bucket storage*[5], and registering the data with Eucalyptus. Note that while all users can bundle, upload and register images, only the administrator has the required permissions to upload and register kernels and ramdisks. The administrator also manages which images a user can access. Each image is bundled, uploaded, and registered separately using these Euca2ools commands:

- **`euca-bundle-image`**
- **`euca-upload-bundle`**
- **`euca-register-image`**

The following sections show you how to use these commands to bundle, upload, and register VM images for use as templates from which to instantiate VM instances in Eucalyptus or compatible services (such as AWS).

3.1.1 Bundling Images

The **`euca-bundle-image`** command lets you bundle an image for use with Eucalyptus (or Amazon). Bundling an image splits the image into several image parts and generates an XML manifest file containing metadata referencing the image. When you bundle an image file, the manifest file and image files are stored by default in the $TMPDIR. (In many cases, the $TMPDIR environmental variable is defined to be "/tmp".)

[3] **bundling** a virtual machine image splits the image into multiple image parts to facilitate ease of uploading. It also generates an XML manifest file containing metadata referencing the image, including image parts and kernel, which is used to assemble instances of the image.
[4] **kernel/ramdisk pair –** ramdisk contains drivers that direct the kernel to launch appropriate system files when instantiating a virtual machine.
[5] **bucket storage –** A storage container that accepts objects via PUT and GET commands.

- **euca-bundle-image** [-d] [-c] –i

```
-i              Name of the image file to bundle.

-d              Destination directory for bundled image (default
                directory is /tmp).

-c              Path to user’s PEM encoded certificate.

-k              Path to user’s PEM encoded private key.
```

To bundle an image:

- Enter the `euca-bundle-image` command followed by the name of the image file you wish to bundle. The following example shows the bundling of a Eucalyptus-provided image named `euca-centos-5.3 x86_64/centos.5-3.x86-64.img.` Note that in the absence of a `–d option (`destination directory), the bundled image parts and manifest file are saved to a default $TMPDIR directory:

```
[bob@client]$ euca-bundle-image -i euca-centos-5.3 \
   x86_64/centos.5-3.x86-64.img
  Checking image
  Tarring image
  Encrypting image
  Splitting image...
  Part: centos.5-3.x86-64.img.part.0
  Part: centos.5-3.x86-64.img.part.1
  Part: centos.5-3.x86-64.img.part.2
  Part: centos.5-3.x86-64.img.part.3
  Part: centos.5-3.x86-64.img.part.4
  Part: centos.5-3.x86-64.img.part.5
  Part: centos.5-3.x86-64.img.part.6
  Part: centos.5-3.x86-64.img.part.7
  Part: centos.5-3.x86-64.img.part.8
  Part: centos.5-3.x86-64.img.part.9
  Part: centos.5-3.x86-64.img.part.10
  Part: centos.5-3.x86-64.img.part.11
  Part: centos.5-3.x86-64.img.part.12
  Part: centos.5-3.x86-64.img.part.13
  Part: centos.5-3.x86-64.img.part.14
  Part: centos.5-3.x86-64.img.part.15
  Part: centos.5-3.x86-64.img.part.16
  Part: centos.5-3.x86-64.img.part.17
  Part: centos.5-3.x86-64.img.part.18
  Generating manifest
```

- If necessary, you can specify separate credentials when bundling an image using the `–c` option for the PEM encoded certificate and `–k` option for the PEM encoded private key. In the following example "cert-xyz.pem" and "pk-xyz.pem" represent the user certificate and private key files, respectively.

```
[bob@client]$ euca-bundle-image -i euca-centos-5.3 \
   x86_64/centos.5-3.x86-64.img -c cert-xyz.pem -k pk-xyz.pem
```

To bundle an image for use with Amazon EC2:

1. Locate the Amazon ec2 cert file that is provided as part of the EC2 AMI tools. (This file is generally located in $EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2.pem.)

2. Enter the **`euca-bundle-image`** command followed by the name of the image file to bundle and the PEM encoded certificate and private key provided by Amazon, as follows:

```
[bob@client]$ euca-bundle-image -i euca-centos-5.3 \
   x86_64/centos.5-3.x86-64.img  -u 123456789111 -c cert- \
   abc.pem -k pk-abc.pem --ec2cert \
   $EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2.pem
```

   Make sure that the "cert-abc.pem" and "pk-abc.pem" files in the above example are your Amazon credentials (not your Eucalyptus credentials).

3.1.2 Uploading Images

After bundling your image with `euca-bundle-image`, the next step in adding an image is to upload your bundled image to Walrus (bucket storage) using the **`euca-upload-bundle`** command.

- **`euca-upload-bundle`** `–b bucket –m manifest_path`

```
-b             Name of the bucket to upload to. (The bucket is
               created if it does not yet exist, but in all
               cases a bucket name must be specified).

-m             Path to manifest file for the bundled image.
```

To upload an image:

- Enter the **`euca-upload-bundle`** command followed by the name of the bucket to which the image to be uploaded, followed by the path to the corresponding image manifest file. In the following example we upload the bundle corresponding to the manifest file /tmp/centos.5-3.x86-64.img.manifest.xml to the bucket “mybucket” as shown:

```
[bob@client]$ euca-upload-bundle -b mybucket -m /tmp/centos.5- \
   3.x86-64.img.manifest.xml
  Checking bucket: mybucket
  Uploading manifest file
  Uploading part: centos.5-3.x86-64.img.part.0
  Uploading part: centos.5-3.x86-64.img.part.1
  Uploading part: centos.5-3.x86-64.img.part.2
  Uploading part: centos.5-3.x86-64.img.part.3
  Uploading part: centos.5-3.x86-64.img.part.4
  Uploading part: centos.5-3.x86-64.img.part.5
  Uploading part: centos.5-3.x86-64.img.part.6
  Uploading part: centos.5-3.x86-64.img.part.7
  Uploading part: centos.5-3.x86-64.img.part.8
  Uploading part: centos.5-3.x86-64.img.part.9
  Uploading part: centos.5-3.x86-64.img.part.10
  Uploading part: centos.5-3.x86-64.img.part.11
  Uploading part: centos.5-3.x86-64.img.part.12
  Uploading part: centos.5-3.x86-64.img.part.13
  Uploading part: centos.5-3.x86-64.img.part.14
  Uploading part: centos.5-3.x86-64.img.part.15
  Uploading part: centos.5-3.x86-64.img.part.16
  Uploading part: centos.5-3.x86-64.img.part.17
  Uploading part: centos.5-3.x86-64.img.part.18
  Uploaded image as mybucket/centos.5-3.x86-64.img.manifest.xml
```

3.1.3 Registering Images

After bundling and uploading the image, the final step in preparing an image is to register the image with Eucalyptus using the **`euca-register`** command.

- **`euca-register`** `image_location`

  `image location    Path to the uploaded image (bucket/manifest)`

To register an image:

- Enter the **`euca-register`** command followed by the path name to the bucket containing the image you wish to register. The following example shows the registering of the bundled image referenced by the manifest file **`centos.5-3.x86-64.img.manifest.xml`** that has been uploaded to the bucket "mybucket" (note output showing the newly registered image file **`emi-216211A7`**):

```
[bob@client]$ euca-register mybucket/centos.5-3.x86- \
   64.img.manifest.xml
 IMAGE  emi-216211A7
```

3.2 Associating Kernels and Ramdisks with VM Instances

To run an instance of a VM image, the image must be associated with a qualified kernel and ramdisk image. You can view available kernel and ramdisk images using the **`euca-describe-images`** command. Note that kernel images begin with the prefix **`eki-`** followed by an eight-character string (e.g., `eki-6CDC12F6`); ramdisk images begin with the prefix **`eri-`** followed by and eight-character string (e.g., `eri-A98C13E4`). For more information on `euca-describe-images`, see *Section 2.1: Querying the System* and *Section 2.3: Instantiating a Virtual Machine.*

To associate a kernel and ramdisk image with a VM instance:

There are three ways to associate a kernel and ramdisk with a VM instance:

1. A user may associate a specific kernel/ramdisk identifier with an image when bundling the image with the **`euca-bundle-image`** command by specifying **`--kernel`** and **`--ramdisk`** options:

```
[bob@client]$ euca-bundle-image -i centos.5-3.x86-64.img \
   --kernel eki-6CDC12F6 --ramdisk eri-A98C13E4
```

2. A user may specify a qualified kernel and ramdisk along with the image id when running an instance with the **`euca-run-instance`** command:

```
[bob@client]$ euca-run-instances emi-6D0D133C \
   --kernel eki-6CDC12F6 --ramdisk eri-A98C13E4
```

3. The administrator can set 'default' pre-registered kernel/ramdisk identifiers to be used if a kernel/ramdisk is unspecified by either of the above options. To do this, log into the administrative interface at **https://<your.cloud.server>:8443/**, click on the 'Configuration' tab, and add a default kernel/ramdisk. Note that kernel images begin with the prefix *eki*- followed by an 8-character string, while ramdisk images begin with the prefix *eri*- followed by an 8-character string.

3.3 Deleting Images

To delete an image from Eucalyptus you must first “deregister” the image using the **`euca-deregister`** command. You can then delete the image from the bucket to which it was uploaded using the **`euca-delete-bundle`** command.

- **`euca-deregister`** `image_id`

`image_id`	`Unique identifier of the image to be deregistered.`

- **`euca-delete-bundle`** `–b`

`-b`	`Name of the bucket to delete bundle from.`
`-m`	`Path to manifest file.`
`-p`	`Filename prefix for bundled files.`
`--clear`	`Delete the bucket containing the image`

To delete an image:

1. Enter the **`euca-deregister`** command followed by the image id of the image you wish to deregister. The following example shows the deregistering of the image file `emi-6D0D133C`:

```
[bob@client]$ euca-deregister emi-6D0D133C
```

2. Enter **`euca-delete-bundle`** followed by the name of the bucket and the prefix of the image file you wish to remove. The following example shows the removal of the image `centos.5-3.x86-64.img` from the bucket `mybucket`:

```
[bob@client]$ euca-delete-bundle -b mybucket -p centos.5-3.x86- \
   64.img
```

or

Remove both the image and the bucket by adding the **`--clear`** option:

```
[bob@client]$ euca-delete-bundle -b mybucket -p centos.5-3.x86- \
   64.img --clear
```

3.4 Downloading an Image

Bundled images that have been uploaded can also be downloaded and deleted from the cloud. You can download bundled images using the **`euca-download-bundle`** command.

- **`euca-download-bundle`** –b

```
–b            Name of bucket to download from.
```

To download an image:

- Enter the **`euca-download-bundle`** command followed by the **`–b`** option and the name of the bucket containing the image(s). The following example shows images downloaded from the bucket “mybucket”:

```
[bob@client]$ euca-download-bundle -b mybucket
 Downloading centos.5-3.x86-64.img.manifest.xml
 Downloading initrd.img-2.6.28-11-generic.manifest.xml
 Downloading vmlinuz-2.6.28-11-generic.manifest.xml
 Downloading centos.5-3.x86-64.img.part.0
 Downloading centos.5-3.x86-64.img.part.1
 Downloading centos.5-3.x86-64.img.part.2
 Downloading centos.5-3.x86-64.img.part.3
 Downloading centos.5-3.x86-64.img.part.4
 Downloading centos.5-3.x86-64.img.part.5
 Downloading centos.5-3.x86-64.img.part.6
 Downloading centos.5-3.x86-64.img.part.7
 Downloading centos.5-3.x86-64.img.part.8
 Downloading centos.5-3.x86-64.img.part.9
 Downloading centos.5-3.x86-64.img.part.10
 Downloading centos.5-3.x86-64.img.part.11
 Downloading centos.5-3.x86-64.img.part.12
 Downloading centos.5-3.x86-64.img.part.13
 Downloading centos.5-3.x86-64.img.part.14
 Downloading centos.5-3.x86-64.img.part.15
 Downloading centos.5-3.x86-64.img.part.16
 Downloading centos.5-3.x86-64.img.part.17
 Downloading centos.5-3.x86-64.img.part.18
 Downloading initrd.img-2.6.28-11-generic.part.0
 Downloading vmlinuz-2.6.28-11-generic.part.0
```

3.5 Deleting a Bundled Image

The **`euca-delete-bundle`** command lets you delete bundled images from Eucalyptus.

To delete a bundled image:

- Enter **`euca-delete-bundle`** followed by the –b option and the name of the bucket containing the bundle you wish to delete. In the following example bundled images contained in bucket “mybucket” are deleted:

```
[bob@client]$ euca-delete-bundle -b mybucket
```

Note that if you wish to delete a specific bundle, you can also specify a manifest file reference using the optional **-m** option.

- To delete the bucket after deleting the bundled image add the `--clear` option, as shown:

```
[bob@client]$ euca-delete-bundle -b mybucket --clear
```

Note that a bucket can only be deleted when empty.

3.6 Unbundling an Image

If you wish to share a previously bundled image with another user, you must first unbundle the image, then bundle, upload, and register the image again with the other user's credentials. Eucalyptus provides the **euca-unbundle** command for unbundling a previously bundled image.

- **euca-unbundle** –m

```
–m             Path to the manifest file for the bundled image.
```

To unbundle an image:

- Enter the **euca-unbundle** command followed by the path to the manifest file for the bundled image. The following example shows the unbundling of the bundled image file referenced by the manifest file `centos.5-3.x86-64.img.manifest.xml`:

```
[bob@client]$ euca-unbundle -m centos.5-3.x86- \
   64.img.manifest.xml
```

3.7 Managing Windows VM Images (Eucalyptus EE only)

Managing Windows VM images in Eucalyptus involves the same basic processes as managing Linux-based VMs with a few distinctions, as described in the following sections.

3.7.1 Preparing Windows VM Images

Before you begin preparing the Windows VM image, confirm that you have installed the latest version of Euca2ools-EE client tools. If necessary, see your administrator for assistance*.*

To prepare a Windows VM image for use in Eucalyptus EE, follow the standard **bundle/upload/register** steps used for preparing Linux VM images (see *Section 3.1: Preparing a VM Image*), keeping in mind these caveats:

1. The filename of your Windows image must begin with the string 'windows.' For example:

   `windows.my2008server.img`

2. Kernel and ramdisk images **are not used** when bundling Windows images.

3. Bundling a Windows image can take several minutes depending on the size of your image and the configuration of the machine on which you are running the bundling task.

   For instructions on uploading and registering VM images, see *Section 3.1.2: Uploading Images* and *Section 3.1.3: Registering Images,* respectively*.*

   For instructions on creating a Windows VM image for use in Eucalyptus, see the *Eucalyptus EE Administrator Guide, Appendix A: Creating a Windows image for Eucalyptus.*

3.7.2 Bundling Windows VM Instances

Euca2ools provides EC2-compatible commands for generating new Windows VM images directly from running Windows VM instances (Eucalyptus EE with Windows VM support only) as follows:

The **`euca-bundle-instance`** command lets you bundle a new Windows VM image from a running Windows VM instance directly to Walrus storage. Using `euca-bundle-instance` is an efficient way to generate modified Windows VMs. You can spin up a Windows VM instance from an existing Windows VM image, modify it as needed, then save the modified image to Walrus storage, where it is immediately available for registering and running with the modifications already in place.

- **`euca-bundle-instance`** `–b instance_id –p –o -w`

```
-b                Name of bucket to save new image to.
```

instance_id	Unique ID tag of running instance from which to bundle new image.
-p	Image prefix
-o	Access key id of the user the image should be uploaded to Walrus as.
-w	Secret key id of the user the image should be uploaded as.

To bundle a Windows VM instance:

Enter the the **`euca-bundle-instance`** command followed by the ID tag of the instance from which you wish to bundle the new image, along with the `-b` option and the name of the bucket where you wish to save the new image; -p followed by the image prefix; -o followed by the access key id of the user; and –w followed by the secret key id of the user.

The following example shows the bundling of a new Windows VM image from an existing Windows VM instance with ID `i-12c4af6a` to the bucket `mybucket`, with image prefix `windows,` access key “ACCESS,” and secret key “SECRET”:

```
[bob@client]$ euca-bundle-instance  –b mybucket i-12c4af6a \
   -p windows –o ACCESS –w SECRET
```

Note that you can also query the progress of a ‘bundle-instance’ operation using the **`describe-bundle-task`** command, or, you can cancel an in-progress ‘bundle-instance’ operation using the **`cancel-bundle-task`** command.

For instructions on creating Windows VM images for use in Eucalyptus, see the *Eucalyptus EE 2.0 Administrator’s Guide; Appendix A: Creating a Windows Image for Eucalyptus*

Section 4: Working with VM Instances

Euca2ools provides commands for working with VM instances generated from VM images. This section shows you the most frequently used commands for running and controlling Linux and Windows-based VM instances. For more information on running VM instances, see *Section 2.3: Running a VM Instance.*

4.1 Querying Instances

The **`euca-describe-instances`** command lets you view information about your currently running instances.

To query running instances:

- Enter **`euca-describe-instances`** to view information about all currently running instances:

  ```
  [bob@client]$ euca-describe-instances
  ```

- To get information about a specific instance, enter **`euca-describe-instances`** followed by the instance id. The following example shows a query for information on an instance with id tag `i-43035890`:

  ```
  [bob@client]$ euca-describe-instances i-43035890
  ```

4.2 Running Instances

The **`euca-run-instances`** command lets you deploy instances of VM images that have been previously uploaded to Eucalyptus.

To run a VM instance:

- Enter the **`euca-run-instances`** command, followed by your keypair name, and the image id of the instance you wish to run, as well as the image id of qualified kernel and ramdisk. For example, to run an instance of the image with id "emi-53444344," kernel "eki-34323333," ramdisk "eri-33344234" and keypair "mykey" enter:
-

```
[bob@client]$ euca-run-instances -k mykey --kernel eki-34323333 \
   --ramdisk eri-33344234 emi-53444344
```

Note that you can simultaneously launch multiple instances by specifying the number of instances you wish to run with the `-n` option. For more information on running instances with the **`euca-run-instances`** command (including sample output), see *Section 2.3: Running a Virtual Machine.*

For instructions on logging into a Linux VM instance, see *Section 2.5: Logging into a VM Instance.* For instructions on logging into a Windows VM instance, see *Section 4.5: Logging into a Windows VM Instance* (Eucalyptus EE only).

4.3 Shutting Down Instances

You can shut down running VM instances using the **`euca-terminate-instances`** command. For example, to terminate the instance "i-34523332" enter the following:

```
[bob@client]$ euca-terminate-instances i-34523332
```

4.4 Rebooting Instances

Rebooting preserves the root filesystem of an instance across restarts.
You can reboot running VM instances using the **`euca-reboot-instances`** command. For example, to reboot the instance "i-34523332," enter:

```
[bob@client] euca-reboot-instances i-34523332
```

4.5 Logging into a Windows VM Instance (Eucalyptus EE only)

Users log into Windows VM instances (in Eucalyptus and EC2) via a Remote Desktop Protocol (RDP) client, which prompts the user for a login name and password. Windows VM instances are configured by default with a single user (named 'Administrator') and a random password generated at boot time. Hence, before you can log into a Windows VM instance via RDP, you must retrieve the random password generated at boot time using the **`euca-get-password`** command.

- **`euca-get-password`** `instance_id –k`

  ```
  instance_id    ID tag for Windows instance.
  ```

`-k`	`Name of the private key file that corresponds to the keypair.`

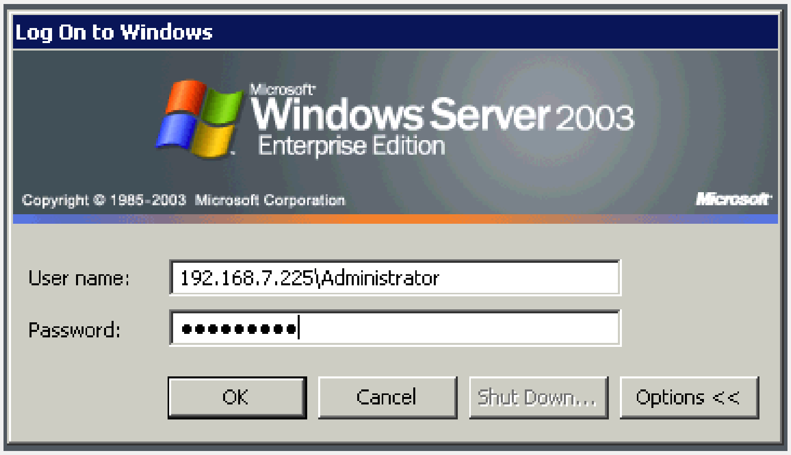
To log into a Windows VM instance:

1. Enter the **`euca-get-password`** command followed by the unique id tag of the Windows VM instance and the `–k` option with the name of private key file that corresponds to your credential keypair. In the following example we retrieve the password for a Windows VM instance with id tag `i-5176095D` and private key file name 'mykey.private.' Note the password for the Windows VM is returned as **`Y4rX4Az2`**.

```
[bob@client]$ euca-get-password i-5176095D –k mykey.private
 Y4rX4Az2
```

2. Log into the RDP client using the public (external) IP address associated with the running Windows VM instance. (You can view the IP addresses associated with your Windows instance using the `euca-describe-instances` query command).

3. At the 'Log On to Windows' prompt, prepend the user name 'Administrator' to the public IP address of the instance, and enter the Password that you retrieved with `euca-get-password`, as shown:



User name: 192.168.7.225\Administrator

Password: •••••••••

You should now be logged in and ready to use your Windows VM instance. For instructions on logging into Linux-based VM instances, see *Section 2.5: Logging into a VM Instance.*

Section 5: VM Networking and Security

Eucalyptus provides networking modes that administrators can configure according to the network and security needs of the enterprise. Depending on the current networking mode configuration, users may have access to such features as *elastic IPs,* which are public (external) IP addresses that users can reserve and dynamically associate with VM instance; and *security groups*, which are sets of firewall rules applied to VM instances associated with the group. Euca2ools provides a means for users to interact with these features with commands for allocating and associating IP addresses, as well as creating, deleting, and modifying security groups.

5.1 Allocating and Associating IP Addresses

The **`euca-allocate-address`** and **`euca-associate-address`** commands let you allocate IP addresses and associate public IP addresses with instances, respectively.

- **`euca-allocate-address`** `(Automatically allocates an IP address)`
- **`euca-associate-address`** `-i`

```
-i              Unique identifier for running instance to
                associate address with.
```

To allocate and associate an IP address:

1. Enter **`euca-allocate-address`** to allocate an IP address:

```
[bob@client]$ euca-allocate-address
 ADDRESS       192.168.17.103
```

2. Enter **`euca-associate-address`** followed by the –i option with a specified instance id and allocated IP address. In the following example the above allocated IP address is associated with the instance id i-56785678, as shown:

```
[bob@client]$ euca-associate-address -i i-56785678 192.168.17.103
 ADDRESS       192.168.17.103 i-2F930625
```

5.1.1 Disassociating and Releasing Addresses

You may use **`euca-disassociate-address`** and **`euca-release-address`** to disassociate an IP address from an instance and to release the IP address to the global pool, respectively.

- **`euca-disassociate-address`** `IP`

  ```
  IP             IP address to disassociate.
  ```

- **`euca-release-address`** `IP`

  ```
  IP             IP address to release.
  ```

To disassociate and release an IP addresses:

1. Enter **`euca-disassociate-address`** followed by the IP address you wish to disassociate:

   ```
   [bob@client]$ euca-disassociate-address 192.168.17.103
    ADDRESS       192.168.17.103
   ```

2. Enter **`euca-release-address`** followed by the IP address you wish to release:

   ```
   [bob@client]$ euca-release-address 192.168.17.103
    ADDRESS       192.168.17.103
   ```

5.2 Creating a Security Group

Security groups let you control network access to instances by applying network rules (in effect a firewall) to VM instances associated with a group. You can create a security group using the **`euca-add-group`** command.

- **`euca-add-group`** `–d group_description group_name`

  ```
  -d             “group_description” (in quotes)
  group_name     Name of security group.
  ```

To create a security group:

- Enter the **`euca-add-group`** command followed by the –d option and a description (in quotes) and the name of the new security group. In the following example we create a new security group described as “newgroup” and named mygroup:

  ```
  [bob@client]$ euca-add-group -d “newgroup” mygroup
  GROUP   mygroup newgroup
  ```

You can also create a security group when running VM instances with the **`euca-run-instances`** command using the `-g` option. Note that security-groups rules only apply to incoming traffic thus all outbound traffic is permitted.

5.2.1 Authorizing Security Group Rules

By default, a security group prevents incoming network traffic from all sources. You can modify network rules and allow incoming traffic to security groups from specified sources using the **`euca-authorize`** command.

- **`euca-authorize`** `[-P] [-p] [-s] group_name`

```
group_name     Name of the group to add the rule to.
-P             Protocol (“tcp” “udp” or “icmp”)
-p             Range of ports for the rule (“from-to”)
-s             The source subnet for the rule.
```

To authorize security group rules:

- Enter **`euca-authorize`**, the name of the security group, and the options of the network rules you wish applied to the specified security group.

  For example, to allow all incoming SSH (port 22) traffic access to the security group "mygroup," use the `euca-authorize` command and specify options for a protocol (tcp), a port (22), and a CIDR[6] source network (0.0.0.0/0, which refers to any source):

```
[bob@client]$ euca-authorize -P tcp -p 22 -s 0.0.0.0/0 mygroup
 GROUP  mygroup ↵
PERMISSION     mygroup ALLOWS  tcp     22      22      FROM    CIDR
```

  or

  Instead of specifying a CIDR source, you can specify another security group, which allows port 22 connections to and from the other security group. In the following example we allow access to the security group “mygroup” from the another security group named “someothergroup”:

```
[bob@client]$ euca-authorize --source-group someothergroup \
       --source-group-user someotheruser -P tcp -p 22 mygroup
```

[6] **CIDR** – CIDR (Classless Inter-Domain Routing) is a scalable methodology for allocating IP addresses and routing Internet Protocol packets.

5.2.2 Revoking Rules from Security Groups

To revoke rules from a security group, use the **`euca-revoke`** command.

- **`euca-revoke`** `[-P] [-p] [-s] group_name`

  ```
  group_name     Name of the group to add the rule to.
  -P             Protocol (“tcp” “udp” or “icmp”)
  -p             Range of ports for the rule (“from-to”)
  -s             The source subnet for the rule.
  ```

To revoke security group rules:

- Enter **`euca-revoke`** followed by the networking options you wish to revoke. The following example shows the revocation of the networking rules authorized for the security group “mygroup” in the previous example:

  ```
  [bob@client]$ euca-revoke -P tcp -p 22 -s 0.0.0.0/0 mygroup
  ```

5.2.3 Deleting a Security Group

- The **`euca-delete-group`** command lets you delete security groups. For example, to delete the security group "mygroup,” input the following:

  ```
  [bob@client]$ euca-delete-group mygroup
  ```

Section 6: Dynamic Block Volumes and Snapshots

Eucalyptus lets you create dynamic block volumes, which are analogous to raw block storage devices. Once you have created a volume, you can attach it to a VM instance, create a filesystem on top of the attached volume (aka formatting), and mount it inside the instance as a block device. You can also detach volumes, delete volumes, create instantaneous snapshots[7] from volumes, and create new volumes from snapshots. (Eucalyptus' dynamic block volumes emulate and are interface compatible with Amazon's EBS (Elastic Block Store). Hence, you can use either Euca2ools or EC2 commands to interact with Eucalyptus' dynamic block volumes.)

6.1 Creating a Volume

You can create a new volume using the **`euca-create-volume`** command. When you create a volume you must specify a volume size (e.g., 1 GB) and availability zone[8] name. Users can create volumes of any size up to the maximum size limit set by the system administrator.

- **`euca-create-volume`** `--size | --snapshot -z`

```
--size          Size of the volume (in GB)

--snapshot      Snapshot ID to create volume from.(Specify
                either size or snapshot, not both).

-z              Availability zone to create volume in.
```

To create a new volume:

1. Query the system to view information about current availability zones and identify the name of the availability zone where you wish the new volume to reside.

```
[bob@client]$ euca-describe-availability-zones
 AVAILABILITYZONE      wind    192.168.7.17
```

2. Enter the **`euca-create-volume`** command followed by the `--size` option specifying the size of the volume you wish to create, and the `–z` option specifying the name of the availability zone where you wish the volume to reside. In the following example we create a 1 GB volume in the availability zone “wind”:

[7] **Snapshot-** A snapshot is a copy of the total existing state of a volume. In Eucalyptus, you can create snapshots from a volume then create an entirely new volume from that snapshot.
[8] **Availability zones** refer to subsets of the cloud that share a local area network. Each availability zone has its own cluster controller and storage controller.

```
[bob@client]$ euca-create-volume -s 1 -z wind
 VOLUME vol-C1AE0988    1      available      ↵
        2010-05-05T02:39:25.139Z
```

6.2 Creating a Volume from a Snapshot

You can also use the **euca-create-volume** command to create a volume from an existing snapshot using the `–snapshot` option.

To create a volume from a snapshot:

- Enter **euca-create-volume** followed by the –snapshot option and the ID number of the snapshot from which you wish to create the volume. In the following example we create a volume from the snapshot "snap-33453345" in the zone "myzone":

```
[bob@client]$ euca-create-volume --snapshot snap-33453345 -z \
        myzone
```

6.3 Attaching a Volume to an Instance

You can attach block volumes to instances using the **euca-attach-volume** command.

- **euca-attach-volume** `–i –d volume_id`

-i	Unique ID of instance running instance to attach the volume to.
-d	Local device name (inside guest VM).
Volume_id	Unique ID for volume to attach.

To attach a volume to an instance:

- Enter the euca-attach-volume command followed by the –i option with the ID of the instance, the -d option with the local block device name (to be used inside the VM), and the ID of the volume to attach. In the following example we attach a volume "vol-33534456" to the instance "i-99838888" at "/dev/sdb":

```
[bob@client]$ euca-attach-volume -i i-99838888 -d /dev/sdb \
        vol- 33534456
```

Note that a volume can be attached to only one instance at a given time.

6.3.1 Detaching a Volume

- To detach a previously attached volume, use the **`euca-detach-volume`** command. For example, to detach the volume "vol-33534456," enter the following:

  ```
  [bob@client]$ euca-detach-volume vol-33534456
  ```

  WARNING: *You must detach a volume before terminating an instance or deleting a volume. If you fail to detach a volume, it may leave the volume in an inconsistent state and important data might be lost.*

6.4 Deleting a Volume

- To delete a volume, use the **`euca-delete-volume`** command. For example, to delete the volume "vol-33534456" enter the following:

  ```
  [bob@client]$ euca-delete-volume vol-33534456
  ```

  Note that you can only delete volumes that are not currently attached to instances.

6.5 Creating a Snapshot

- The **`euca-create-snapshot`** command lets you create an instantaneous snapshot of a volume. A volume could be attached and in use during a snapshot operation. For example, to create a snapshot of the volume "vol-33534456," enter the following command:

  ```
  [bob@client]$ euca-create-snapshot vol-33534456
  ```

6.5.1 Deleting a Snapshot

- To delete a snapshot, use the **`euca-delete-snapshot`** command. For example, to delete the snapshot snap-33453345, input the following command:

  ```
  [bob@client bob]$ euca-delete-snapshot snap-33453345
  ```

Appendix A: Installing and Using ElasticFox

ElasticFox is a Firefox web browser plug-in that provides a graphical interface for managing both Eucalyptus and Amazon EC2 accounts. You can download Elasticfox at https://s3.amazonaws.com/ec2-downloads/elasticfox.xpi

Section A.1: Installing ElasticFox

To install ElasticFox:

1. Download and save the elasticfox.xpi file from the website listed above.
2. Click on the Firefox Tools menu, and choose Add-ons.
3. Drag and drop elasticfox.xpi into the Add-ons window.
4. Click Install.
5. Restart Firefox.

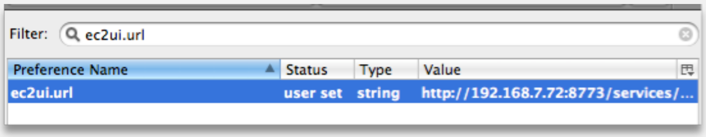
Section A.2: Configuring ElasticFox

ElasticFox comes pre-configured to communicate with AWS. Thus it is necessary to reconfigure ElasticFox to communicate with Eucalyptus, as follows:

To configure ElasticFox for Eucalyptus:

1. Open Firefox.
2. In the main menu select Tools > ElasticFox.

   Elasticfox opens and asks if you would like to provide your AWS credentials.
3. Click No.
4. In the Firefox URL bar, enter: about:config
5. In the Filter field, enter: ec2ui.url (**Figure A.1**).



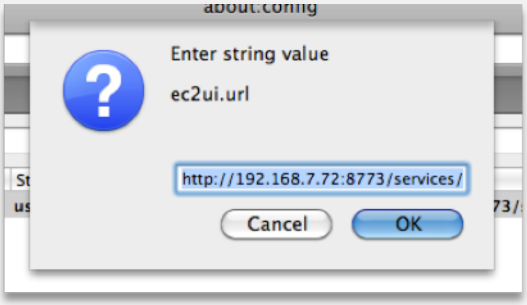
Filter: ec2ui.url

Preference Name	Status	Type	Value
ec2ui.url	user set	string	http://192.168.7.72:8773/services/...

Figure A.1

6. Double-click and replace the default URL with:

http://<ip of cloud controller>/services/Eucalyptus (**Figure A.2**).



Enter string value

ec2ui.url

http://192.168.7.72:8773/services/

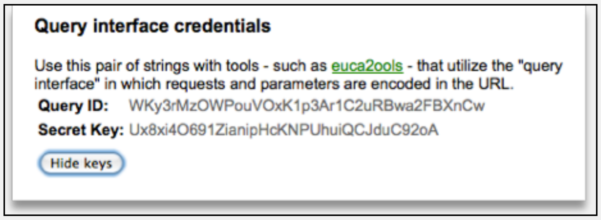
Figure A.2

Section A.3: Providing Credentials

ElasticFox requires your Eucalyptus query credentials to communicate with the various web services.

To provide your Eucalyptus credentials to ElasticFox:

1. Log into the Eucalyptus Cloud Controller administrative interface and fetch your query credential values (**Figure A.3**):



Query interface credentials

Use this pair of strings with tools - such as euca2ools - that utilize the "query interface" in which requests and parameters are encoded in the URL.

Query ID: WKy3rMzOWPouVOxK1p3Ar1C2uRBwa2FBXnCw

Secret Key: Ux8xi4O691ZianipHcKNPUhuiQCJduC92oA

Figure A.3

2. Copy and paste the Query ID and Secret Key values into a temporary location such as Notepad, or open the page in a new Firefox window.

3. Open ElasticFox (**Figure A.4**):

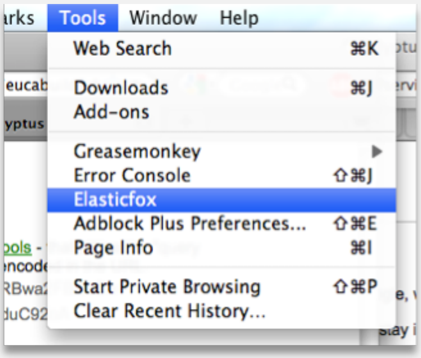
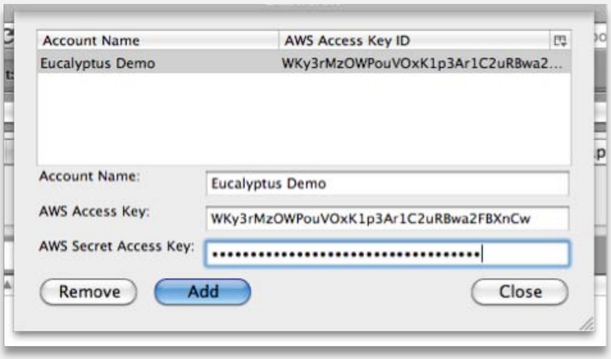


Figure A.4

4. Click on the **Credentials** button on the ElasticFox toolbar. Enter a name for the account, and then paste in the two values: the access key, and the secret key, and press Add (**Figure A.5**).



Account Name	AWS Access Key ID
Eucalyptus Demo	WKy3rMzOWPouVOxK1p3Ar1C2uRBwa2...

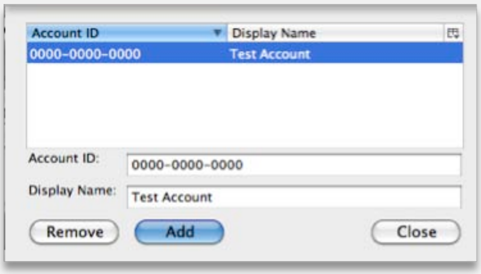
Account Name: Eucalyptus Demo

AWS Access Key: WKy3rMzOWPouVOxK1p3Ar1C2uRBwa2FBXnCw

AWS Secret Access Key: ••

Figure A.5

5. Click on **Account IDs**. The actual account ID is irrelevant to Eucalyptus, so enter 0000-0000-0000 and a name for the ID, and press Add (**Figure A.6**).



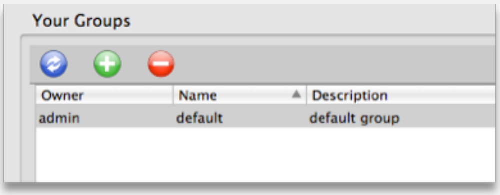
Account ID	Display Name
0000-0000-0000	Test Account

Account ID: 0000-0000-0000

Display Name: Test Account

Figure A.6

6. Click on **Security Groups** and press the refresh button (blue). You should see one item in the list (**Figure A.7**).



Owner	Name	Description
admin	default	default group

Figure A.7

ElasticFox should now be properly configured for use with your Eucalyptus cloud.

Appendix B: Overview of Cloud Computing

Today's IT organizations face a difficult challenge: They must continue to deliver the productivity-enhancing technical innovation that gives the enterprise its competitive edge, while simultaneously streamlining operations in response to severe budgetary constraints. As a result, rather than replacing infrastructure with expensive purchases, IT professionals are pursuing innovative technologies, such as cloud computing, to improve IT efficiency and enhance the performance and longevity of existing IT resources.

B.1 What is Cloud Computing?

Cloud computing is the delivery of computer resources through a Web service interface (e.g., SOAP or REST) on an as-needed basis. The term "cloud" refers to the organization of the underlying physical infrastructure remaining opaque (not visible) to the end user. In other words, cloud computing gives a user access to computer resources (i.e. machines, storage, operating systems, application development environments, application programs) over a network (Internet or intranet) through Web services, while the actual physical location and organization of the equipment hosting these resources—be it in the next room or spread across the globe—is not necessarily known to the user. As such, these resources appear to the user as being "in the cloud."

Improved Efficiencies

All types of cloud computing—public clouds, private clouds, and hybrid clouds—share an intrinsic set of characteristics. These characteristics help define cloud computing as a uniquely efficient model for delivering computer resources:

- *Underlying organization of physical computing resources unknown to end user, or "in the cloud."*
- *Computing resources available as-needed.*
- *Web-services-based interfaces.*
- *Dynamic Scalability (a.k.a. "Elasticity").*
- *Automated resource provisioning.*
- *Self-service.*

B.2 The Efficiency Benefits of Cloud Computing

The above characteristics of cloud computing work synergistically to bring a high level of efficiency to IT operations and significant benefits to the enterprise, as follows:

Computing Resources Located "In the Cloud"

Regardless of the type of cloud, public, private, or hybrid, redesigning the IT organization to provide computing resources from a cloud can provide substantial efficiency benefits. In the case of public clouds, the IT organization simply facilitates access to the cloud for users through Web service protocols. Thus capital that may otherwise be invested into building an internal data center infrastructure goes instead to paying for rental of the public cloud. In the case of private and hybrid clouds, existing data center infrastructure is highly optimized by consolidation and virtualization. This translates into reduced power consumption and enhanced server longevity, and thus reduced infrastructure-related expense.

Computing Resources Available "as-needed"

With computing resources available on an "as-needed" basis, cloud-computing customers simply use the resources they need when they need them. In the case of public clouds, such as AWS (Amazon Web Services), there are no proprietary license agreements required. Customers simply pay via credit card on a per time-unit basis. When services are no longer required, they are simply discontinued and no further charges incurred. In the case of private clouds, as-needed means that resources are provided to users when they are needed, then efficiently redistributed to other users when need ceases.

Web-services-based Interfaces

Using standard Web-based interfaces enables users to easily access computing resources over common networks using universally accepted network communication protocols. These protocols are highly compatible with existing networking devices. Thus cloud systems are designed for efficient operation over both the public Internet and private enterprise intranets.

Dynamic Scalability (a.k.a. Elasticity)

> In a cloud system, computing resources are highly scalable. This means that many resources can be committed to a single application or user. *Elasticity* refers to the ability to scale up (add resources) or scale down (decommission them) "on the fly," dynamically in response to changing application or user needs. Elasticity gives the enterprise the ability to repurpose resources immediately as demand (generated by users or auto-scaling applications) fluctuates.

Automated Resource Provisioning

> Clouds perform the distribution of computer resources immediately and automatically without direct human intervention. Thus users can quickly access the resources they require, while IT staff is freed from many repetitive acquisition, configuration, and servicing tasks.

Self-service

> In a cloud setup, the end user directly selects the computer resources desired through a self-service Web interface. This gives the user direct control over computer resource deployment and configuration, and it helps assure that the users needs are more closely met. Self-service also frees the user from waiting in long IT service queues, and can thus enhance the productivity of the enterprise workforce.

In addition, **SLAs** (service-level agreements) combined with auditing and metrics tools help assure that users receive the computing resources expected.

B.3 Cloud Types (public, private, and hybrid)

While all cloud types share the same fundamental characteristics, there are some important distinctions. Each cloud type addresses a different set of needs for an organization:

Public Clouds

Public clouds provide computing resources over the Internet to the general public on a pay-as-you-go basis. Computing resources are provided to customers through self-serve interfaces. Customers generally "rent" access to resources as needed, on a per time-unit basis (e.g., per hour), or in some cases through subscription. Payments are generally made online via credit card.
Public clouds address the needs of an organization requiring access to highly scalable computing resources on a temporary or periodic basis. The main advantage of the public cloud is the ability to acquire access to high-quality pools of

resources immediately and with minimal capital investment. The public cloud can thus serve the needs of individuals, small companies, or startups with limited finances; or any organization with periodic spikes in demand that require access to large-scale computing resources.

The disadvantage of the public cloud is that expenses accrue over time, and for an organization with significant computing needs the public cloud can become expensive. In addition, though encryption protocols can be used both for communication and storage, the current virtual machine technology requires that code and data be made available un-encrypted when it is executed and processed respectively. That is, if the data is processed in a public cloud, it must be "in the clear" when it is processed, as must the code that processes it, even if both are transmitted and stored in encrypted form. This requirement makes it impossible to create a "chain of trust" that does not involve the public cloud provider; hence public clouds are potentially inappropriate for particularly sensitive data and applications. Well-known public cloud providers include Amazon Web Services (AWS), Google Apps, Microsoft, IBM, Rackspace, and Salesforce.com.

Private Clouds

Private clouds provide computing resources to users within an enterprise over an intranet. Computing resources are distributed through an automated provisioning system accessed by users via a self-serve interface, just as they are in public clouds.

Private clouds address the needs of organizations requiring a more efficient IT infrastructure as well as a high level of security over sensitive data. In a private cloud, data center resources are consolidated through virtualization and highly optimized to reduce operational expense. Automated resource provisioning and user self-service let the IT workforce shift focus to planning and customization projects that support a more productive enterprise workforce. And because a private cloud operates within the security perimeter of the enterprise, it uses the physical and electronic security measures in place in the data center to ensure the security of code and data.

Private clouds are limited however by the resource capacity of the enterprise's data center, which might be an issue during large spikes in demand. In this case, a hybrid cloud that offers the ability to "spill out" or burst into one or more public clouds is a viable option.

In addition, it is important to note that due to the entrenched nature of many IT organizational practices, some resistance could be met by IT staff to the fundamental process changes required to successfully implement a private cloud within an enterprise. While private clouds share many common characteristics with virtualized data centers, their scaling and self-service characteristics often require organizational process and policy changes where simply virtualizing data center resources does not.

Hybrid Clouds

Hybrid clouds combine elements of both private and public clouds. In a hybrid cloud, users inside an organization with a private cloud can breakthrough the boundaries of the enterprise firewall or “cloudburst” to access additional computing resources in the public cloud.

Hybrid clouds address the needs of organizations that require periodic access to highly scalable computing resources beyond the capacity of the enterprise’s existing data center infrastructure. Organizations benefit from the enhanced efficiency and security of the private cloud, but are prepared for large and unexpected spikes in demand.

B.4 Cloud Service Styles (SaaS, PaaS, IaaS)

Computing resources accessed via any type of cloud are usually grouped into service “styles” according to the kind of resources provided—software, platform, or infrastructure. These styles correspond to different levels of programming and/or operational abstraction that applications deployed in each style must use. The three most common cloud service styles are generally referred to by the following acronyms:

- **SaaS** (Software as a Service): Provides network accessible access to software application programs.
- **PaaS** (Platform as a Service): Provides network accessible access to a programming or runtime environment with scalable compute and data structures embedded in it.
- **IaaS** (Infrastructure as a Service): Provides access to virtualized computer hardware resources, including machines, network resources, and storage.

Many public cloud companies tailor their market offerings to address specific needs of customers according to these service styles. As a SaaS example, Salesforce.com provides access to hosted business application software packages through network facing APIs. Google’s App Engine (GAE) is a PaaS offering that allows users to upload Python or Java programs so that they can take advantage of Google’s Big Table infrastructure through a set of hosted interfaces. Finally, Amazon.com’s Amazon Web Services (AWS) is an example of IaaS style cloud computing, offering time-limited rental of virtualized machine collections via network accessible web services.

B.5 Private Clouds: The Benefits of Automated Self-service

Automated resource provisioning and user self-service allow users to access computer resources as needed through a simple Web-interface, without assistance from IT staff. Giving users direct control over resource provisioning helps ensure that users needs are more closely met. And it promotes smooth workflow by minimizing time spent by users waiting for help in long service queues.

For IT staff, automated resource provisioning and self-service means a reduction in time spent servicing user requests, as well as a degree of freedom from manual acquisition, setup, configuration, and service tasks. IT staff can focus instead on more innovative pursuits that further enhance efficiency and user productivity, including such things as data center capacity planning, enhancing automation, image management, maintaining archival provenance and automated legacy support, and providing users with customized Web interfaces and resource allocation options.

Automated resource provisioning and self-service deliver efficiency benefits through outsourcing responsibility for user service from IT staff directly to the end user. This increases the "concurrency "of the IT service system, which significantly improves its efficiency.

Another concept with roots in the e-commerce business model, concurrency refers to the simultaneous processing of multiple user requests. Increasing concurrency in the IT service system allows for multiple user requests to be processed immediately and simultaneously and thus reduces or eliminates long IT service queues. For example, if ten users each have a provisioning request that takes IT one-hour to complete, a user might wait ten hours before receiving help—then wait another hour for the task to be completed. With the increased concurrency of a self-service automated provisioning system, all ten users can be serviced immediately and simultaneously.

B.6 The Role of Open Source

Seeking more cost-effective options that provide greater adaptability and flexibility, IT professionals have gravitated towards open source. The term "open source" refers to software source code that is human-readable (as opposed to machine-readable) and thus open to use and change as desired by developers. Due to open source's unrestricted and highly malleable quality, Linux-based open source operating systems have become the platform of choice for many IT organizations, as well as a very active and innovative open source developer community.

About Linux

Many groups and companies provide and support one or more of the standard distributions of Linux, including Ubuntu, CentOS, RHEL, OpenSUSE, Debian, and Fedora. Most of these can be downloaded for free over the Internet. As a result of these efforts, Linux has become a stable, reliable, and cost-effective operating system platform and a viable alternative to more well known proprietary operating systems.

Open Source Business Model

Open source solutions are generally available in both source code and binary forms on-line and without charge. Communities of users and developers contribute the labor necessary to maintain them. Often, however, a principle entity (a person or a company) coordinates and manages each specific open source system (the Linux kernel being a notable exception). Companies that create open source software generally give-away their core product, and instead generate income by offering paid support services and customized versions that extend the baseline capabilities of the core. This business model keeps the software open to the value-adding creativity and innovations of the open source developer community.

Benefits of Transparency and Extensibility

The transparency and extensibility of Linux-based open source solutions make them highly adaptable and flexible to the needs of IT organizations, as follows:

Transparency

> Transparency refers to the ability to look into and fix elements of a system if necessary. Providing unrestricted human-readable code that can be modified by developers makes open source solutions highly transparent and thus highly adaptable and compatible with the broadest range of existing data center technologies.

Extensibility

> Extensibility refers to extending the capabilities of a system. Open source systems are highly extensible and can be freely built-upon by developers as desired. This extensibility promotes experimentation and innovation—developers can build customized add-on features that extend the usefulness and value of a program. Open source's extensibility thus lets IT organizations develop efficiency-enhancing customizations that better serve the organization and users.

B.7 Benefits of the Eucalyptus Cloud

The Eucalyptus private cloud gives IT organizations the features so essential to improving the efficiency of an IT infrastructure, including the following:

- **Data center optimization.** Eucalyptus optimizes existing data center resources with consolidation through virtualization of all data center elements, including machines, storage and network. Eucalyptus is compatible with most widely used virtualization technologies, including Xen, KVM, and ESX hypervisors.

- **Automated self-service**. Eucalyptus automates computer resource provisioning by allowing users to access their own flexible configurations of machines, storage, and networking devices as needed through a convenient self-service Web interface.

- **Customizable Web-interface.** Eucalyptus uses universally accepted Web-based network communication protocols that allow users to access computing resources through a highly customizable Web-interface.

- **Scalable data center infrastructure.** Eucalyptus clouds are highly scalable, which enables an organization to efficiently scale-up or scale-down data center resources according to the needs of the enterprise.

- **Elastic resource configuration.** The elasticity of a Eucalyptus cloud allows users to flexibly reconfigure computing resources as requirements change. This helps the enterprise workforce remain adaptable to sudden changes in business needs.

- **Open source core**. Highly transparent and extensible, Eucalyptus' open source core architecture remains entirely open and available for value-adding customizations and innovations provided by the open source development community. The Eucalyptus open source software core is available for free download at www.eucalyptus.com.

- **Hybrid cloud capability.** Engineered to emulate Amazon Web Services (AWS), Eucalyptus interacts seamlessly with Amazon public cloud services, including EC2 and S3, with no software modification required. This allows IT organizations to quickly "cloudburst" into the public cloud space without purchasing additional data center hardware during very large spikes in enterprise resource demand. The vibrant eco system built around the Amazon AWS can be leveraged. For example, RightScale, CohesiveFT, Zmanda, rPath are just a few of the partners that deliver solutions for Amazon AWS that in turn work seamlessly with Eucalyptus

Glossary

ami - Abbreviation for Amazon Machine Image. Used as a prefix in naming Amazon machine image files.

ATA (Advanced Technology Attachment) - Computer bus technology used primarily for transferring data to and from hard drives.

availability zones - An availability zone for Amazon denotes a large subset of their Cloud environment. Within Eucalyptus, we refine this definition to denote a subset of the cloud that shares a local area network. Each availability zone has its own cluster controller and storage controller. Properly credentialed administrators can view detailed information on availability zones with the following command:

```
euca-describe-availability-zones verbose
```

bridge names - To properly configure Eucalyptus networking modes you must know the correct bridge name for your system. Bridge names for systems using Xen or KVM hypervisors include the following: Xen 3.2: `eth0;` Xen 3.0 or earlier: `xenbr0;` kvm: `br0.`

Bucket - A bucket is a storage container that stores objects. Objects are added or removed from a bucket via the PUT and GET commands. Any number of objects can be added to a bucket up to the maximum storage capacity of the bucket.

Bundling - Bundling prepares a VM image to be uploaded into the cloud. (Before you can run an instance you must first bundle the image (`euca-bundle-image`); then upload the bundled image (`euca-upload-bundle`). "Bundling" an image separates the image into multiple image parts and generates an XML manifest file containing metadata about the image. When you run an instance of the image (`euca-run-instances manifest.xml`), Eucalyptus uses the manifest file to reference image parts and kernel in assembling the virtual machine.

Cloudburst - Cloudburst refers to the capability for private cloud users to extend beyond the enterprise firewall and access computing resources in the public cloud. Eucalyptus' hybrid cloud capability lets users seamlessly cloudburst into the AWS public cloud should additional resources be required.

cluster - In Eucalyptus, a cluster is a physical collection of a number of resources within single network segment. Each cluster contains a Cluster Controller, Storage Controller and a number of computer resources. An *availability zone* is a virtual abstraction of a cluster.

command-line tools (e.g., Euca2ools)**-** Text commands that let you control and manage VMs and other Eucalyptus cloud operations via a standard Terminal interface window.

credentials - Credentials refer to data values that are used to authenticate a user. Credentials can be provided in some cases by certificates and in other cases by keypairs.

Daemon - A daemon is a process that runs in the background to perform a specific function or system task. The CLC, CC, etc. are daemons that are part of the Eucalyptus Cloud environment.

ElasticFox - ElasticFox is an add-on for Firefox that lets you configure and manage both Amazon EC2 and Eucalyptus accounts through a graphical interface.

eki- -Abbreviation for Eucalyptus Kernel Image. Used as prefix in naming Eucalyptus kernel image files.

emi- -Abbreviation for Eucalyptus Machine Image. Used as prefix in naming Eucalyptus machine image files.

eri- - Abbreviation for Eucalyptus Ramdisk Image. Used as prefix in naming Eucalyptus ramdisk image files.

eucarc - The ‘eucarc’ file is a resource configuration file included with the credentials zip-file you download through the Eucalyptus Web interface. Before you can run Euca2ools against Eucalyptus you must first ‘source’ the eucarc file via a ssh-compatible shell (by entering `. eucarc` at the command line). By sourcing this file, a set of environment variables is defined to provide user-specific information to Euca2ools command line tools.

front end - The “front end” refers to the physical machine that hosts the Cloud Controller (CLC) and in most cases the Cluster Controller (CC) components. As such, the front end is the entry point into the cloud providing an interface through which underlying virtualized resources (servers, network, and storage) can be evaluated and managed. In this guide we use the terms “front end” and “front-end machine” interchangeably.

hypervisor - A hypervisor (e.g., Xen, KVM, ESX/ESXi) is a piece of software that enables multiple operating systems to run concurrently on a single host computer. Type 1 (aka native or bare-metal) hypervisors, including Xen, run directly on a host machine’s hardware. KVM (Kernel-based Virtual Machine) is a variation on a Type 1 hypervisor that embeds in a platform’s firmware. This way KVM transforms the Linux kernel into a hypervisor. Type 2 hypervisors are software applications that allow multiple operating systems to run on top of conventional operating systems.

instance – An instance refers to an actively running virtual machine. Instances are deployed by instantiating virtual machine images. Multiple identical VM instances can be instantiated from a single virtual machine image.

instantiate (aka 'running instances') – Instantiate refers to the process of deploying a running virtual machine instance from an encrypted virtual machine image. To instantiate a virtual machine image in Eucalyptus use the following command:

```
euca-run-instances –k <private key) –n <number> <emi-id>
```

kernel - The kernel is the central software component of a computer operating system. Kernels facilitate the functioning of application programs; and they bridge application processes with underlying data processing and management functions performed at the computer hardware level.

keypair - Keypairs are used in Eucalyptus to authenticate a user's identity. Before running a VM instance, you must first create a keypair as follows:

```
euca-add-keypair mykey >mykey.private
```

A pair of keys are created; one public key and one private key. The public key, "mykey" is stored in Eucalyptus. The private key "mykey.private" is stored within a file on a local machine. After creating a keypair, you must change access permissions to enable the private key in your local directory as follows:

```
chmod 0600 mykey.private
```

(Note: Use the file mykey.private when logging into VMs via the ssh command)

KVM (Kernel-based Virtual Machine) - KVM is a Linux kernel-based hypervisor that supports the instantiation of virtual machine instances and other virtualized computing resources.

libvirt - libvirt is a library, accessible via a C API. The library manages and supports the virtualization capabilities of Linux and other operating systems.

libvirt virsh - Command-line utility used for managing user domains via the libvirt library.

Linux - Linux refers to Unix-like operating systems based on the open source Linux kernel. Standard distributions of Linux include Ubuntu, CentOS, RHEL, OpenSUSE, Debian and Fedora. Most Linux distributions can be downloaded for free over the Internet.

manifest file - The manifest file (`manifest.xml`) is a file generated when bundling images prior to uploading them to the cloud. The manifest file contains metadata used by Eucalyptus to reference associated image parts and kernel when instantiating a virtual machine.

networking mode - Networking modes, including SYSTEM, STATIC, MANAGED, and MANAGED-NOVLAN, offer the administrator varying degrees of control over VM network access and security. Each mode contains a unique set of configuration parameters and features. System mode is the default mode. Managed mode offers the greatest number of network configurable features.

node - A node is a single physical machine on which any number of VM instances can be hosted and managed. In Eucalyptus, the Node Controller (NC) executes on each node designated for hosting VM instances. The NC manages all VM operations on a particular node via query and control requests from the Eucalyptus Cluster Controller.

open source - open source refers to software source code that is human-readable (as opposed to machine-readable) and thus open to use and change as desired by the general public. Due to open source's unrestricted and highly malleable quality, Linux-based open source operating systems have become the platform of choice for many IT organizations, as well as a very active and innovative open source developer community.

persistent data – persistent data refers to data that remains in an unchanged state regardless of the change in state of the software or processes that created that data.

PEM (Privacy Enhanced Mail) – PEM is an ASCII Base64 encryption format frequently used for creating standard X.509 digital certificates and public keys.

ramdisk – ramdisk contains a set of drivers loaded temporarily into memory that instructs the kernel to launch appropriate system files when instantiating a virtual machine.

reservation – In Eucalyptus, a reservation refers to an ID tag that is created each time one or more VM instances are instantiated using the `euca-run-instances` command. You can view reservation ID tags associated with VM instances by querying the system with the `euca-describe-instances` command.

REST – REST (Representational State Transfer) is an HTTP-based query method used by Web services that uses encoded representational messages to conduct client-server (request and response) interactions. Rest/Query APIs use common directory-like URLs containing these encoded messages to prescribe specific service requests and subsequent responses returned in any number of formats, including HTML, XML, media files, etc.

root directory– In Unix-based operating systems that use a hierarchical file system, root is the directory containing all other directories within a file system. The root directory in Unix-based systems is represented with a forward slash (/).

RPM – RPM is a software package file format used by Linux distributions. Initially developed by Red Hat for Red Hat Linux (RHEL), RPM is now used by many Linux-based open source distributions, including Centos, OpenSUSE, Fedora, and others.

security group – A security group is a set of networking rules (in effect a firewall) applied to VMs that are associated with a group.

SLA (Service Level Agreement) – An SLA is an agreement that formally defines a measurable degree of service a customer can expect from a service provider. SLAs are frequently used to specify minimum and/or target levels of availability, performance, operation, and other service-related attributes, including accounting and billing.

SOAP (Simple Object Access Protocol) – SOAP is a structured messaging protocol used by Web services to exchange data over computer networks. SOAP messages generally rely on XML and HTTP protocols for message specification, transmission, and negotiation.

source code – 'source code' refers to the original 'human-readable' computer program written by a programmer in any programming language. To execute a program its source code must be compiled and translated into 'machine-readable' code understood by computers.

sourcing a file (`. eucarc`) - When a file is 'sourced' (by typing either **`source filename`** or **`. filename`** at the command line), the lines of code in the file are executed as if they were entered at the command line. Sourcing the 'eucarc' file (Eucalyptus resource configuration file) establishes necessary environmental variables and validates required user authentications for your Eucalyptus cloud.

SSH (TCP, port 22)– The Unix/Linux shell is used to make a secure connection between two network devices. By default, SSH utilizes TCP port 22.

snapshot – A snapshot is a copy of the total existing state of a volume. Eucalyptus lets you create instantaneous snapshots of a volume (`euca-create-snapshot <vol-xxxxxxxx>`). You can then create a new volume (with identical file system and/or data) from that snapshot (`euca-create-volume --snapshot <snapshot id> -z <zone>`).

YUM (Yellowdog Updater, Modified) – YUM is a metapackage management utility that computes dependencies and implements processes that install and maintain RPM based Linux distributions.

Ubuntu Enterprise Cloud – The Ubuntu Enterprise Cloud (UEC) is an open source cloud computing platform included with the Ubuntu Linux distributions. The Eucalyptus open source software core is the key component of UEC.

virtual machine image – A virtual machine image provides the root file system for a virtual machine instance. Machine images can be created from a variety of Linux operating systems, including Ubuntu, Centos, RHEL, OpenSUSE, Debian, and Fedora. You can create your own images or use one of the pre-made images available from the Eucalyptus Web interface or a variety of other sources.

virtual network – A virtual network is an abstraction of a computer network that combines hardware and software network resources and network functionality into a single, software-based administrative entity.

virtualization – Virtualization is a general term that refers to the abstraction of computer resources in software (or a combination of software and hardware).

virtual machine (VM) – A virtual machine is an abstraction of computer hardware within software. As such, virtual machines execute programs as if they were actual physical machines.

Web service – A Web service is a software system that supports automated machine-to-machine interaction over a network. Web services use interfaces described in Web Services Description Language (WSDL). Other systems interact with Web services using SOAP or REST messages via HTTP with XML and other web-related standards.