EUCALYPTUS

3.2.0 User Guide

2012-12-16 Eucalyptus Systems

Contents

Welcome

Eucalyptus is a Linux-based software architecture that creates scalable private and hybrid clouds within your existing IT infrastructure. It allows you to provision your own collections of resources on an as-needed basis.

Eucalyptus provides 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.

Eucalyptus Features

Eucalyptus offers ways to implement, manage, and maintain your own collection 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 your identity when you log into VMs using SSH. You can add, describe, and delete keypairs.
Image Management	Before running instances, someone must prepare VM images for use in the cloud. This can be an administrator or a user. You can bundle, upload, register, describe, download, unbundle, and deregister VM images.
Linux-based VM Management	Eucalyptus lets you run their own VM instances in the cloud. You can run, describe, terminate, and reboot a wide variety of Linux-based VM instances that were prepared using image management commands.
Windows-based VM Management	You can run, describe, terminate, reboot, and bundle instances of Windows VMs.
IP Address Management	Depending on the networking mode, you might have access to elastic IPs. Elastic IPs are public IP addresses that users can reserve and dynamically associate with VM instances. You 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. You can create, describe, delete, authorize, and revoke security groups.
Volume and Snapshot Management	Eucalyptus allows you to create dynamic block volumes. A dynamic block volume is similar to a raw block storage device that can be used with VM instances. You can create, attach, detach, describe, bundle, and delete volumes. You can also create and delete snapshots of volumes and create new volumes from snapshots.

Who Should Read this Guide?

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

What’s in this Guide?

This guide contains instructions for users of the Eucalyptus 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.

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.

In order to use Eucalyptus, you need to:

1. *Sign Up for an Account*
2. *Get User Credentials*
3. *Set Up Euca2ools*

After you have the credentials and a way to access Eucalyptus, you can start using your cloud.

Sign Up for an Account

Before you can access your Eucalyptus cloud, you must first sign up for an account using the Eucalyptus Administrator Console.

Important: 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.

To sign up for an account:

1. Open your browser to the Eucalyptus Administrator Console at https://<your_CLC_IP_address>:8443/.
   Ask your system administrator for the URL if you don't know it.
2. Click **Apply** to access the application form.
3. Fill out the Eucalyptus account application form.
   An approval email is sent to your mailbox.
4. Click the URL in the confirmation email that you received from the cloud administrator.

This creates an account with an admin user. You can now log in to the Eucalyptus Administrator Console as the admin user, using the username and password that you chose when filling out the application form.

Get User Credentials

You 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 log in to the Eucalyptus Administrator Console as the admin user and obtain these credentials.

To get user credentials

1. Log in to the Eucalyptus Administrator Console using your username and password.
2. Click your username at the top of the screen and then click **Download new credentials**.
   The download contains a zip-file with your public/private key pair, a bash script, and several other required files.
3. 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:

```
mkdir ~/.euca
cd ~/.euca
unzip <filepath>/euca2-<user>-x509.zip
```

```
chmod 0700 ~/.euca
chmod 0600 *
```

You're now ready to use the command-line interface (CLI), called "Euca2ools."

Set Up Euca2ools

Eucalyptus provides a command-line interface (CLI), called Euca2ools. The remainder of this guide uses Euca2ools. These tools conform to the command-line tools that Amazon distributes as part of their EC2 service (EC2 tools).

Tip: 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 `euca-bundle-image --help` or `man euca-bundle-image`.

Overview of Euca2ools

Euca2ools is the Eucalyptus command line interface for interacting with web services. This set of tools was written in Python, relying on the Boto library and M2Crypto toolkit.

Most Euca2ools commands are compatible with Amazon’s EC2, S3, and IAM services and generally accept the same options and honor the same environmental variables. This means that you can use Euca2ools with both the Eucalyptus cloud-computing platform and Amazon EC2, S3, and IAM. A few other commands are specific to Eucalyptus and are noted in the command description.

Note: Some operations in Euca2ools are not supported in certain Eucalyptus networking modes. For example, security groups and elastic IPs are not supported in both Static and System networking modes. See your cloud administrator for details. For more information about networking modes, see the *Installation Guide*.

Installing Euca2ools

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

Setting Environment Variables

Euca2ools uses two kinds of credentials to authenticate user identity: X.509 PEM-encoded certificates and access keys. In addition, you must also specify service endpoints.

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.

You can specify each value individually using the command line or set all of them collectively by sourcing the eucarc file. For more information, see *Source a Eucarc File*.

Variable	Option	Description
EC2_URL	`-U, --url <url>`	http://<host_ip>:8773/services/Eucalyptus
S3_URL	`-U, --url <url>`	http://<host_ip>:8773/services/Walrus
EUCALYPTUS_CERT	`--ec2cert_path <file>`	Path to cloud certificate
EC2_CERT	`-c, --cert <file>`	Path to your PEM-encoded certificate
EC2_PRIVATE_KEY	`-k, -privatekey <file>`	Path to your PEM-encoded private key
EC2_ACCESS_KEY	`-a, --access-key <key>`	Your access key ID
EC2_SECRET_KEY	`-s, --secret-key <key>`	Your secret access key

Source a Eucarc File

`eucarc` is a resource configuration file that defines the correct values for each associated variable. `eucarc` is in your credentials zip file. Sourcing eucarc sets these variables to the correct values.

Note: When you source eucarc, you are only setting the correct values for the session. If you open a new terminal you must source eucarc again.

To source the eucarc file, enter the following:

Open a terminal and enter the following:

```
source eucarc
```

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

Using Images

An image is a snapshot of a system's root file system and it provides the basis for instances. When you run a new virtual machine, you choose a machine image to use as a template. The new virtual machine is then an instance of that machine image that contains its own copy of everything in the image. The instance keeps running until you stop or terminate it, or until it fails. If an instance fails, you can launch a new one from the same image. You can create multiple instances of a single machine image. Each instance will be independent of the others.

You can use a single image or multiple images, depending on your needs. From a single image, you can launch different types of instances. An *instance type* defines what hardware the instance has, including the amount of memory, disk space, and CPU power.

A machine image contains all the information needed to boot instances of your software. For example, a machine image might contain software to act as an application server or Hadoop node.

Note: This section describes instances that are not backed by dynamic block volumes, or Eucalyptus block storage (EBS) devices. For information about EBS, see *Using EBS*.

Available Images

Existing machine images available at the *Eucalyptus Machine Images* page.

You might find that an existing machine image meets your needs. However, you might want to customize a machine image or create a machine image for your own use. For more information about creating your own machine image, see *Creating an Image*.

Image Types

Machine images are either backed by persistent storage or backed by instance store. An image backed by persistent storage means that the root device is a snapshot and appears as a persistent volume when an instance is launched from the machine image. A machine that is backed by instance store means that the root device is stored in Walrus and appears as instance store when an instance is launched from the machine image

.

Image Overview

An image defines what will run on a guest instance with your Eucalyptus cloud. Typically, an image contains one of the Linux distributions like CentOS, Fedora, Ubuntu, Debian or others. It could also contain one of the supported Windows server versions. The format for these is identical.

Normally when we use the term "image" we mean the root file system. Once bundled, uploaded, and registered with Eucalyptus, such an image is known as a Eucalyptus machine image (EMI).

There are, however, other types of images that support the EMI. They are the kernel (EKI) and ramdisk (ERI). They contain kernel modules necessary for proper functioning of the image. Often, one set of these ERIs and EKIs are used by multiple EMIs. Once loaded into the Eucalyptus cloud, the EKI and ERI are referred to by the image and you don't have much interaction with them directly.

Tip: When you run an image, you can override the image's associated kernel and ramdisk if necessary (for example, if you want to try out another kernel. For more information, see *Associate a Kernel and Ramdisk*.

To help get you started, Eucalyptus provides pre-packaged virtual machines that are ready to run in your cloud. You can get them at the *Eucalyptus Machine Images* page. Each Eucalyptus image from this site comes bundled with a correspoding EKI and ERI. You can manually download these images from the web page, or you can use the Eualyptus Image Store commands to list and describe these images, as well as to install an image in your cloud. For more information see the Eucalyptus Image Store section in the *Command Line Reference Guide*.

If you find that the pre-packaged images don't meet your needs, you can add an image from an existing, non-registered image, or create your own image.

Once you've selected and downloaded the image(s) you plan to use, read the details in the following section for directions about how to bundle, upload and register the images with your Eucalyptus cloud.

To view the status of your newly created and registered EKI, ERI, and EMI images, use the `euca-describe-images` command. You can also view image status using the Eucalyptus Administrator Console's **Images** page.

Once you have added your image to Eucalyptus, see the *User Guide* for information about launching and using instances based on the image.

Image Tasks

This section explains the tasks that you can perform on images in Eucalyptus.

You can perform the following image-related tasks listed in the following sections:

- *Add an Image*
- *Browse and Install Images from EuStore*
- *Associate a Kernel and Ramdisk*
- *Modify an Image*
- *Creating an Image*
- *Bundle an Image for Amazon EC2*
- *Bundle a Windows Instance*

Add an Image

An image is the basis for instances that you spin up for your computing needs. If you have access to images that meet your needs, you can skip this section. This section explains how to add an image to your Eucalyptus cloud that is customized for your needs.

There are a few ways you can add an image to Eucalyptus:

- **Add an image based on an existing image.** If you have an image already, read the rest of this section. Eucalyptus has stock images available to help you get started right away. You can find links to these images in the Eucalyptus Administrator Console's start page. You can also get images from the *Eucalyptus Machine Images* page.
- Add an image that you modify from an existing image. If you have an image that you want to modify, see *Modify an Image*.
- Add an image that you create. If you want to create a new image, see *Creating an Image*.

Each of these three ways has a different first step, but the way you get image to Eucalyptus is the same. To enable an image as an executable entity, you do the following:

1. Bundle a root disk image and kernel/ramdisk pair
2. Upload the bundled image to Walrus bucket storage
3. Register the data with Eucalyptus

Important: 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.

Once you have an image that meets your needs, perform the tasks listed in this section to add the image to your cloud.

Add a Kernel

When you add a kernel to Walrus, you bundle the kernel file, upload the file to a bucket in Walrus that you name, and then register the kernel with Eucalyptus.

To add a kernel to Walrus:

Use the following three commands:

```
euca-bundle-image -i <kernel_file> --kernel true
euca-upload-bundle -b <kernel_bucket> -m /tmp/<kernel_file>.manifest.xml
euca-register <kernel_bucket>/<kernel_file>.manifest.xml
```

For example:

```
euca-bundle-image -i
euca-fedora-10-x86_64/xen-kernel/vmlinuz-2.6.27.21-0.1-xen --kernel
 true
...
Generating manifest /tmp/vmlinuz-2.6.27.21-0.1-xen.manifest.xml

euca-upload-bundle -b example_kernel_bucket -m
/tmp/vmlinuz-2.6.27.21-0.1-xen.manifest.xml
...
Uploaded image as
example_kernel_bucket/vmlinuz-2.6.27.21-0.1-xen.manifest.xml

euca-register
example_kernel_bucket/vmlinuz-2.6.27.21-0.1-xen.manifest.xml
IMAGE eki-XXXXXXXX
```

Where the returned value `eki-XXXXXXXX` is the unique ID of the registered kernel image.

Add a Ramdisk

When you add a ramdisk to Walrus, you bundle the ramdisk file, upload the file to a bucket in Walrus that you name, and then register the ramdisk with Eucalyptus.

To add a ramdisk to Walrus:

Use the following three commands:

```
euca-bundle-image -i <ramdisk_file> --ramdisk true
euca-upload-bundle -b <ramdisk_bucket> -m /tmp/<ramdisk_file>.manifest.xml
euca-register <ramdisk_bucket>/<ramdisk_file>.manifest.xml
```

For example:

```
euca-bundle-image -i
euca-fedora-10-x86_64/xen-kernel/initrd-2.6.27.21-0.1-xen --ramdisk
 true
...
Generating manifest /tmp/initrd-2.6.27.21-0.1-xen.manifest.xml

euca-upload-bundle -b example_rd_bucket -m
/tmp/initrd-2.6.27.21-0.1-xen.manifest.xml
...
Uploaded image as
example_rd_bucket/initrd-2.6.27.21-0.1-xen.manifest.xm

euca-register
example_rd_bucket/initrd-2.6.27.21-0.1-xen.manifest.xml
```

```
IMAGE eri-XXXXXXXX
```

Where the returned value `eri-XXXXXXXX` is the unique ID of the registered ramdisk image.

Add a Root Filesystem

When you add a root filesystem to Walrus, you bundle the root filesystem file, upload the file to a bucket in Walrus that you name, and then register the root filesystem with Eucalyptus. The bundle operation can include a registered ramdisk (ERI ID) and a registered kernel (EKI ID). The resulting image will associate the three images.

You can also bundle the root file system independently and associate the ramdisk and kernel with the resulting EMI at run time. For more information, see *Associate a Kernel and Ramdisk*.

To add a root filesystem to Walrus:

Use the following three commands:

```
euca-bundle-image -i <root_filesystem_file>
euca-upload-bundle -b <root_filesystem_file_bucket> -m
/tmp/<root_filesystem_file>.manifest.xml
euca-register <root_filesystem_file_bucket>/<root_filesystem_file>.manifest.xml
```

For example:

```
euca-bundle-image -i euca-fedora-10-x86_64/fedora.10.x86-64.img
--ramdisk eri-722B3CBA --kernel eki-5B3D3859
...
Generating manifest /tmp/fedora.10.x86-64.img.manifest.xml

euca-upload-bundle -b example_rf_bucket -m
/tmp/fedora.10.x86-64.img.manifest.xml
...
Generating manifest /tmp/fedora.10.x86-64.img.manifest.xml

euca-register example_rf_bucket/fedora.10.x86-64.img.manifest.xml
IMAGE  emi-XXXXXXXX
```

Where the returned value `emi-XXXXXXXX` is the unique ID of the registered machine image.

Browse and Install Images from EuStore

Eucalyptus provides a resource - called EuStore - that contains images that you can download and install. This task explains how to browse and install images from EuStore.

To browse and install an image from EuStore:

1. Find an image on EuStore:

```
eustore-describe-images
```

This command returns a list of images available from the EuStore. For example:

```
0400376721 fedora      x86_64  starter        kvm               Fedora 16
x86_64 - SELinux / iptables disabled. Root disk of 4.5G. Root user enabled.
2425352071 fedora      x86_64  starter        kvm               Fedora 17
x86_64 - SELinux / iptables disabled. Root disk of 4.5G. Root user enabled.
```

```
1107385945 centos     x86_64  starter       xen, kvm, vmware  CentOS 5 1.3GB
 root, Hypervisor-Specific Kernels
3868652036 centos      x86_64  starter        kvm               CentOS 6.3
x86_64 - SELinux / iptables disabled. Root disk of 4.5G. Root user enabled.
1347115203 opensuse    x86_64  starter        kvm               OpenSUSE 12.2
 x86_64 - KVM image. SUSE Firewall off. Root disk of 2.5G. Root user enabled.
 Working with kexec kernel and ramdisk. OpenSUSE minimal base package set..
.
.
.
```

For additional information regarding the images on eustore (for example, who is the maintainer of the image), use the -v option:

```
# eustore-describe-images -v
0400376721 fedora      x86_64  starter        kvm               Fedora 16
x86_64 - SELinux / iptables disabled. Root disk of 4.5G. Root user enabled.
     20121107181713      d13e-1e35   fedora-based
olivier.renault@eucalyptus.com
2425352071 fedora      x86_64  starter        kvm               Fedora 17
x86_64 - SELinux / iptables disabled. Root disk of 4.5G. Root user enabled.
     20121107181713      6369-6e28   fedora-based
olivier.renault@eucalyptus.com
1107385945 centos     x86_64  starter       xen, kvm, vmware  CentOS 5 1.3GB
 root, Hypervisor-Specific Kernels
     20120517102326      84ae-59db   centos-based
images@lists.eucalyptus.com
3868652036 centos      x86_64  starter        kvm               CentOS 6.3
x86_64 - SELinux / iptables disabled. Root disk of 4.5G. Root user enabled.
     20121107181713      48df-52d4   centos-based
olivier.renault@eucalyptus.com
1347115203 opensuse    x86_64  starter        kvm               OpenSUSE 12.2
 x86_64 - KVM image. SUSE Firewall off. Root disk of 2.5G. Root user enabled.
 Working with kexec kernel and ramdisk. OpenSUSE minimal base package set..
     20121120130646      a981-db13   opensuse-based
lester.wade@eucalyptus.com
```

2. Pick an available image from the returned list and note the image ID. For this example, we will choose:

```
3868652036 centos      x86_64  starter        kvm               CentOS 6.3
x86_64 - SELinux / iptables disabled. Root disk of 4.5G. Root user enabled.
```

3. Install the image from EuStore using the eustore-install-image command. For this example, we only need to specify the image ID and the name of a bucket (the bucket will be created if it doesn't already exist):

   Note: Some images may require additional parameters (for example, that you specify a kernel type with the -k option). Please see the eustore-install-image topic in the Eucalyptus Command Line Reference for more information.

```

eustore-install-image -b centos-testbucket -i 3868652036 -k kvm

```

This command performs a number of tasks for you, including downloading the image from the central Eucalyptus image store and installing the image on your own Eucalyptus private cloud. The output from this command will look similar to the following example:

```
Downloading Image :  CentOS 6.3 x86_64 - SELinux / iptables disabled. Root
disk of 4.5G. Root user enabled.
0-----1-----2-----3-----4-----5-----6-----7-----8-----9-----10
##############################################################

Checking image bundle
Unbundling image
going to look for kernel dir : kvm-kernel
Bundling/uploading ramdisk
Checking image
Compressing image
Encrypting image
Splitting image...
Part: initrd-2.6.32-279.14.1.el6.x86_64.img.part.00
Generating manifest
/tmp/olEuG_/initrd-2.6.32-279.14.1.el6.x86_64.img.manifest.xml
Checking bucket: centos-testbucket
Creating bucket: centos-testbucket
Uploading manifest file
Uploading part: initrd-2.6.32-279.14.1.el6.x86_64.img.part.00
Uploaded image as
centos-testbucket/initrd-2.6.32-279.14.1.el6.x86_64.img.manifest.xml
centos-testbucket/initrd-2.6.32-279.14.1.el6.x86_64.img.manifest.xml
eri-064B387A
Bundling/uploading kernel
Checking image
Compressing image
Encrypting image
Splitting image...
Part: vmlinuz-2.6.32-279.14.1.el6.x86_64.part.00
Generating manifest /tmp/olEuG_/vmlinuz-2.6.32-279.14.1.el6.x86_64.manifest.xml
Checking bucket: centos-testbucket
Uploading manifest file
Uploading part: vmlinuz-2.6.32-279.14.1.el6.x86_64.part.00
Uploaded image as
centos-testbucket/vmlinuz-2.6.32-279.14.1.el6.x86_64.manifest.xml
centos-testbucket/vmlinuz-2.6.32-279.14.1.el6.x86_64.manifest.xml
eki-A4D6398A
Bundling/uploading image
Checking image
Compressing image
Encrypting image
Splitting image...
Part: centos-6.3-x86_64.part.00
Part: centos-6.3-x86_64.part.01
Part: centos-6.3-x86_64.part.02
Part: centos-6.3-x86_64.part.03
Part: centos-6.3-x86_64.part.04
Part: centos-6.3-x86_64.part.05
Part: centos-6.3-x86_64.part.06
Part: centos-6.3-x86_64.part.07
Part: centos-6.3-x86_64.part.08
Part: centos-6.3-x86_64.part.09
Part: centos-6.3-x86_64.part.10
Part: centos-6.3-x86_64.part.11
Part: centos-6.3-x86_64.part.12
Part: centos-6.3-x86_64.part.13
Part: centos-6.3-x86_64.part.14
```

```
Part: centos-6.3-x86_64.part.15
Part: centos-6.3-x86_64.part.16
Part: centos-6.3-x86_64.part.17
Part: centos-6.3-x86_64.part.18
Part: centos-6.3-x86_64.part.19
Generating manifest /tmp/olEuG_/centos-6.3-x86_64.manifest.xml
Checking bucket: centos-testbucket
Uploading manifest file
Uploading part: centos-6.3-x86_64.part.00
Uploading part: centos-6.3-x86_64.part.01
Uploading part: centos-6.3-x86_64.part.02
Uploading part: centos-6.3-x86_64.part.03
Uploading part: centos-6.3-x86_64.part.04
Uploading part: centos-6.3-x86_64.part.05
Uploading part: centos-6.3-x86_64.part.06
Uploading part: centos-6.3-x86_64.part.07
Uploading part: centos-6.3-x86_64.part.08
Uploading part: centos-6.3-x86_64.part.09
Uploading part: centos-6.3-x86_64.part.10
Uploading part: centos-6.3-x86_64.part.11
Uploading part: centos-6.3-x86_64.part.12
Uploading part: centos-6.3-x86_64.part.13
Uploading part: centos-6.3-x86_64.part.14
Uploading part: centos-6.3-x86_64.part.15
Uploading part: centos-6.3-x86_64.part.16
Uploading part: centos-6.3-x86_64.part.17
Uploading part: centos-6.3-x86_64.part.18
Uploading part: centos-6.3-x86_64.part.19
Uploaded image as centos-testbucket/centos-6.3-x86_64.manifest.xml
centos-testbucket/centos-6.3-x86_64.manifest.xml
Installed image: emi-233637E1
```

Note the last line in the output, which provides the image ID for the image you just installed from the euca store. In this example, the image ID is emi-233637E1.

4. Verify the image was installed on your Eucalyptus cloud. To do this, use the euca-describe-images command, which returns a list of the available images on your Eucalyptus cloud:

```
euca-describe-images | grep centos-testbucket
```

This command will return output similar to the following example:

```
IMAGE   eki-A4D6398A
centos-testbucket/vmlinuz-2.6.32-279.14.1.el6.x86_64.manifest.xml
345590850920    available   public      x86_64  kernel
instance-store
IMAGE   eri-064B387A
centos-testbucket/initrd-2.6.32-279.14.1.el6.x86_64.img.manifest.xml
345590850920    available   public      x86_64  ramdisk
instance-store
IMAGE   emi-233637E1    centos-testbucket/centos-6.3-x86_64.manifest.xml
345590850920    available   public      x86_64  machine eki-A4D6398A
eri-064B387A        instance-store
```

Note the ID of the last image in the output -

```
emi-233637E1
```

- matches that of the image we installed from EuStore.

The image has been successfully downloaded from EuStore and installed on your Eucalyptus cloud.

You can now run an instance from this image and connect to it using SSH.

Associate a Kernel and Ramdisk

There are three ways that you can associate a kernel and ramdisk with an image.

Tip: This only applies to Linux images. Windows images do not use kernels or ramdisks.

- You can associate a specific kernel and ramdisk identifier with an image at the `euca-bundle-image` step.

```
euca-bundle-image -i <vm_image_file> --kernel <eki-XXXXXXXX> --ramdisk
<eri-XXXXXXXX>
```

- You can specific kernel and ramdisk at instance run time as an option to `euca-run-instances` command.

```
euca-run-instances --kernel <eki-XXXXXXXX> --ramdisk <eri-XXXXXXXX>
<emi-XXXXXXXX>
```

- An administrator can set default registered kernel and ramdisk identifiers that will be used if a kernel and ramdisk are unspecified by either of the other options.

Modify an Image

You might find that existing images in your cloud don't meet your needs.

To modify an existing image:

1. Create a mount point for your image.

```
mkdir temp-mnt
```

2. Associate a loop block device to the image.

```
losetup /dev/loop5 <image_name>
```

where `loop5` is a free device

3. Mount the image.

```
mount /dev/loop5 temp-mnt
```

4. Make procfs, dev and sysfs available in your chroot environment.

```
mkdir -p temp-mnt/proc
mkdir -p temp-mnt/sys
mkdir -p temp-mnt/dev
mount -o bind /proc temp-mnt/proc
mount -o bind /sys temp-mnt/sys
mount -o bind /dev temp-mnt/dev
```

5. You now have the image under `temp-mnt` and you can copy over what you want into it. If you want to install packages into it, you have few options:
    - `chroot temp-mnt` and use `apt-get`, `yum`, or `zypper` to install what you want.
    - Instruct the package manager program to use a different root (for example both dpkg and rpm uses the --root option)
6. Unmount the drive.

```
umount /dev/loop5
losetup -d /dev/loop5
```

You now have an image with your modifications. You are ready to *add the image* to Eucalyptus.

Creating an Image

You can create your own image if you to deploy an instance that is customized for your specific use. For example, you can create an EMI to deploy a LAMP server, MySQL database server, or a Postgres database server. All you need is some familiarity with the creation process.

About Linux Images

A Linux image is a root partition of a Linux installation. We use the convention followed by EC2 images:

- The first partition is the root partition (where the EMI is attached)
- The second is the ephemeral partition (for additional storage)
- The third is the swap partition

Any image you create must conform to these conditions (for example, `/etc/fstab` must follow this convention). Typically, the disk provided by Eucalyptus (as well as EC2) is the first SCSI disk (sda). However, you can customize it to be, for example, the first IDE disk (hda) or the first virtual disk on a paravirtual machine (xvda).

About Windows Images

Eucalyptus supports a licensed version of Windows images. Before you bundle a Windows image, you must have a valid version of Windows OS installed on your hypervisor and the Eucalyptus Windows Integration Service installed in the created Windows VM. The following sections detail how to perform these tasks.

Eucalyptus is compatible with images created from licensed versions of Windows Server 2003 R2, Windows Server 2008, Windows Server 2008 R2, and Windows 7. Windows OS is sensitive to physical and virtual hardware changes made after installation. We recommend that you create any Windows image on the specific hypervisor (Xen, KVM, or VMware) where you plan to run instances of that Windows image. Eucalyptus does not support running a Windows image across different hypervisors.

Before you begin, you will need the following room for a blank disk file:

- 8GB minimum, for Windows Server 2003 R2 (32-bit and 64-bit)
- 15GB minimum for Windows Server 2008 (32-bit and 64-bit), Windows Server 2008 R2 (64-bit), and Windows 7 (32-bit and 64-bit)

Tip: Eucalyptus implements ephemeral disks by which a running Windows instances are allocated an additional disk space based on the instance type. For example, if you assign an instance type 15 GB of disk and the registered Windows EMI is 8 GB, 7 GB of disk spaces are ephemeral disks accessible under `D:\`. If your Windows application uses scratch space most of time, we recommend keeping the Windows EMI small because it takes longer to launch bigger EMIs.

The windows image is a root file system that has no kernel and ramdisk associated with it. After creating your own Windows images, you can bundle, upload, and register it with Eucalyptus.

Hypervisor and OS Information

The following sections detail how to create an image based on the hypervisor and image OS you wish to use. To create an image using Windows, see *Create Windows Image*. To create an image using Linux with either KVM or Xen, see *Create Linux Image (Xen/KVM)*. To create an image using Linux with VMware, see *Create Linux Image (VMware)*.

Create Windows Image

For KVM and Xen hypervisors, we recommend that you perform this task on a node controller (NC), or a host running the same Linux distributions and hypervisors as your NCs. If you are creating the Windows image on a machine currently running as a NC, terminate all running instances and stop the NC. To stop the NC, enter:

```
service eucalyptus-nc stop
```

Template files that closely match those that Eucalyptus generates at VM instantiation time exist for KVM and Xen. These files are located at `/usr/share/eucalyptus/doc/libvirt-[hypervisor]-windows-example.xml`. We recommend that you review the appropriate file to acquaint yourself with its contents, noting required files, bridges, and resources. For more information about configuring the libvirt.xml file, go to the *Domain XML format* page in the libvirt documentation.

To create an image from a Windows OS in VMware you will need one network interface and one disk.

Note: If you are using VMware, make sure that the Windows VM uses the LSI Logic Parallel driver as the SCSI controller. For some Windows versions, this is not the default SCSI controller in the VM setting.

Install Base Windows OS

The first task for creating a Windows image is installing a base Windows operating system (OS). To install a base Windows OS using KVM or Xen:

1. Log in to the stopped NC server or a host that runs the same hypervisor as the NCs.
2. Create a blank disk file. Specify your Windows VM image name using the parameter `of`.

   ```
   dd if=/dev/zero of=windows.<image_name>.img bs=1M count=1 seek=16999
   ```

   Important: Your image name must start with the word, `windows` (all lower-case).

3. Create a floppy and secondary blank disk to be attached to the image later, in order to test paravirtualization drivers

   ```
   dd if=/dev/zero of=floppy.img \
    bs=1k count=1474
    dd if=/dev/zero of=secondary.img \
    bs=1M count=1 seek=1000
   ```

4. Copy all of the .img and .iso files to the `/var/lib/libvirt/images/` directory.
5. Copy the `libvirt-[hypervisor]-windows-example.xml` file to your working directory and rename it to `libvirt-[hypervisor]-windows.xml`, where `[hypervisor]` is one of `xen` or `kvm`.

   ```
   cp libvirt-kvm-windows-example.xml
   /var/lib/libvirt/images/libvirt-kvm-windows.xml
   ```

   or

   ```
   cp libvirt-xen-windows-example.xml
   /var/lib/libvirt/images/libvirt-xen-windows.xml
   ```

6. Open the new `libvirt-[hypervisor]-windows.xml` file and provide fully qualified paths to the VM image file and iso. Make sure that the name of the bridge is the same as the one used by the hypervisor on which you are creating the Windows image.

   Your file should look similar to one of the following examples.

   For Xen:

```
<domain type='xen'>
    <name>eucalyptus-windows</name>
    <os>
            <type>hvm</type>
            <loader>/usr/lib/xen/boot/hvmloader</loader>
            <boot dev='cdrom'/>
    </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='localtime'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
    <memory>524288</memory>
    <vcpu>1</vcpu>
    <devices>
        <emulator>/usr/lib64/xen/bin/qemu-dm</emulator>
        <disk type='file'>
            <source file='/root/windows_2003.img'/>
            <target dev='hda' bus='ide'/>
        </disk>
        <!--<disk type='file' device='disk'>
            <driver name='tap' type='aio'/>
            <source file='fully_qualified_path_to_secondary_disk'/>
            <target dev='xvda' bus='xen'/>
        </disk>
        <disk type='file' device='floppy'>
             <source file='fully_qualified_path_to_floppy_disk'/>
             <target dev='fda'/>
        </disk> -->
        <disk type='file' device='cdrom'>
            <source
file='/root/en_win_srv_2003_r2_enterprise_with_sp2_cd1_x13-05460.iso'/>
            <target dev='hdc'/>
            <readonly/>
        </disk>
        <interface type='bridge'>
            <source bridge='xenbr0'/>
            <script path='/etc/xen/scripts/vif-bridge'/>
        </interface>
        <serial type='pty'>
           <source path='/dev/pts/3'/>
           <target port='0'/>
        </serial>
        <input type='tablet' bus='usb'/>
        <input type='mouse' bus='ps2'/>
        <graphics type='vnc' port='-1' autoport='yes' keymap='en-us'
listen='0.0.0.0'/>
    </devices>
</domain>
```

For KVM:

```
<domain type='kvm'>
   <name>eucalyptus-windows</name>
    <os>
    <type>hvm</type>
    <boot dev='cdrom'/>
    </os>
    <features>
        <acpi/>
    </features>
    <memory>524288</memory>
    <vcpu>1</vcpu>
    <devices>
        <emulator>/usr/libexec/qemu-kvm</emulator>
        <disk type='file'>
            <source file='/var/lib/libvirt/images/windows_2003.img'/>
            <target dev='hda'/>
        </disk>
        <!-- <disk type='file' device='disk'>
             <source file='fully_qualified_path_to_secondary_disk'/>
             <target dev='vda' bus='virtio'/>
        </disk>
        <disk type='file' device='floppy'>
             <source file='fully_qualified_path_to_floppy_disk'/>
             <target dev='fda'/>
        </disk> -->
        <disk type='file' device='cdrom'>
            <source
file='/var/lib/libvirt/images/en_win_srv_2003_r2_enterprise_with_sp2_cd1_x13-05460.iso'/>

            <target dev='hdc'/>
            <readonly/>
        </disk>
        <interface type='bridge'>
            <source bridge='br0'/>
            <model type='rtl8139'/>
        </interface>
        <!--<interface type='bridge'>
            <source bridge='br0'/>
            <model type='virtio'/>
        </interface> -->
        <graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0'/>
    </devices>
</domain>
```

7. Start the VM.

```
cd /usr/share/eucalyptus/doc/
virsh create libvirt-[hypervisor]-windows.xml
```

8. Connect to the virtual console using the VNC client of your choice. On the NC, check the display number that has been allocated by looking at the process table (`ps axw | grep vnc`). For example, if the display number is 0, then connect to the NC using the VNC client:

```
vinagre <machine-hosting-vm>:0
```

9. Follow the standard Windows installation procedure until the VM has completed installing Windows.

   Tip: On some hosts, the VNC's display number will change when an image restarts. Use `ps` to find the current number.

10. Run `virsh list` to display the domain name.
11. Shut down the Windows VM you have just created. The easiest way to shutdown your VM is to use the `virsh destroy` command, as shown:

```
virsh destroy <domain_name>
```

To install the base Windows operating system using VMware, create a new VM using the VMware vSphere Client. Install Windows on the VM following standard VMware procedures, and install VMware Tools.

Install Eucalyptus Windows Integration

To install the Eucalyptus Windows Integration Service:

1. Download the most recent version of the Windows Image Preparation Tool from *http://downloads.eucalyptus.com/software/tools/windows-prep/* to the working directory of your NC or the host running the vSphere client.
2. For KVM and Xen, open the libvirt-[hypervisor]-windows-example.xml file you used in the previous section and make the following edits:

   - Comment out the lines of XML code directing the hypervisor to boot the Windows image from the CDROM.
   - Change the text so that `windows-prep-tools-3.1.0.iso` replaces the Windows .iso image and is mounted as `cdrom`.
   - Enter the path to the secondary disk file you created in the previous task.
   - Uncomment the lines that direct attachment of a floppy disk, secondary disk, and secondary network interface.

   Tip: If you plan on using virtio networking for instances (via USE_VIRTIO_NET option on node controllers), uncommenting the virtio interface in the xml is mandatory

   Your finished file should look similar to one of the following examples.

   For Xen:

```
<domain type='xen'>
    <name>eucalyptus-windows</name>
    <os>
            <type>hvm</type>
            <loader>/usr/lib/xen/boot/hvmloader</loader>
            <!-- <boot dev='cdrom'/> -->
    </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='localtime'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
    <memory>524288</memory>
    <vcpu>1</vcpu>
    <devices>
        <emulator>/usr/lib64/xen/bin/qemu-dm</emulator>
        <disk type='file'>
            <source file='/root/windows_2003.img'/>
            <target dev='hda' bus='ide'/>
        </disk>
        <disk type='file' device='disk'>
            <driver name='tap' type='aio'/>
            <source file='/root/secondary.img'/>
            <target dev='xvda' bus='xen'/>
        </disk>
        <disk type='file' device='floppy'>
             <source file='/root/floppy.img'/>
             <target dev='fda'/>
        </disk>
        <disk type='file' device='cdrom'>
            <source file='windows-prep-tools-3.1.0.iso'/>
            <target dev='hdc'/>
            <readonly/>
        </disk>
        <interface type='bridge'>
            <source bridge='xenbr0'/>
            <script path='/etc/xen/scripts/vif-bridge'/>
        </interface>
        <serial type='pty'>
           <source path='/dev/pts/3'/>
           <target port='0'/>
        </serial>
        <input type='tablet' bus='usb'/>
        <input type='mouse' bus='ps2'/>
        <graphics type='vnc' port='-1' autoport='yes' keymap='en-us'
listen='0.0.0.0'/>
    </devices>
</domain>
```

For KVM:

```
<domain type="kvm">
   <name>eucalyptus-windows</name>
    <os>
    <type>hvm</type>
    <!-- <boot dev='cdrom'/> -->
    </os>
    <features>
        <acpi/>
    </features>
    <memory>524288</memory>
    <vcpu>1</vcpu>
    <devices>
        <emulator>/usr/libexec/qemu-kvm</emulator>
        <disk type='file'>
            <source file='/root/windows_2003.img'/>
            <target dev='hda'/>
        </disk>
        <disk type='file' device='disk'>
             <source file='/root/secondary.img'/>
             <target dev='vda' bus='virtio'/>
        </disk>
        <disk type='file' device='floppy'>
             <source file='/root/floppy.img'/>
             <target dev='fda'/>
        </disk>
        <disk type='file' device='cdrom'>
            <source file='windows-preps-tools-3.1.0.iso'/>
            <target dev='hdc'/>
            <readonly/>
        </disk>
        <interface type='bridge'>
            <source bridge='br0'/>
            <model type='rtl8139'/>
        </interface>
        <interface type='bridge'>
            <source bridge='br0'/>
            <model type='virtio'/>
        </interface>
        <graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0'/>
    </devices>
</domain>
```

3. There is an issue with Xen paravirtualization drivers on some Windows versions. 64-bit Windows Server 2008, Windows Server 2008R2, and Windows 7 have driver signing requirements that prevent unsigned drivers from being installed. The supplied drivers in the package are not signed. To resolve this issue:

   a) Log in to the Windows VM.

   b) Launch `cmd.exe` as administrator, and execute the following commands:

   ```
   bcdedit.exe -set loadoptions DISABLE_INTEGRITY_CHECKS
   bcdedit.exe -set TESTSIGNING ON
   ```

   c) Reboot the VM.

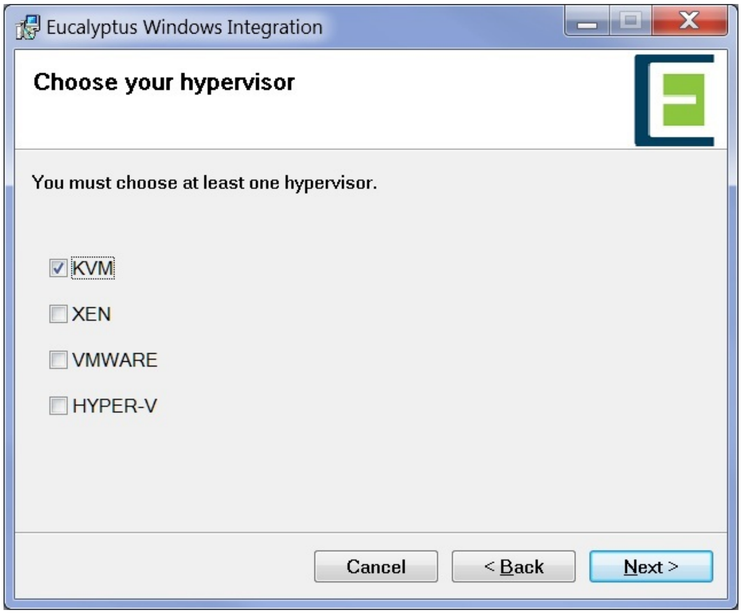
4. Start the VM with the newly modified libvirt xml file:

```
virsh create libvirt-[hypervisor]-windows.xml
```

5. For VMware, use the VMware vSphere client to upload the ISO file to the VSphere datastore. Attach the `windows-prep-tools-3.1.0.iso` to the Windows VM.
6. Log in to Windows and find the Eucalyptus installation files in the CDROM drive.

   - For Windows Server 2003 R2, run `setup.exe`. This automatically installs the .NET framework 2.0, which is not bundled in Server 2003 R2.
   - For all other versions, run `EucalyptusWindowsIntegration.msi`. (setup.exe will automatically install .NET framework 2.0, which is not bundled in Server 2003 R2).

7. In the **Choose your hypervisor** step, select your hypervisor and then click **Next**.



Choose your hypervisor

You must choose at least one hypervisor.

☑ KVM

□ XEN

□ VMWARE

□ HYPER-V

Click **Next** and continue until the end of installation.

8. Reboot the Windows VM
9. For KVM and Xen, open the Windows device manager and check that the following drivers are found for each device.

For KVM:

- Floppy disk drive
- Disk drivers: Red Hat VirtIO SCSI Disk Device
- SCSI and RAID controllers: Red Hat VirtIO SCSI controller
- Network adapters: Red Hat VirtIO Ethernet Adapter

For Xen:

- Floppy disk drive
- Disk drivers: XEN PV SCSI Disk Device
- SCSI and RAID controllers: Xen Block Device Driver
- Network adapters: Xen Net Device Driver

If the correct drivers are not found, question marks display on the devices. To install the devices, do the following:

- Right-click on the devices in question and select **Update Drivers** to open the New Hardware Wizard.
- When the new hardware wizard asks if Windows update is to be connected, click **No, not this time**.
- Choose **Install software automatically (recommended)**.
- If a confirmation popup message displays, click **Continue**.

You are now ready to *Configure Active Directory*.

Configure Active Directory

The Eucalyptus Integration service lets an enterprise with existing Active Directory domains attach Windows instances to the domains and control access to these instances using the existing AD user database. Users can log into the instance either using their domain credentials or the Administrator's password generated with the `euca-get-password` command.

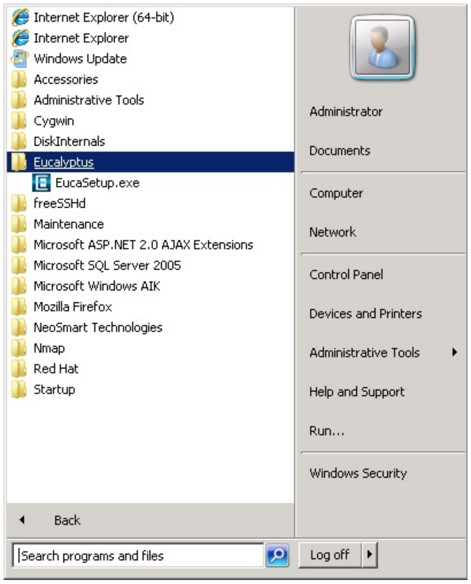
Because AD technology is tightly integrated with domain name service (DNS), the default name server contacted by the instance must be able to resolve the AD address as a proper domain controller. You can do this for all networking modes except System, by configuring the following line the CC's eucalyptus.conf file:

```
VNET_DNS=<domain_controller_IP_address>
```

If there is no such pre-existing DNS set-up or your networking mode is System, you might need to change the VM's network interface so that the preferred DNS server points to the domain controller.

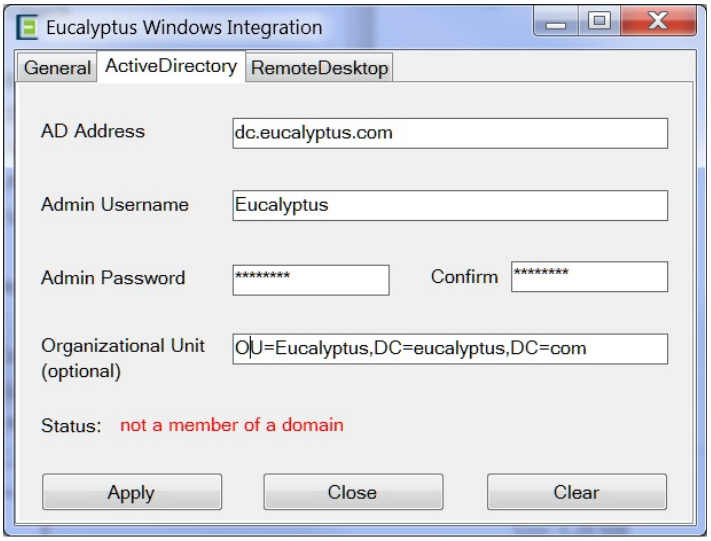
To set up Active Directory:

1. Click **Windows Programs** > **Eucalyptus** > **Eucalyptus Setup**.



The **Eucalyptus Windows Integration** popup displays.

2. Click the **Active Directory** tab in the Eucalyptus Windows Integration window and enter the following information:



AD Address: dc.eucalyptus.com

Admin Username: Eucalyptus

Admin Password: ******** Confirm: ********

Organizational Unit (optional): OU=Eucalyptus,DC=eucalyptus,DC=com

Status: not a member of a domain

- Enter the address of the existing Active Directory domain controller in the **AD Address** field.
- Enter the administrator username in the **Admin Username** field. We recommend using a generic user account that has permission to join a computer to a domain or a specific organizational unit.
- Enter and confirm the password in the **Admin Password** field. Note that the Admin username and password are required to join an instance to an Active Directory. When launched in Eucalyptus, these properties will be deleted as soon as the instance joins (or fails to join) the domain.
- Optionally, enter an organizational unit in the **Organizational Unit** field. This specifies a container that the instances launched from this image will be attach to.

Tip: If the values entered in this section are incorrect, the launched instances will fail to join the domain. We recommend that you verify the information by manually joining a computer to a domain using the same information that you entered in this step. You may first log in the launched instance using the administrator password (`euca-get-passcword`) and manually join the domain for verification.

3. Click **Apply**.

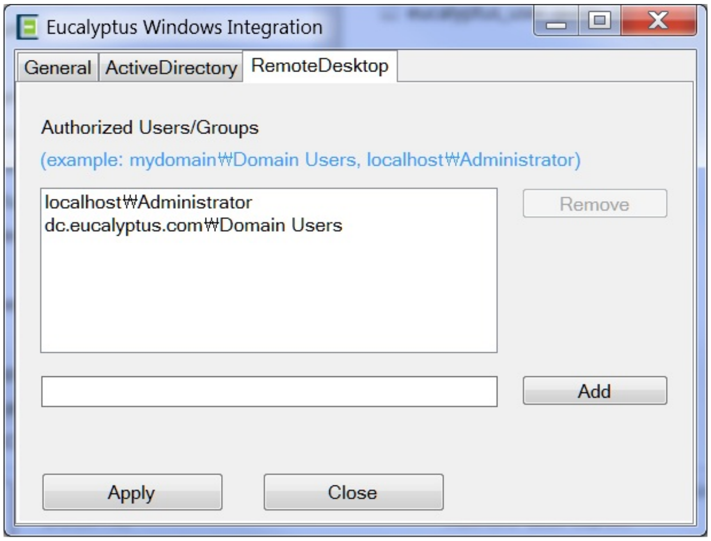
You are now ready to *Configure Remote Desktop*.

Configure Remote Desktop

Domain users or groups require remote desktop permission to log into an instance. By default, only the local administrator has the remote desktop permission. The Eucalyptus Integration Service provides a way to grant remote desktop permission to additional domain users or groups.

To configure remote desktop permission:

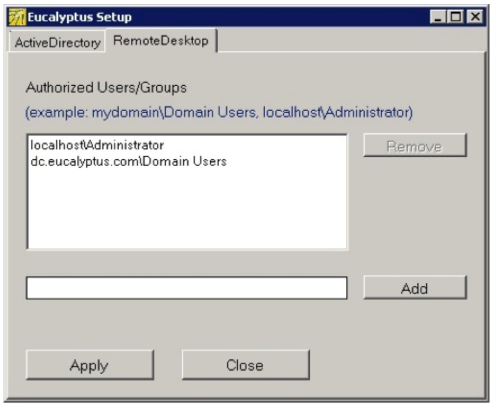
1. Open the **Eucalyptus Windows Integration** popup (**Windows Programs** > **Eucalyptus** > **Eucalyptus Setup**).
2. Click the **RemoteDesktop** tab


Eucalyptus Windows Integration
General
ActiveDirectory
RemoteDesktop
Authorized Users/Groups
(example: mydomain₩Domain Users, localhost₩Administrator)
Remove
Add
Apply
Close

localhost₩Administrator
dc.eucalyptus.com₩Domain Users

The names of authorized domain users and groups display in the **Authorized User/Groups** field. By default, only the local administrator is listed as authorized.

3. In the text field below the list, enter a user and group account name in the format `[DOMAIN]\[USER or GROUP]`. If you add a new local user or local group, prepend the account name `localhost\` instead of the domain name.



localhost\Administrator
dc.eucalyptus.com\Domain Users

4. Click **Add**.
5. Repeat for all user/groups that you want to add.
6. Click **Apply**.
   When the instance launches, the members of the groups you added can log in to the instance through remote desktop.

You are now ready to *Run Sysprep*.

Run Sysprep

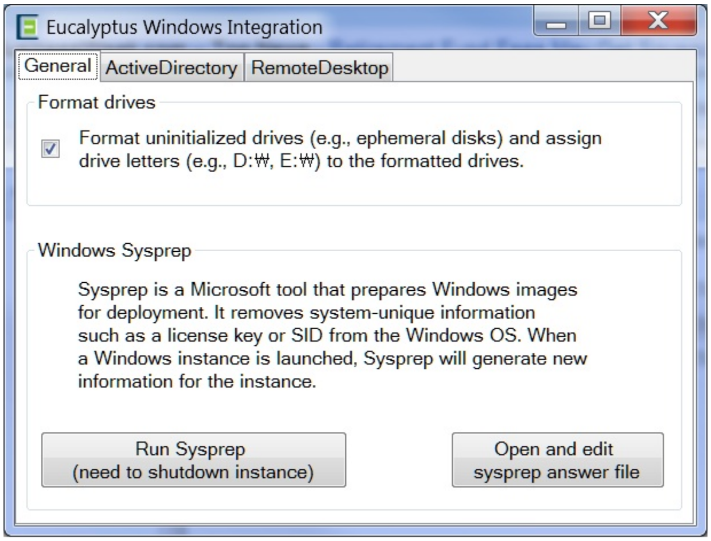
Sysprep is a Microsoft tool for deploying multiple Windows operating systems in an enterprise. Running Sysprep removes system-specific information such as security ID (SID) from the Windows OS before you clone an image. Sysprep then re-initializes the OS after the image is cloned and started on multiple computers. Use Sysprep to prepare images when you use Microsoft Key Management Service to activate license keys. Also, use Sysprep when your Windows systems are attached to Active Directory.

In Eucalyptus, you can run Sysprep before you bundle images with the `euca-bundle-image` or `euca-bundle-instance` commands.

Note: The Eucalyptus Integration Service supports Sysprep for Windows Server 2008, Windows Server 2008 R2, and Windows 7.

To configure and run Sysprep:

1. Open the **Eucalyptus Windows Integration** popup (**Windows Programs** > **Eucalyptus** > **Eucalyptus Setup**).


Eucalyptus Windows Integration
General
ActiveDirectory
RemoteDesktop
Format drives
Windows Sysprep
Run Sysprep (need to shutdown instance)
Open and edit sysprep answer file

☑ Format uninitialized drives (e.g., ephemeral disks) and assign drive letters (e.g., D:₩, E:₩) to the formatted drives.

Sysprep is a Microsoft tool that prepares Windows images for deployment. It removes system-unique information such as a license key or SID from the Windows OS. When a Windows instance is launched, Sysprep will generate new information for the instance.

2. Click the **General** tab.
3. Ensure that **Format uninitialized drives** is checked
4. If you want to edit the Sysprep answer file, click the **Open and change** button in the General tab. Otherwise skip this step.
5. Click the **Run Sysprep** button.
   Sysprep starts.
6. After Sysprep is complete, close the application and shutdown the Windows VM using the Windows Programs menu.

You are now ready to *Add Image to Eucalyptus*.

Add Image to Eucalyptus

To enable an image as an executable entity, you must bundle and upload the Windows disk image to Walrus, and then register the uploaded image with Eucalyptus.

Run the following command to bundle, upload, and register your Windows disk image:

```
euca-bundle-image -i <vm_image_file>
euca-upload-bundle -b <image_bucket> -m /tmp/<vm_image_file>.manifest.xml
euca-register <image_bucket>/<vm_image_file>.manifest.xml
```

Your Windows image is now ready to run as an instance.

After you register the image, Walrus decrypts the image bundle. This process might take a few minutes for a large Windows image to be decrypted. For example, a 10G image requires that you wait about 10 minutes before you launch the instance.

Create Linux Image (Xen/KVM)

Important: Before you begin, make sure that your hypervisor is installed and properly functioning for the account you are going to use.

To create a root filesystem to be used with Eucalyptus using Xen or KVM and an ISO image of the guest OS you want to install:

1. Download the ISO image of the distro (guest OS) you want to install.
2. Create a virtual disk with the desired size. For example, to create a 4 GiB disk:

```
dd if=/dev/zero of=<image_name> bs=1M count=4096
```

3. If you are using Xen, complete the following steps:

a) Create a configuration file (for example, `xen.cfg`) to boot off the ISO image. The file should look like the following:

```
name = "bootFromISO"
kernel = "/usr/lib/xen/boot/hvmloader"
memory = 1024
builder = "hvm"
device_model = "/usr/lib64/xen/bin/qemu-dm"
boot = "d"
disk = ['file:PATH_TO_ISO,hdc:cdrom,r',
'file:PATH_TO/<image_name>,sda,w']
vif = ['mac=00:01:01:00:00:03, bridge=xenbr0']
dhcp="on"
vnc = 1
vncdisplay = 7
pae = 1
```

Important: You must modify `device_model` and `kernel` since they depend on where the distribution puts such files. Look into the Xen package file list. Also be sure that all the options are correct for your needs.

b) Start the domU.

```
xen create xen.cfg
```

c) Connect with a VNC viewer. For example:

```
xvncviewer localhost:7
```

4. If you are using KVM, start KVM using the CDROM as the boot device and new image as the disk.

```
/usr/libexec/qemu-kvm -cdrom iso_image -drive
if=scsi,file=<image_name>,boot=off
```

Note: You need to have a SCSI as a bus because Eucalyptus expects the disk to be in `/dev/sda`.

5. Install the distro as you would normally would. Do not use `lvm` or `md`. Install everything on one partition. Remember the partition on which the distro is installed.

Important: Most distributions use the MAC address to associate each network interface with an interface name. To ensure that each instance started from the image sees the appropriate network interfaces as `eth0` etc, you must remove this association. Refer to your distribution's documentation for information on how to do this. On Debian and Ubuntu, `/etc/udev/rules.d/*net*` may be relevant. On CentOS and RHEL, `/etc/network-scripts/ifcfg-eth0` may be relevant. Failure to remove this association may leave your instance unable to connect to the network.

6. If you are using Xen, stop the domU.

7. Run `parted` to find out the starting block and the block size of the root file system, in bytes.

```
parted <image_name>
```

8. Change the units shown by `parted`to blocks.

```
(parted) u
Unit? [compact]? b
```

9. Print the current partition table using `p`.
   Note the start block and block size for the partition to which you will extract the file system. This example uses a start block of `32256` and a block size of `1024000`.
10. Extract the file system.

```
dd if=<image_name> of=rootfs.img bs=1 skip=32256 count=1024000
```

Tip: If this process is running slowly, you can increase the block size. Set `bs` to a power of two, and divide `skip` and `count` by the same number. For example, using a block size of 512, the above becomes `dd if=<image_name> of=rootfs.img bs=512 skip=63 count=2000`.

You now have a root filesystem. Make sure that it is compatible with the kernel/initrd you are using in your cloud environment. In particular you may want to be sure you have the modules of the kernel you are going to use. Then you are ready to *add the image* to Eucalyptus.

Create Linux Image (VMware)

To create a root filesystem to be used with Eucalyptus using VMware and an ISO image of the guest OS you want to install:

1. Run an instance of the image.
2. Attach a volume to the instance.

```
euca-attach-volume -i <instance_id> -d <local_device_name> <volume_id>
```

3. Make a file system on the volume and mount it to `/mnt`.

```
mkfs -t ext3 /dev/vdb
mount /dev/vdb /mnt
```

4. SCP your Eucalyptus credential zip file to the instance.

```
scp -i yourkey.private euca2-xxxxxxx-x509.zip root@xxx.xxx.xxx.xxx:/mnt
```

5. SSH into the instance.

```
ssh -i yourkey.private root@xxx.xxx.xxx.xxx
```

6. Install the tools you need.

```
yum install rsync unzip
```

7. Change to the `/mnt` device.

```
cd /mnt
```

8. Unzip your Eucalyptus credentials zip to `/mnt/euca`.

```
unzip euca2-admin-x509.zip -d euca/
```

9. Install any other software you want on the instance so it will be part of your new image.

10. Remove the udev rules.

```
rm /etc/udev/rules.d/70*
```

You now have a root filesystem. Make sure that it is compatible with the kernel/initrd you are using in your cloud environment. In particular you may want to be sure you have the modules of the kernel you are going to use. Then you are ready to *add the image* to Eucalyptus.

Bundle an Image for Amazon EC2

EMIs can be uploaded unchanged to Amazon EC2 and run as AMIs in the public cloud. To upload an EMI image file to Amazon, do the following:

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 following command:

```
euca-bundle-image -i <image_name> -r <architecture>\
-c <cert_filename> -k <private_key_filename> \
--ec2cert <path_to_cert_file>
```

For example:

```
euca-bundle-image -i euca-centos-5.3 -r x86_64\
  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
```

Bundle a Windows Instance

The euca-bundle-instance command lets you bundle a new Windows image from a running Windows 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.

To bundle a Windows instance:

Enter the following command:

```
euca-bundle-instance -b <bucket_name> -p
<prefix_starting_with_windows> -o $EC2_ACCESS_KEY -w
$EC2_SECRET_KEY <instance_ID>
```

Tip: You can query the progress of a bundling tasks using the `euca-describe-bundle-task` command. You can cancel the active bundle task using `euca-cancel-bundle-task` command.

The following example bundles and registers a new Windows image from an existing Windows instance with ID `i-12c4af6a`, to the bucket `mybucket`, with image prefix `windows2003_bundle`:

```
euca-bundle-instance -b mybucket -p windows2003_bundle -o
$EC2_ACCESS_KEY -w $EC2_SECRET_KEY  i-12c4af6a
```

```
euca-register mybucket/windows2003_bundle.manifest.xml
```

Using Instances

After an virtual image is launched, the resulting running system is called an instance. This section gives information about instances and their basic characteristics. It also describes how to launch an instance and connect to it.

Instance Overview

An instance is a virtual machine. A virtual machine is essentially an operational private computer that contains an operating system, applications, network accessibility, and disk drives. Eucalyptus allows you to run instances from both Linux-based images and Windows-based images.

The following sections describe each action that you can perform an instance.

Instance Tasks

Find an Image

To find an image:

1. Enter the following command:

   ```
   euca-describe-images
   ```

   The output displays all available 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. Look for the image ID in the second column and write it down. The image ID starts with `emi-`.

Once you find a suitable image to use, make sure you have a *keypair* to use.

Create Key Pairs

Eucalyptus uses cryptographic key pairs to verify access to instances. Before you can run an instance, you must create a key pair. Creating a key pair 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 using 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.

Create Key Pairs with the Console

1. From the main dashboard screen, click the **Key Pairs** link in the **Network and Security** section, or select the Network and Security submenu from the Manage Resources navigation menu. The **Manage Keypairs** screen will appear.
2. On the **Manage Key Pairs** screen, click the **Create new key pair** link. The **Create New Key Pair** dialog will appear.
3. Type a name for the new key pair into the **Name** text box.

4. Click the **Create and Download** button. The private half of the key pair is saved to the default download location for your browser.

   Note: Keep your private key file in a safe place. If you lose it, you will be unable to access instances created with the key pair.

5. Change file permissions to enable access to the private key file in the local directory. For example, on a Linux or Mac OS X system:

```
chmod 0600 <keypair_name>.private
```

Create Key Pairs with the Command Line

1. Enter the following command:

```
euca-add-keypair <keypair_name> > <keypair_name>.private
```

   where `<keypair_name>` is a unique name for your keypair. For example:

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

   The private key is saved to to a file in your local directory.
2. Change file permissions to enable access to the private key file in the local directory:

```
chmod 0600 <keypair_name>.private
```

3. Query the system to view the public key:

```
euca-describe-keypairs
```

   The command returns output similar to the following:

```
KEYPAIR alice-keypair
ad:0d:fc:6a:00:a7:e7:b2:bc:67:8e:31:12:22:c1:8a:77:8c:f9:c4
```

Authorize Security Groups

Before you can log in to an instance, you must authorize access to that instance. This done by configuring a security group for that instance.

A security group is a set of networking rules applied to instances associated with a group. When you first create an instance, it is assigned to a default security group that denies incoming network traffic from all sources. To allow login and usage of a new instance, you must authorize network access to the default security group with the euca-authorize command.

To authorize a security group, use euca-authorize with the name of the security group, and the options of the network rules you want to apply.

```
euca-authorize <security_group>
```

Use the following command to grant unlimited network access using SSH (TCP, port 22) and VNC (TCP, ports 5900 to 5910) to the security group `default`:

```
euca-authorize -P tcp -p 22 -s 0.0.0.0/0 default
euca-authorize -P tcp -p 5900-5910 -s 0.0.0.0/0 default
```

Use the following command to grant unlimited network access using Windows Remote Desktop (TCP, port 3389) to the security group `windows`:

```
euca-authorize -P tcp -p 3389 -s 0.0.0.0/0 windows
```

Launch an Instance

To launch an instance:

1. Use the euca-run-instances command and provide an image ID and the name of a keypair, in the format `euca-run-instances <image_id> -k <mykey>` . For example:

```
euca-run-instances emi-EC1410C1 -k alice-keypair
```

Eucalyptus returns output similar to the following example.

```
RESERVATION r-460007BE alice alice-default
INSTANCE i-2F930625 emi-EC1410C1 0.0.0.0 0.0.0.0 pending alice-keypair
2010-03-29T23:08:45.962Z eki-822C1344 eri-BFA91429
```

2. Enter the following command to get the launch status of the instance:

```
euca-describe-instances <instance_id>
```

Log in to an Instance

For a Linux Instance

When you create an 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 Administrator's Guide.

To use an 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 the following command to view the IP addresses of your instance:

```
euca-describe-instances
```

Eucalyptus returns output similar to the following:

```
RESERVATION r-338206B5 alice default
INSTANCE i-4DCF092C  emi-EC1410C1  192.168.7.24   10.17.0.130   running
alice-keypair  0  m1.small  2010-03-15T21:57:45.134Z
```

Note that the public IP address appears after the image name, with the private address immediately following.

2. Look for the instance ID in the second field and write it down. Use this ID to manipulate and terminate this instance.
3. Use SSH to log into the instance, using your private key and the external IP address. For example:

```
ssh -i alice-keypair.private root@192.168.7.24
```

You are now logged in to your Linux instance.

For a Windows Instance:

Log into Windows VM instances using a Remote Desktop Protocol (RDP) client. An RDP prompts you for a login name and password. By default, Windows VM instances are configured with a single user (named `Administrator`) and a random password generated at boot time. So, 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.

To log into a Windows instance:

1. Use `euca-describe-groups` to make sure the port for remote desktop (3389) is authorized in your security group.

   The response from this command will look like the following example.

```
GROUP 955340183797 default default group
PERMISSION 955340183797 default ALLOWS tcp 3389 3389 FROM CIDR 0.0.0.0/0
```

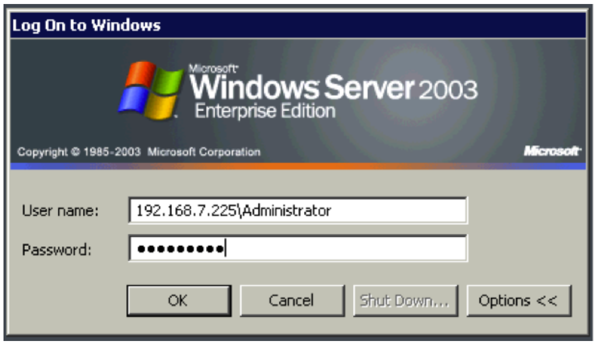
2. Enter the `euca-get-password` command followed by the unique id tag of the Windows VM instance and the `-k` option with the name of private key file that corresponds to your credential keypair. In the following example we retrieve the password for a Windows VM instance with id tag `i-5176095D` and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

```
euca-get-password i-5176095D -k mykey.private
```

3. Log into the RDP client using the public (external) IP address associated with the running Windows VM instance. Enter the following command to view the IP addresses of your instance:

```
euca-describe-instances
```

4. 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 are now logged in and ready to use your Windows instance.

Reboot an Instance

Rebooting preserves the root filesystem of an instance across restarts. To reboot an instance:

Enter the following command:

```
euca-reboot-instances <instance_id>
```

To reboot the instance `i-34523332`, enter:

```
euca-reboot-instances i-34523332
```

Terminate an Instance

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. You can obtain the ID strings of your instances using the euca-describe-instances command.

⚠ **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.

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:

```
euca-describe-instances
RESERVATION r-338206B5 alice 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:

```
euca-terminate-instances i-4DCF092C
 INSTANCE i-3ED007C8
```

Using EBS

Eucalyptus offers persistent storage that you can attach to a running instance. These Eucalyptus block storage (EBS) volumes persist autonomously from the running life of an instance. After you attach a block volume to an instance, you can use it like any other physical hard drive.

This section details what you need to know about EBS, as well as what you can do with EBS-backed instances.

EBS Overview

This section describes the qualities of an EBS image, including structural information.

EBS Image Structure

An EBS image has the following differences in structure from a standard image:

- Unlike a standard image, an EBS image uses a full hard disk image, including a partition table and a master boot record.
- When you register an EBS image, you do not specify a kernel or ramdisk. Instances are be booted using the bootloader within the image.

EBS Qualities

EBS images are characterized by the qualities detailed in the following list.

- **Size limit:** Regular images are limited to 10 GiBs for the root device. EBS-backed images, however, can be up to 1 TiB.
- **Stopped state:** A regular instance can only be terminated. You can't start a terminated instance. An EBS-backed instance, however, can be stopped and started again.
- **Data persistance:** EBS volumes store date independently of the life of an instance.
- **Boot times:** EBS-backed instances have faster boot times than regular instances.

EBS Tasks

Concept definition.

Use Case

The following steps detail all of the tasks that you would take to create and use an EBS volume. To create an image that boots from an external volume you need to have a pre-existing instance running in Eucalyptus. The following example uses an instance called `i-00000000` running in availability zone `zone1`.

Important: Make sure that the volume you create is large enough to hold the image you want to boot.

1. Create a new EBS volume in the same availability zone as the running instance.

```
euca-create-volume --zone zone1 --size 10
```

   The command displays the ID of the newly-created volume.

2. Attach the newly-created volume to the instance.

```
euca-attach-volume vol-00000000 -i i-00000000 -d /dev/sdf
```

3. Copy the image (not a bundle) you wish to register to the instance.
4. On the instance, copy the image onto the block device that corresponds to the EBS volume you attached.

```
dd if=/path/to/image of=/dev/sdf
sync
```

5. Detach the EBS volume from the instance.

```
euca-detach-volume vol-00000000
```

6. The EBS volumes that EBS-backed images use are created from EBS volume snapshots. When the EBS volume finishes detaching, create a snapshot of it. The command will display the ID of the newly-created snapshot.

```
euca-create-snapshot vol-00000000
```

7. When Eucalyptus finishes creating the snapshot, register a new machine image as you would for an instance-store image, but substitute `-snapshot` and the newly-created snapshot's ID where you would normally supply the path to an image manifest.

```
euca-register --kernel eki-00000000 --ramdisk eri-00000000 --name
"my-new-ebs-image" --snapshot snap-00000000
```

   When the command finishes it display the ID of the new EBS-backed machine image.

Create an EBS Image

This section explains how to create a Eucalyptus block storage-backed image. Please note the following considerations before you create an EBS image:

- Note that you cannot use a standard image as an EBS image without modification. This is due to the lack of a bootloader on a standard image.
- Also, because an EBS image must be written to a volume, you must already have a running instance in your cloud to load an EBS image.
- virtio systems need virtio_blk and virtio_balloon
- KVM without virtio systems need sym53c8xx
- Xen systems need xenblk and xenblk_front

You can create an EBS image from an *existing image* or create your *own image* for its use.

Create an EBS Image from an Existing Image

If you have been given a hard disk to be used as an EBS starter image:

1. Run an instance of an existing Eucalyptus machine image.

```
euca-run-instances <image_id>
```

   Note the instance ID returned by Eucalyptus.

   Tip: To find out the instance IP address, run `euca-describe-instances <instance_id>`.

2. Create a volume large enough to hold the hard disk image.

```
euca-create-volume --zone <partition_name> --size <volume_size>
```

   where:

- `zone` is the availability zone to create the volume in
- `size` is the size of the volume in GBs

Note the volume ID returned by Eucalyptus.

Tip: To find out when the volume creation completes, run `euca-describe-volume <volume_id>`. Wait for the volume's state to reach `available`.

3. Attach the volume to the instance.

```
euca-attach-volume <volume_id> -i <instance_id> -d <device_name>
```

Eucalyptus returns information about the attachment state.

4. Log in to the instance using your private SSH key and the instance's public IP address.

```
ssh -i mykey.private root@<instance_ip>
```

5. Copy the image (not a bundle) you want to register to the instance.

6. Examine `/proc/partitions` to determine the actual device name.

```
cat /proc/partitions
```

The response will be similar to the following:

```
major minor  #blocks  name

 252        0   10485791 vda
 252        1    1548288 vda1
 252        2    8413184 vda2
 252        3     524288 vda3
 252       16    5242880 vdb
```

The attached volume will typically be the last one in the list. In the previous example, this is `vdb`.

7. Copy the image onto the block device that corresponds to the volume you attached.

- If you already have a local copy of the image, run:

```
dd if=<image_file> bs=1M| ssh root@<instance_ip> "dd
of=/dev/<device> bs=1M"
```

- If you are downloading the image from a URL, in your existing SSH session run:

```
curl <image_url> > /dev/<device>
```

8. Detach the volume.

```
euca-detach-volume <volume_id>
```

9. Create a snapshot of the volume.

```
euca-create-snapshot <volume_id>
```

Note the snapshot ID returned by Eucalyptus.

10. Register the snapshot.

- For Linux:

  ```
  euca-register -n <image_name> --root-device-name /dev/sda1 -b
  /dev/sda1=<snapshot_id>
  ```

- For Windows:

  ```
  euca-register -n <image_name> --kernel windows --root-device-name /dev/sda1
   -b /dev/sda1=<snapshot_id>
  ```

You now have an EBS image.

Create a New EBS Image

To create your own EBS image, use one of the following options.

- Use `virt-install` on a system with the same distribution and hypervisor as your NC. If you create and successfully boot and connect the image to the network in this environment, it will boot from an EBS volume. Note that for CentOS or RHEL images, you may need to edit `/etc/sysconfig/network-scripts/ifcfg-eth0` and remove the `HWADDR` line. This is because an instance's network interface will always have a different hardware address.

  Warning: If you use an image created with virt-install under a different distribution or hypervisor combination, it is likely that it will not install the correct drivers into the ramdisk, and the image will not boot.

- Use a tool like boxgrinder that can create a hard disk image capable of booting under multiple hypervisors.

Detach a Block Volume

To detach a block volume from an instance:

Enter the following command:

```
euca-detach-volume <volume_id>
```

```
euca-detach-volume vol-00000000
```

Managing Access

Eucalyptus manages access to the cloud by policies attached to accounts, groups, and users. This section details access-related tasks you can perform once your administrator allows you access to Eucalyptus. These tasks are split into the following areas: tasks for groups, and tasks for users, and tasks for credential management.

Groups

Groups are used to share resource access authorizations among a set of users within an account. Users can belong to multiple groups.

Important: A group in the context of access is not the same as a security group.

This section details tasks that can be performed on groups.

Create a Group

Using the CLI

To create a group using the CLI:

Enter the following command:

```
euare-groupcreate -g <group_name>
```

Eucalyptus does not return anything.

Using the Eucalyptus Administrator Console

To create a group using the Eucalyptus Administrator Console:

1. Click **Accounts** in the Quick Links section.
   The **Accounts** page displays.
2. Click the **ID** of the account you want to add a group to.
   The account, name, and Registration status are highlighted.
3. Click **New groups** in the **Accounts** page.
   The **Create new groups** popup displays.
4. Enter the group name in the **Group name** field.

   Tip: You can add more than one group at a time. Every group you add, however, will be in the same path.

5. Enter the group path in the **Group path** field.
6. Click **OK**.

The group is associated with the account you chose. You can see the information if you select the account in the **Accounts** page and click the **Member groups** link, located in the **Properties** section of the screen.

Add a Group Policy

Using the CLI

To add a policy to a group using the CLI:

Enter the following command:

```
euare-groupaddpolicy -g <group_name> -p <policy_name> -e <effect> -a
       <actions> -o
```

The optional `-o` parameter tells Eucalyptus to return the JSON policy, as in this example:

```
{"Version":"2008-10-17","Statement":[{"Effect":"Allow",
"Action":["ec2:RunInstances"], "Resource":["*"]}]}
```

Using the Eucalyptus Administrator Console

To add a policy to a group using the Eucalyptus Administrator Console:

1. Click **Groups** in the Quick Links section.
   The **Groups** page displays.
2. Click the **ID** of the group you want to add a policy to.
   The ID, Name, Path, and Owner account line is highlighted.
3. Click **Add policy**.
   The **Add new policy** popup displays.
4. Enter the policy name in the **Policy name** field.
5. Enter the policy content in the **Policy content** field.
6. Click **OK**.

The policy is now added to the group.

Modify a Group

Modifying a group is similar to a "move" operation. Whoever wants to modify the group must have permission to do it on both sides of the move. That is, you need permission to remove the group from its current path or name, and put that group in the new path or name.

For example, if a group changes from one area in a company to another, you can change the group's path from `/area_abc/` to `/area_efg/`. You need permission to remove the group from `/area_abc/`. You also need permission to put the group into `/area_efg/`. This means you need permission to call `UpdateGroup` on both `arn:aws:iam::123456789012:group/area_abc/*` and `arn:aws:iam::123456789012:group/area_efg/*`.

Using the CLI

To modify a group using the CLI:

1. Enter the following command to modify the group's name:

   ```
   euare-groupmod -g <group_name> --new-group-name <new_name>
   ```

   Eucalyptus does not return a message.
2. Enter the following command to modify a group's path:

   ```
   euare-groupmod -g <group_name> -p <new_path>
   ```

   Eucalyptus does not return a message.

Using the Eucalyptus Administrator Console

To modify a group using the Eucalyptus Administrator Console:

1. Click **Groups** in the Quick Links section.
   The **Groups** page displays.
2. Click the **ID** of the group you want to rename.
   The group's **Properties** area displays.
3. In the **Name** field, enter the new name of the group.
4. In the **Path** field, enter the new path for the group.
5. Click **Save**.

The new group name displays in the **Groups** page.

Add a User to a Group

Using the CLI

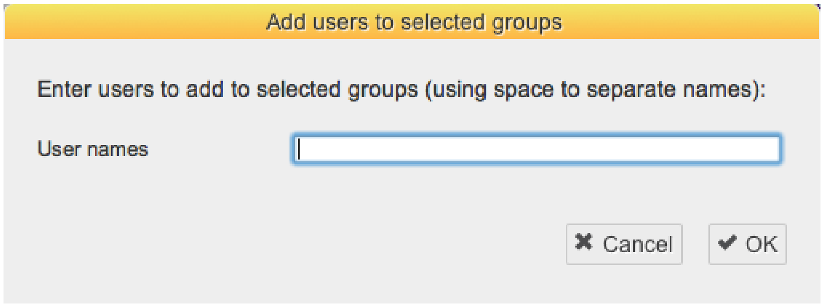
To add a user to a group using the CLI:

Enter the following command:

```
euare-groupadduser -g <group_name> -u <user-name>
```

Using the Eucalyptus Administrator Console

1. Click **Groups** in the Quick Links section.
   The **Groups** page displays.
2. Click the **ID** of the group you want to add the user to.
   The ID, Name, Path, and Owner account line is highlighted.
3. Click **Add users**.
   The **Add users to selected groups** popup displays.



Add users to selected groups

Enter users to add to selected groups (using space to separate names):

User names

4. Enter the name of the user you want to add and click **OK**.

The user is now added to the group.

Remove a User from a Group

Using the CLI

To remove a user from a group using the CLI:

Enter the following command:

```
euare-groupremoveuser -g <group_name> -u <user-name>
```

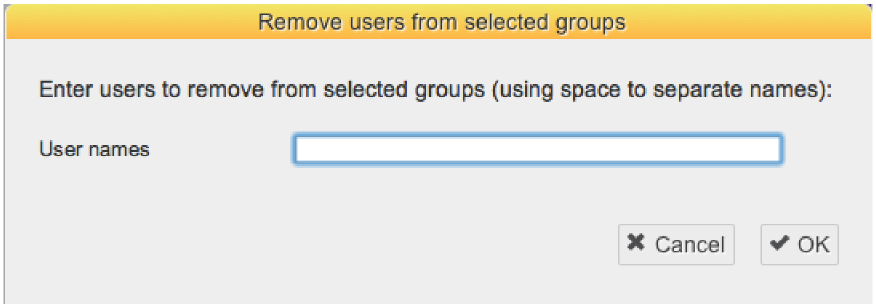
Using the Eucalyptus Administrator Console

To remove a user from a group using the Eucalyptus Administrator Console:

1. Click **Groups** in the Quick Links section.

The **Groups** page displays.

2. Click the **ID** of the group you want to remove the user from.
   The ID, Name, Path, and Owner account line is highlighted.
3. Click **Remove users**.
   The **Remove users to selected groups** popup displays.



Remove users from selected groups

Enter users to remove from selected groups (using space to separate names):

User names

4. Enter the name of the user you want to remove and click **OK**.

The user is now removed from the group.

Delete a Group

Using the CLI

When you delete a group, you have to remove users from the group and delete any policies from the group. You can do this with one command, using the `euare-groupdel` command with the `-r` option. Or you can follow the following steps to specify who and what you want to delete.

1. Individually remove all users from the group.

```
euare-groupremoveuser -g <group_name> -u <user_name>
```

2. Delete the policies attached to the group.

```
euare-groupdelpolicy -g <group_name> -p <policy_name>
```

3. Delete the group.

```
euare-groupdel -g <group_name>
```

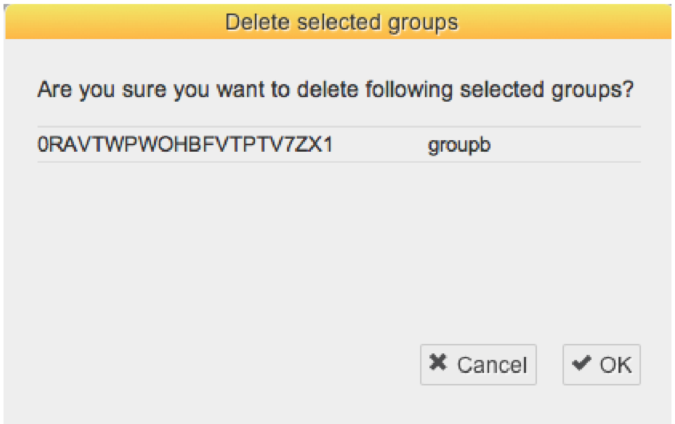
The group is now deleted.

Using the Eucalyptus Administrator Console

To delete a group using the Eucalyptus Administrator Console:

1. Click **Groups** in the Quick Links section.
   The **Groups** page displays.
2. Click the **ID** of the group you want to delete.
   The ID, Name, Path, and Owner account line is highlighted.
3. Click **Delete groups**.

   The **Delete selected groups** popup displays.



Are you sure you want to delete following selected groups?

0RAVTWPWOHBFVTPTV7ZX1 groupb

4. Click **OK**.

The group is now deleted.

List Users

You can list users within a path.

Using the CLI

Use the `euare-userlistbypath` command to list all the users in an account or to list all the users with a particular path prefix. The output lists the ARN for each resulting user.

```
euare-userlistbypath -p <path>
```

Using the Eucalyptus Administrator Console

To list users in the same path using the Eucalyptus Administrator Console:

1. Click **Users** in the Quick Links section.
   The **Users** page displays.
2. Click the **Path** column to sort all users by path.

Users

Users are subsets of accounts and are added to accounts by an appropriately credentialed administrator. While the term **user** typically refers to a specific person, in Eucalyptus, a **user** is defined by a specific set of credentials generated to enable access to a given account. Each set of user credentials is valid for accessing only the account for which they were created. Thus a user only has access to one account within a Eucalyptus system. If an individual person wishes to have access to more than one account within a Eucalyptus system, a separate set of credentials must be generated (in effect a new ‘user’) for each account (though the same username and password can be used for different accounts).

When you need to add a new user to your Eucalyptus cloud, you'll go through the following process:

1	*Create a user*
2	*Add user to a group*
3	*Give user a login profile*

Add a User

Using the CLI

To add a user using the CLI:

Enter the following command

```
euare-usercreate -u <user_name> -g <group_name> -k
```

Eucalyptus does not return a response.

Tip: If you include the `-v` parameter, Eucalyptus returns a response that includes the user's ARN and GUID.

Using the Eucalyptus Administrator Console

To add a user using the Eucalyptus Administrator Console:

1. Click **Accounts** in the Quick Links section.
   The **Accounts** page displays.
2. Click the **ID** of the account you want to rename.
   The account's **Properties** area displays.
3. Click **New Users**.
   The **Create new users** popup window displays.
4. Enter a name in the **User names** field.

   Tip: You can add more than one user at a time. Every user you add, however, will be in the same path.

5. Enter a path in the **User path** field.
6. Click **OK**.

The user is added to the account.

Create a Login Profile

Once you create a user, you must generate a password for the user to use the Eucalyptus Administrator Console.

Using the CLI

To create a login profile using the CLI:

Enter the following command:

```
euare-useraddloginprofile -u <user_name> -p <password>
```

Eucalyptus does not return a response.

Using the Eucalyptus Administrator Console

To create a login profile using the Eucalyptus Administrator Console:

1. Click **Users** in the Quick Links section.
   The **Users** page displays.
2. Click the **ID** of the user whose path you want to change.
   The user's **Properties** area displays.
3. Click **Password**.
   The **Change password** popup window displays.
4. Enter the new password in the **New user password** field, and repeat in the **New password again** field.
5. Click **OK**.

The login profile is now complete. If you are generating the password for a different user, let the user know the password and the URL to the Eucalyptus Administrator Console.

Modify a User

Modifying a user is similar to a "move" operation. Whoever wants to modify a user must have permission to do it on both sides of the move. That is, you need permission to remove the user from the current path or name, and put that user in the new path or name.

For example, if a user changes from one team in a company to another, you can change the user's path from `/team_abc/` to `/team_efg/`. You need permission to remove the user from `/team_abc/`. You also need permission to put the user into `/team_efg/`. This means you need permission to call UpdateUser on both `arn:aws:iam::123456789012:user/team_abc/*` and `arn:aws:iam::123456789012:user/team_efg/*`.

Using the CLI

To rename a user using the CLI:

1. Enter the following command to rename a user:

   ```
   euare-usermod -u <user_name> --new-user-name <new_name>
   ```

   Eucalyptus does not return a message.
2. Enter the following command:

   ```
   euare-groupmod -u <user_name> -p <new_path>
   ```

   Eucalyptus does not return a message.

Using the Eucalyptus Administrator Console

To rename a user using the Eucalyptus Administrator Console:

1. Click **Users** in the Quick Links section.
   The **Users** page displays.
2. Click the **ID** of the user you want to rename.
   The user's **Properties** area displays.
3. In the **Name** field, enter the new name of the user.
4. Click **Save**.

The new user name displays in the **Users** page.

Change User Path

Using the CLI

To change a user's path using the CLI:

Enter the following command:

```
euare-groupmod -u <user_name> -p <new_path>
```

Eucalyptus does not return a message.

Using the Eucalyptus Administrator Console

To change a group's path using the Eucalyptus Administrator Console:

1. Click **Users** in the Quick Links section.
   The **Users** page displays.
2. Click the **ID** of the user whose path you want to change.
   The user's **Properties** area displays.

3. In the **Path** field, enter the new path for the user.
4. Click **Save**.

The user's path is now changed and displays in the **Users** page.

Create User Password

Context for the current task

Task step.

List Users

You can list users within a path.

Using the CLI

Use the `euare-userlistbypath` command to list all the users in an account or to list all the users with a particular path prefix. The output lists the ARN for each resulting user.

```
euare-userlistbypath -p <path>
```

Using the Eucalyptus Administrator Console

To list users in the same path using the Eucalyptus Administrator Console:

1. Click **Users** in the Quick Links section.
   The **Users** page displays.
2. Click the **Path** column to sort all users by path.

List Groups

Using the CLI

To list all the groups a specific user is in:

Enter the following command:

```
euare-grouplistbypath
```

Eucalyptus returns a list of paths followed by the ARNs for the groups in each path. For example:

```
arn:aws:iam::eucalyptus:group/groupa
```

Using the Eucalyptus Administrator Console

To list groups using the Eucalyptus Administrator Console:

Click **Groups** in the Quick Links section.
The **Groups** page displays.

The **Groups** page displays all groups in your cloud.

Delete a User

Using the CLI

To delete a user using the CLI:

Enter the following command

```
euare-userdel -u <user_name>
```

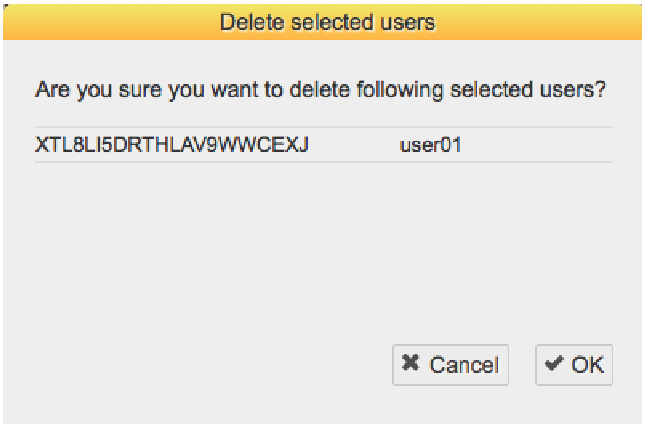
Eucalyptus does not return a response.

Using the Eucalyptus Administrator Console

To delete a user using the Eucalyptus Administrator Console:

1. Click **Users** in the Quick Links section.
   The **Users** page displays.
2. Click the **ID** of the user you want to delete.
   The user's information is highlighted.
3. Click **Delete Users**.

   The **Delete selected users** popup window displays.



Delete selected users

Are you sure you want to delete following selected users?

XTL8LI5DRTHLAV9WWCEXJ user01

4. Click **OK**.

The user is deleted.

Credentials

Eucalyptus uses different types of credentials for different purposes. This section details tasks needed to allow access to Eucalyptus services.

Create Credentials

You can generate new credentials a number of ways. The first time you get credentials using either the Eucalyptus Administrator Console or the `euca_conf` command, a new secret access key is generated. On each subsequent request to get credentials, an existing active secret Key is returned. You can also generate new keys using the `eucare-useraddkey` command.

Tip: Each request to get a user's credentials either via the download link in the Eucalyptus Administrator Console or using euca_conf, a new pair of a private key and X.509 certificate

- To generate a new key for a user by an account administrator, enter the following

```
euare-useraddkey -u <user_name>
```

- To generate a private key and an X.509 certificate pair, enter the following:

```
euare-usercreatecert -u <user_name>
```

Get Credentials

Eucalyptus provides two main ways of getting user credentials. In both cases, Eucalyptus returns a zip file that contains keys, certificates, a bash script, and several other required files. To use these credentials with such CLI tools as euca2ools or ec2-tools, unzip your credentials zip file to a directory of your choice.

- An administrator with a root access to the machine on which CLC is installed can get credentials using euca_conf CLI tool on that machine.

```
/usr/sbin/euca_conf --cred-account <account> --cred-user <user_name>
 --get-credentials <filename>.zip
```

  Where <account> and <user_name> are the names of the account and the user whose credentials are retrieved.

  Tip: You can omit the `--cred-account` and `--cred-user` options when you get credentials for the **admin** user of the **eucalyptus** account.

- A user can get his or her credentials by logging in into the Eucalyptus Administrator Console and clicking **Download new credentials** in the drop-down menu at the top of the screen. This will result in a download of a zip file.

In the following example we download the credentials zip file to `~/.euca`, then change access permissions, as shown:

```
mkdir ~/.euca
cd ~/.euca
unzip <filepath>/<creds_zipfile>.zip
chmod 0700 ~/.euca
chmod 0600 *
```

Important: The zip file with credentials contains security-sensitive information. We recommend that you remove or read- and write-protect the file from other users after unzipping.

Alternatively, you can view and copy your access keys and X.509 certificates from the Eucalyptus Administrator Console after logging in, using the Navigation menu.

Upload a Certificate

To upload a certificate provided by a user:

Enter the following command:

```
euare-useraddcert -u <user_name> -f <cert_file>
```

Using 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.

Associate an IP Address with an Instance

To associate an IP address with an instance:

1. Allocate an IP address:

```
euca-allocate-address ADDRESS <IP_address>
```

2. Associate the allocated IP address with an instance ID:

```
euca-associate-address -i <instance_ID> <IP_address>

```

```
euca-associate-address -i i-56785678 192.168.17.103
```

Release an IP Address

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.

To release an IP address:

1. Enter the following command to disassociate an IP address from an instance:

```
euca-disassociate-address <IP_address>
```

2. Enter the following command to release an IP address:

```
euca-disassociate-address <IP_address>
```

The following example releases the IP address, `192.168.17.103`

```
euca-release-address 192.168.17.103
```

Create a Security Group

Security groups let you control network access to instances by applying network rules to instances associated with a group.

To create a security group:

Enter the following command:

```
euca-add-group -d <description> <group_name>
```

Tip: You can also create a security group you run an instance. Use the `euca-run-instances` command with the `-g` option. Security group rules only apply to incoming traffic thus all outbound traffic is permitted.

The following example creates a new security group named `mygroup` and described as `newgroup`.

```
euca-add-group -d "newgroup" mygroup
```

Delete a Security Group

The euca-delete-group command lets you delete security groups. To delete a security group:

Enter the following command:

```
euca-delete-group <group_name>
```

The following example deletes the security group, `mygroup`.

```
euca-delete-group mygroup
```

Authorize 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.

To authorize security group rules:

Enter the following command:

```
euca-authorize -P <protocol> -p <port_number> \
-s <CIDR_source_network> <group_name>
```

The following example allows all incoming SSH traffic on port 22 to access to the security group mygroup. The CIDR source network, 0.0.0.0/0, refers to any source.

```
euca-authorize -P tcp -p 22 -s 0.0.0.0/0 mygroup
 GROUP mygroup
 PERMISSION mygroup ALLOWS tcp 22 22 FROM CIDR
```

Instead of specifying a CIDR source, you can specify another security group. The following example allows access to the security group mygroup from the someothergroup security group using SSH on port 22.

```
euca-authorize --source-group someothergroup \
--source-group-user someotheruser -P tcp -p 22 mygroup
```

Revoke Security Group Rules

To revoke security group rules:

Enter the following command:

```
euca-revoke -P <protocol> -p <port_number> -s <CIDR_source_network>
<group_name>
```

The following example revokes the network rules authorized for the security group mygroup.

```
euca-revoke -P tcp -p 22 -s 0.0.0.0/0 mygroup
```

Glossary

cluster
A group of resources that contains a CC, an SC and, optionally, a Broker.

availability zone
An availability zone for AWS denotes a large subset of their cloud environment. Eucalyptus refines 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.

AWS
Amazon Web Services

bucket storage
A storage container that accepts objects via PUT and GET commands.

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.

Cloud Controller
The Cloud Controller (CLC) is the entry-point into the cloud for administrators, developers, project managers, and end-users. The CLC queries the node managers [SM1] for information about resources, makes high-level scheduling decisions, and makes requests to the Cluster Controllers (CCs). As the interface to the management platform, the CLC is responsible for exposing and managing the underlying virtualized resources (servers, network, and storage). You can access the CLC through Amazon's Elastic Compute Cloud (EC2) and through a web-based Eucalyptus Administrator Console.

Cluster Controller
The Cluster Controller (CC) generally executes on a machine that has network connectivity to both the machines running the Node Controller (NC) and to the machine running the CLC. CCs gather information about a set of node machines and schedules virtual machine (VM) execution on specific nodes. The CC also manages the virtual machine networks and participates in the enforcement of SLAs[SM3] as directed by the CLC. All NCs associated with a single CC must be in the same broadcast domain (Ethernet).

dynamic block volume
A dynamic block volume is similar to a raw block storage device that can be used with VM instances. You can create, attach, detach, describe, bundle, and delete volumes. You can also create and delete snapshots of volumes and create new volumes from snapshots

elastic IP
Public IP addresses that you can reserve and dynamically associate with VM instances.

instance type
An instance type defines what hardware the instance has, including the amount of memory, disk space, and CPU power.

kernel/ramdisk pair
A ramdisk contains drivers that direct the kernel to launch appropriate system files when instantiating a virtual machine.

Node Controller
The Node Controller (NC) executes on any machine that hosts VM instances. The NC controls VM activities, including the execution, inspection, and termination of VM instances. It also fetches and maintains a local cache of instance images, and it queries and controls the system software (host OS and the hypervisor) in response to queries and control requests from the CC. The NC is also responsible for the management of the virtual network endpoint.

Storage Controller
The Storage Controller (SC) provides functionality similar to the [5] Amazon Elastic Block Storage (EBS) and is capable of interfacing with various storage systems (NFS, iSCSI, SAN devices, etc.). Elastic block storage exports storage volumes that can be attached by a VM and mounted or accessed as a raw block device. EBS volumes persist past VM termination and are commonly used to store persistent data. An EBS volume cannot be shared between VMs and can only be accessed within the same availability zone in which the VM is running. Users can create snapshots from EBS volumes. Snapshots are stored in Walrus and made available across availability zones. Eucalyptus with SAN support lets you use your enterprise-grade SAN devices to host EBS storage within a Eucalyptus cloud.

VMware Broker

VMware Broker (Broker or VB) is an optional Eucalyptus component activated only in versions of Eucalyptus with VMware support. VMware Broker enables Eucalyptus to deploy VMs on VMware infrastructure elements and mediates all interactions between the Cluster Controller (CC) and VMware hypervisors (ESX/ESXi) either directly or through VMware vCenter.

Walrus

Walrus allows users to store persistent data, organized as buckets and objects. You can use Walrus to create, delete, and list buckets, or to put, get, and delete objects, or to set access control policies. Walrus is interface compatible with Amazon's Simple Storage Service (S3), providing a mechanism for storing and accessing virtual machine images and user data. Note that Walrus access is global to the entire Eucalyptus cloud. This means that it can be accessed by end-users, whether the user is running a client from outside the cloud or from a virtual machine instance running inside the cloud.

Index