

Eucalyptus User's Guide (2.0)

This guide is for users of existing Eucalyptus 2.0 installations. If you would like to install Eucalyptus 2.0 on a cluster, see the Eucalyptus Administrator's Guide.)

Getting Started Using Eucalyptus 2.0

These instructions walk you through the essential steps for using a Eucalyptus cloud. Those who have worked with Amazon's EC2 system will find most of these instructions familiar (in fact, you may continue using Amazon's command-line tools with Eucalyptus).

1. Install command-line tools

The instructions below rely on the euca2ools command-line tools distributed by the Eucalyptus Team. Please, install them if you haven't done so already.

2. Sign up

If you are using the [Eucalyptus Community Cloud](#), use <https://ecc.eucalyptus.com> instead of **your.cloud.server**.

Load in your browser the Web page of the Eucalyptus cloud installation that you would like to use. Ask your system administrator for the URL if you don't know it. (The URL will be of the form <https://your.cloud.server:8443/>, where *your.cloud.server* is likely to be the front-end of the cluster.)



Please, sign in:

Username:

Password:

☐ Remember me on this computer

Sign in

[Apply](#) for account | [Recover](#) password

Click the "Apply" link and fill out the form presented to you. You may not be able to use the system until the (human) administrator receives the notification of your application and approves it. The more information you supply the easier it may be for the administrator to make the decision.

Please, fill out the form:

Mandatory fields:

Preferred username:

Password:

Password, again:

Full Name:

Email address:

Optional fields:

Telephone Number:

Project Leader:

Affiliation:

Project Description:

Sign up

 Or

Cancel

Load the confirmation URL that you receive in the approval email message from the cloud administrator. **Log in** to the system with the login and password that you chose when filling out the application form.

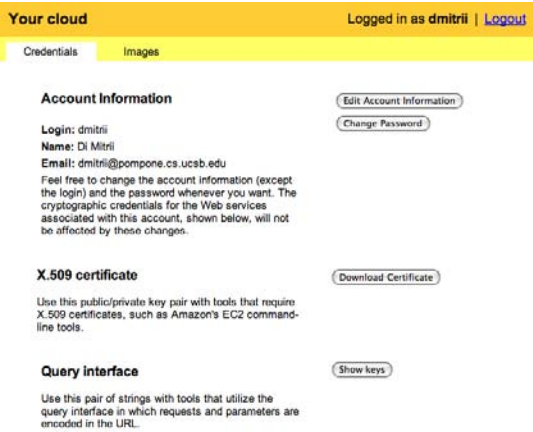
Your Eucalyptus account was approved! Please, click on the following link to log into the system (with the login and password that you've chosen when signing up) and confirm your account:

<https://localhost:8443/?action=confirm&code=4f53646f72796c50736c7773586773465f4e5f53697343786d47672e>

However, if you never requested a Eucalyptus account then, please, disregard this message.

3. Obtain Credentials

Once you have logged in, you will see the 'Generate Certificate' button under the 'Credentials' tab. Generating a certificate for your account is necessary before you can use Amazon's EC2 command-line tools for querying and controlling Eucalyptus instances. Currently, the Web interface to Eucalyptus is limited and, hence, the use of command-line tools is practically inevitable.



Click the button to generate the certificate and save it. You can keep these keys in a secure place on any host. The following command-line instructions apply to any Unix-flavored machine with bash (not necessarily the cluster where Eucalyptus was installed). (See Amazon's Getting Started Guide for the similar instructions to use under Windows.)

Unzip the keys using the following command and **protect** them from exposure. The zip-file contains two files with the .pem extension; these are your public and private keys.

```
mkdir ~/.euca
cd ~/.euca
unzip name-of-the-key-zip.zip
chmod 0700 ~/.euca
chmod 0600 ~/.euca/*
```

Finally, ensure that the environment variables necessary for euca2tools to work are set by sourcing the `eucaarc` file:

```
. ~/.euca/eucaarc
```

4. Euca2ools Quick Start

You are now ready to begin uploading and running VM instances in your Eucalyptus cloud. This section provides a quickstart guide to help familiarize you with frequently used Euca2ools (Eucalyptus' EC2-compatible command line tools) commands. We begin by introducing Euca2ools query commands that enable you to view information about the status of resources, images, and instances. Next, we show you how to create "keypairs" that are used to authenticate user identity. Finally, we show you how to run (instantiate), log into, and terminate VM instances.

4.1 Querying the System

The following Euca2ools commands let you query the system to view information about uploaded images, running instances, available clusters (zones), and uploaded keypairs:

```
euca-describe-images
euca-describe-instances
euca-describe-availability-zones
euca-describe-keypairs
```

4.2 Creating Keypairs

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

```
euca-add-keypair mykey | tee mykey.private
```

A pair of keys are created; one public key, stored in Eucalyptus, and one private key stored in the file `mykey.private` and printed to

standard output. The `ssh` client requires strict permissions on private keys:

```
chmod 0600 mykey.private
```

4.3 Running a VM Instance

You can now run instances that are accessible with the newly generated private key:

```
euca-run-instances -k mykey -n <number of instances to start> <emi-id>
euca-describe-instances
```

4.4 Authorizing Security Groups and Allocating IPs

If your administrator has configured Eucalyptus to provide security groups and elastic IPs, you may be required to allow logins to your instance, allocate a public IP (if you have not done so before, check 'euca-describe-addresses' as a reminder), and assign it to your running instance:

Allow 'ssh' connections from the Internet:

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

Allocate a public IP if you have not done so already:

```
euca-allocate-address
```

Associate an allocated IP with your running instance:

```
euca-associate-address <IP from allocate> -i <instance ID>
```

Once the instance is shown as 'Running', it will also show two IP addresses assigned to it.

4.5 Logging into a VM Instance

You can now log into it with the SSH key that you created:

```
ssh -i mykey.private root@<accessible-instance-ip>
```

To terminate instances, use:

```
euca-terminate-instances <instance-id1> <instance-id2> ... <instance-idn>
```

For more information on Euca2ools, see our Euca2ools User Guide. For more information on EC2 command line tools, see the EC2 Getting Started Guide. Please note that depending on the networking mode used to implement your Eucalyptus cloud, some command line tools may not be applicable (security groups/elastic IPs, etc.). For more information, consult your administrator or see Eucalyptus Network Configuration (2.0).

Euca2ools User Guide

Euca2ools are command-line tools for interacting with Web services that export a REST/Query-based API compatible with Amazon EC2 and S3 services. The tools can be used with both Amazon's services and with installations of the Eucalyptus open-source cloud-computing infrastructure. The tools were inspired by command-line tools distributed by Amazon (api-tools and ami-tools) and largely accept the same options and environment variables. However, these tools were implemented from scratch in Python, relying on the Boto library and M2Crypto toolkit.

Summary of Features

- Query of availability zones (i.e. clusters in Eucalyptus)
- SSH key management (add, list, delete)
- VM management (start, list, stop, reboot, get console output)
- Security group management
- Volume and snapshot management (attach, list, detach, create, bundle, delete)
- Image management (bundle, upload, register, list, deregister)
- IP address management (allocate, associate, list, release)

Installation from source

Euca2ools can be installed from source or as a binary package (DEB or RPM). The latest source tarball and binary packages can be found here:

<http://www.eucalyptus.com/download/eucalyptus>

Please, download the correct package for your distribution or the tarball. Euca2ools are written in Python, relying on the Boto library and

the M2Crypto cryptography and SSL toolkit. The acceptable versions for the dependencies are:

- Python 2.5 (dev) or higher
- Boto 1.9b or higher
- M2Crypto 0.20.2 or higher

In what follows substitute the desired version (e.g., 1.3.1) for \$VERSION either manually or by setting a shell variable. For example

```
export VERSION="1.3.1"
```

You will need to download euca2ools-\$VERSION-src-deps.tar.gz, which contains boto-1.9b.tar.gz and M2Crypto-0.20.2.tar.gz.

Build the dependencies and install as follows.

You will need to install python-dev, swig, help2man, and libssl-dev to build the following libraries.

- Download euca2ools-\$VERSION.tar.gz and euca2ools-\$VERSION-src-deps.tar.gz. Below, we will assume that these tarballs are located in the current directory.

- Install Boto

```
tar zxvf euca2ools-$VERSION-src-deps.tar.gz
cd euca2ools-$VERSION-src-deps
tar zxvf boto-1.9b.tar.gz
cd boto-1.9b
sudo python setup.py install
cd ..
```

- Install M2Crypto

```
tar zxvf M2Crypto-0.20.2.tar.gz
cd M2Crypto-0.20.2
sudo python setup.py install
cd ..
```

- Install Euca2ools in /usr/local/bin, adding it to your \$PATH, if necessary

```
cd ..
tar zxvf euca2ools-$VERSION.tar.gz
cd euca2ools-$VERSION
sudo make
export PATH=/usr/local/bin:$PATH # not necessary on most installations
```

- Uninstalling

```
sudo make uninstall
```

You may also wish to delete euca2ools, boto and M2Crypto from your python package installation directory.

Installing Euca2ools on CentOS 5.5

This document shows you how to install Euca2ools 1.3 from RPM packages on Centos 5.5. In the following steps, the value of \$VERSION must be substituted accordingly (e.g., as 1.2, 1.3.1, etc.). For example, we can set the value of 1.3.1 using bash:

```
export VERSION=1.3.1
```

In addition, the value of \$ARCH must be substituted with appropriate architecture (64-bit: x86_64 or 32-bit: i386). For example, for a 64-bit installation:

```
export ARCH=x86_64
```

There are two options for downloading and installing the packages:

1. Yum option

These packages are available from our yum repository. To use this option, create '/etc/yum.repos.d/euca.repo' file with the following four lines:

```
[euca2ools]
name=Euca2ools
baseurl=http://www.eucalyptussoftware.com/downloads/repo/euca2ools/$VERSION/yum/centos/
enabled=1
```

Now install euca2ools

```
yum install euca2ools.$ARCH --nogpgcheck
```

1. Tarball option

Download the appropriate tarball from <http://www.eucalyptus.com/download/eucalyptus>

Untar the bundle in a temporary location, install Python 2.5, and install euca2ools

```
tar zxvf euca2ools-$VERSION-*.tar.gz
cd euca2ools-$VERSION-*
sudo -s
yum install -y swig
rpm -Uvh python25-2.5.1-bashton1.$ARCH.rpm python25-libs-2.5.1-bashton1.$ARCH.rpm euca2ools-$VERSION-*. $ARCH.rpm
```

NOTE: please use '-Uvh' and not '-i'.

Installing Euca2ools on OpenSUSE 11.2

This document shows you how to install Euca2ools 1.3 from RPM packages on OpenSUSE 11.2. In the following steps, the value of \$VERSION must be substituted accordingly (e.g., as 1.2, 1.3.1, etc.) for example we can set the value of 1.3.1 using bash:

```
export VERSION=1.3.1
```

There are two options for downloading and installing the packages:

1. Zypper option

These packages are available from our repository. To use this option:

```
zypper ar --refresh http://www.eucalyptussoftware.com/downloads/repo/euca2ools/$VERSION/yum/opensuse Euca2ools
```

next, refresh the repository

```
zypper refresh Euca2ools
```

and answer "yes" to the question about trusting the packages:

```
File 'repomd.xml' from repository 'Euca2ools' is unsigned, continue? [yes/no] (no): yes
```

now install Euca2ools

```
zypper install euca2ools
```

1. Tarball option

Download the appropriate tarball for your architecture (64-bit: x86_64 or for 32-bit: i386) from <http://www.eucalyptus.com/download/eucalyptus>

Untar the bundle in a temporary location, and install euca2ools

```
tar zxvf euca2ools-$VERSION-*.tar.gz
cd euca2ools-$VERSION-*
sudo -s
zypper install swig
rpm -Uvh euca2ools-$VERSION-*.rpm
```

NOTE: please use '-Uvh' and not '-i'.

Installing Euca2ools on Fedora 12

This document shows you how to install Euca2ools 1.3 from RPM packages on Fedora 12. In the following steps, the value of \$VERSION must be substituted accordingly (e.g., as 1.2, 1.3.1, etc.) for example we can set the value of 1.3.1 using bash:

```
export VERSION=1.3.1
```

In addition, the value of \$ARCH must be substituted with appropriate architecture (64-bit: x86_64 or 32-bit: i386). For example, for a 64-bit installation:

```
export ARCH=x86_64
```

There are two options for downloading and installing the packages:

1. Yum option

These packages are available from our yum repository. To use this option, create '/etc/yum.repos.d/euca.repo' file with the following four lines:

```
[euca2ools]
name=Euca2ools
baseurl=http://www.eucalyptussoftware.com/downloads/repo/euca2ools/$VERSION/yum/fedora/
enabled=1
```

```
now install euca2ools

yum install euca2ools.$ARCH --nogpgcheck
```

1. Tarball option

Download the appropriate tarball from <http://www.eucalyptus.com/download/eucalyptus>

Untar the bundle in a temporary location, and install euca2ools

```
tar zxvf euca2ools-$VERSION-*.tar.gz
cd euca2ools-$VERSION-*
sudo -s
yum install -y swig m2crypto
rpm -Uvh euca2ools-$VERSION-*. $ARCH.rpm
```

NOTE: please use '-Uvh' and not '-i'.

Installing Euca2ools on Debian Squeeze

This document shows you how to install Euca2ools from DEB packages on Debian squeeze. In the following steps, the value of \$VERSION must be substituted accordingly (e.g., as 1.2, 1.3.1, etc.). For example, we can set the value of 1.3.1 using bash:

```
export VERSION=1.3.1
```

Download DEBs

There are two options for downloading the DEB packages:

1. Remote repository option

DEB packages are available from our repository. To install them, add our repository to the list of repositories for your system to use. To do so, add somewhere in `/etc/apt/sources.list` file the following line:

```
deb http://eucalyptussoftware.com/downloads/repo/euca2ools/$VERSION/debian squeeze main
```

1. Tarball (local repository) option

Euca2ools DEB packages are also available in a single "tarball." Download the tarball from <http://www.eucalyptus.com/download/eucalyptus>

Next, make sure that dpkg-dev is installed, unpack the tarball, and create the local repository:

```
apt-get install dpkg-dev
tar zxvf euca2ools-$VERSION-squeeze.tar.gz
cd euca2ools-$VERSION-squeeze
dpkg-scanpackages . > Packages
```

Now add the appropriate directory for your architecture to your `sources.list` as root:

For 32-bit:

```
echo deb file://${PWD} ./dists/squeeze/main/binary-i386/ >> /etc/apt/sources.list
apt-get update
```

For 64-bit

```
echo deb file://${PWD} ./dists/squeeze/main/binary-amd64/ >> /etc/apt/sources.list
apt-get update
```

Install DEBs

To install Euca2ools, now run

```
apt-get update
apt-get install euca2ools python-boto
```

You will have to type "Y" if you see a warning like,

```
WARNING: The following packages cannot be authenticated!
...
Install these packages without verification [y/N]? y
```

After installation you may remove the entry from `sources.list` if you don't want to update Eucalyptus packages automatically.

Using Euca2ools Overview

Euca2ools use cryptographic credentials for authentication. Two types of credentials are issued by EC2- and S3-compatible services: x509 certificates and keys. While some commands only require the latter, it is best to always specify both types of credentials. Furthermore, unless the Web services reside on 'localhost', the URLs of the EC2- and S3-compatible service endpoints must also be specified.

The credentials and URLs can be specified via the command line option or by setting environment variables as follows:

Variable	Option	Explanation
EC2_URL	-U or --url [url]	http://host:8773/services/Eucalyptus or http://ec2.amazonaws.com or https://ec2.amazonaws.com:443
S3_URL	-U or --url [url]	http://host:8773/services/Walrus or http://s3.amazonaws.com or https://s3.amazonaws.com:443
EC2_ACCESS_KEY	-a or --access-key [key]	Access Key ID / Query ID
EC2_SECRET_KEY	-s or --secret-key [key]	Secret Access Key / Secret Key
EC2_CERT	-c or --cert [file]	user's PEM-encoded certificate
EC2_PRIVATE_KEY	-k or --privatekey [file]	user's PEM-encoded private key
EUCALYPTUS_CERT	--ec2cert_path [file]	OPTIONAL path to cloud cert

If you are running Euca2ools against Eucalyptus, sourcing the **euca**rc file that is included as part of the credentials zip-file that you downloaded from the Eucalyptus Web interface should be enough to set up all of the above variables correctly.

Commands start with `euca-` and typing `<command name> --help` will print a basic help message. In addition, running `man <command name>` will bring up a man page.

Image Management

In order to use run instances from images that you have created (or downloaded), you need to bundle the images with your cloud credentials, upload them and register them with the cloud. Following examples show how you would perform the necessary steps.

Bundling images

The examples here assume that you have sourced the **euca**rc config file obtained when you downloaded user credentials.

"euca-bundle-image" can be used to bundle an image for use with Eucalyptus or Amazon. A bundled image consists of a manifest file and several image parts.

For instance, to bundle an image "image.img" for user id "123456789111" in the directory "image-dir"

```
euca-bundle-image -i image.img -u 123456789111 -d image-dir
```

OR, if you wish to specify credentials separately ("cert-xyz.pem" and "pk-xyz.pem" are the user certificate and private key PEM files, respectively).

```
euca-bundle-image -i image.img -u 123456789111 -d image-dir -c cert-xyz.pem -k pk-xyz.pem
```

To bundle an image for use with Amazon, make sure you 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`

```
euca-bundle-image -i image.img -u 123456789111 -d image-dir -c cert-abc.pem -k pk-abc.pem --ec2cert $EC2_AMITOOL_HOME/etc/ec2/amitools/cert-ec2.pem
```

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

For more options, type,

```
euca-bundle-image --help
```

or refer to the manpage for "euca-bundle-image."

Uploading an image

To upload an image bundled with "euca-bundle-image" you can use "euca-upload-bundle."

For example, to upload the bundle corresponding to the manifest "image.img.manifest.xml" to the bucket "image-bucket," you would run the following command,

```
euca-upload-bundle -b image-bucket -m image.img.manifest.xml
```

For more options, type

```
euca-upload-bundle --help
```

or refer to the manpage for "euca-upload-bundle."

Registering an image

Bundle images that have been uploaded to the cloud need to be registered with the cloud prior to running instances.

For instance, to register a bundled image referenced by the manifest file "image.img.manifest.xml" that has been uploaded to the bucket "image-bucket" type the following command,

```
euca-register image-bucket/image.img.manifest.xml
```

For more options, refer to the manpage for "euca-register" or type,

```
euca-register --help
```

Downloading an image

Bundled images that have been uploaded may also be downloaded or deleted from the cloud.

For instance, to download the image(s) that have been uploaded to the bucket "image-bucket" you may use the following command,

```
euca-download-bundle -b image-bucket
```

For more options, type,

```
euca-download-bundle --help
```

Deleting a bundled image

To delete a bundled image, use "euca-delete-bundle."

For instance, to delete the images in bucket "image-bucket" you can use the following command,

```
euca-delete-bundle -b image-bucket
```

You can specify a manifest using the "-m" or "--manifest" argument if you wish to delete a specific bundle.

To delete the bucket after deleting the bundled image,

```
euca-delete-bundle -b image-bucket --clear
```

A bucket can only be deleted when it is empty.

For more options, type,

```
euca-delete-bundle --help
```

Unbundling an image

To unbundle a previously bundled image, use "euca-unbundle"

For instance, to unbundle the bundled image referenced by the manifest "image.img.manifest.xml" to the directory image-dir, use the following command,

```
euca-unbundle -m image.img.manifest.xml -d image-dir
```

For more options, try,

```
euca-unbundle --help
```

Networking and Security

You can assign IP address to instances dynamically, unassign addresses, create security groups and assign networking rules to security groups.

Allocating and associating IP addresses

You may use "euca-allocate-address" and "euca-associate-address" to allocate IP addresses and associate public IP addresses with instances, respectively.

In the following example, we will allocate an IP address and associate it with the instance "i-56785678".


```
euca-allocate-address  
ADDRESS    a.b.c.d  
  
euca-associate-address -i i-56785678 a.b.c.d
```

Disassociating and Releasing addresses

You may use "euca-disassociate-address" and "euca-release-address" to disassociate an IP address from an instance and to release the IP address to the global pool. For instance, to release and disassociate the address "a.b.c.d."

```
euca-disassociate-address a.b.c.d  
  
euca-release-address a.b.c.d
```

Creating a security group

You can create a security group using the "euca-add-group" command. For instance, to create a group named "mygroup," you may use the following command,

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

Security groups may be specified when running instances with "euca-run-instances" using the "-g" parameter.

Adding networking rules to security groups

By default, a security group denies incoming network traffic from all sources. You may add networking related rules to security groups using the command "euca-authorize."

To see the entire list of options, type,

```
euca-authorize --help
```

For example, to allow incoming ssh (port 22) traffic to the security group "mygroup" you may use the following command, which specifies a protocol (tcp) a port (22) and a CIDR source network (0.0.0.0/0, which refers to any source):

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

Instead of specifying a CIDR source, you may instead specify another security group to allow access from:

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

Revoking networking rules from security groups

Revocation works the same way as addition (i.e. the command takes the same parameters), except that you should use the "euca-revoke"

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

will list all options.

Deleting a security group

You may use "euca-delete-group" to delete a security group. For example,

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

will delete the security group "mygroup."

Using Block Storage

You can create dynamic block volumes, attach volumes to instances, detach volumes, deletes volumes, create snapshots from volumes and create volumes from snapshots with your cloud. Volumes are raw block devices. You can create a filesystem on top of an attached volume and mount the volume inside a VM instance as a block device. You can also create instantaneous snapshots from volumes and create volumes from snapshots.

Creating a volume

To create a dynamic block volume, use "euca-create-volume."

For instance, to create a volume that is 1GB in size in the availability zone "myzone" you may use the following command,

```
euca-create-volume --size 1 -z myzone
```

To list availability zones, you may use "euca-describe-availability-zones"

You may also create a volume from an existing snapshot. For example, to create a volume from the snapshot "snap-33453345" in the zone "myzone" try the following command,

```
euca-create-volume --snapshot snap-33453345 -z myzone
```

For more options, type,

```
euca-create-volume --help
```

Attaching a volume to an instance

You may attach block volumes to instances using "euca-attach-volume." You will need to specify the local block device name (this will be used inside the instance) and the instance identified. For instance, to attach a volume "vol-33534456" to the instance "i-99838888" at "/dev/sdb" use the following command,

```
euca-attach-volume -i i-99838888 -d /dev/sdb vol-33534456
```

You can attach a volume to only one instance at a given time.

Detaching a volume

To detach a previously attached volume, use "euca-detach-volume." For example, to detach the volume "vol-33534456"

```
euca-detach-volume vol-33534456
```

You must detach a volume before terminating an instance or deleting a volume. If you fail to detach a volume, it may leave the volume in an inconsistent state and you risk losing data.

Delete a volume

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

```
euca-delete-volume vol-33534456
```

You may only delete volumes that are not currently attached to instances.

Creating a snapshot

You may create an instantaneous snapshot of a volume. A volume could be attached and in use during a snapshot operation. For example, to create a snapshot of the volume "vol-33534456" use the following command

```
euca-create-snapshot vol-33534456
```

Deleting a snapshot

To delete a snapshot, use "euca-delete-snapshot." For example, to delete the snapshot snap-33453345, use the following command,

```
euca-delete-snapshot snap-33453345
```

VM Control

A cloud will let users control virtual machine (VM) instances using uploaded images as a template. The following commands can be used to control VM instances.

Displaying instances currently running

You may use "euca-describe-instances," which will display a list of currently running instances.

```
euca-describe-instances
```

To get information about a specific instance, you can use the instance id as an argument to euca-describe-instances. For example,

```
euca-describe-instances i-43035890
```

For more options, type,

```
euca-describe-instances --help
```

Running instances

"euca-run-instances" will allow you to deploy VM instances of images that have been previously uploaded to the cloud.

For instance, to run an instance of the image with id "emi-53444344" with the kernel "eki-34323333" the ramdisk "eri-33344234" and the

keypair "testkey" you can use the following command,

```
euca-run-instances -k testkey --kernel eki-34323333 --ramdisk eri-33344234 emi-53444344
```

To run more than one instances, you may use the "-n" or "--instance-count" option.

For more help, try,

```
euca-run-instances --help
```

or refer to the manpage for "euca-run-instances."

Shutting down instances

You may shutdown running instances using the "euca-terminate-instances" command. For example, to terminate an instance "i-34523332"

```
euca-terminate-instance i-34523332
```

For more options, type,

```
euca-terminate-instances --help
```

or refer to the manpage.

Rebooting instances

To reboot running instances, you can use "euca-reboot-instances." For example, to reboot the instance "i-34523332"

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

A reboot will preserve the root filesystem for the instance across restarts.

Changelog

Version 1.3.1 (2010-09-03)

- Fixed image registration issue #629118.
- CentOS packages fixed and updated to correctly reference the python executable.

Version 1.3 (2010-08-24)

- Windows API support (bundle instance, describe bundle tasks, cancel bundle task, get password data).
- Updates spec file for Fedora packages.
- Updated bash completion rules.
- API refresh to support new command line options (including block device mappings, updates to euca-register).
- Support newer versions of dependencies.
- A number of bug fixes including :

```
#522398 #522396 #522396 #523332 #516738 #525137
#527477 #526697 #530296 #531076 #526591 #536876
#543221 #544706
```

Version 1.2 (2010-02-12)

- Support for bash completion of commands
- A number of bug fixes including:

```
#479823 #450044 #485106 #495618 #439366 #497495
#507840 #510982 #510982 #516486 #516738
```

Version 1.1 (2009-11-05)

- Addresses incompatibilities in some uncommon command line options
- A number of bug fixes including

```
#461301 #450044 #444747 #444105 #444097 #436950
#436947 #404951 #429010 #427367 #424212 #423500
#419583 #417937 #413735 #409732 #407710 #401225
#403244 #402670 #401172 #401222 #401190 #404951
#429010 #427367 #424212 #423500 #419583 #417937
#419583 #413735 #409732 #407710 #401225 #403244
#402670 #401172 #401222 #401190
```

- Addressed incompatibility in several command line options.
- Number of fixes for euca-bundle-vol

- euca-describe-availability-zones accepts "--region"
- Better error reporting for fault strings.
- Incorporated changes to debian rules from Steffen Moeller et al

Version 1.0 (2009-07-17)

License

Software License Agreement (BSD License)
Copyright (c) 2009, Eucalyptus Systems, Inc. All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Interacting with Walrus (2.0)

Walrus is a storage service included with Eucalyptus that is interface compatible with Amazon's S3. Walrus allows users to store persistent data, organized as buckets and objects (see Amazon's S3 Getting Started Guide for more information). Walrus system options can be modified via the administrator web interface.

If you would like to use Walrus to manage Eucalyptus VM images, you can use Amazon's tools to store/register/delete them from Walrus.

Otherwise, you may use other third party tools to interact with Walrus directly.

Third party tools for interacting with Walrus/S3

- s3curl S3 Curl is a command line tool that is a wrapper around curl.
- s3cmd is a tool that allows easy command line access to storage that supports the S3 API.
- s3fs is a tool that allows users to access S3 buckets as local directories.

s3cmd

Author	Michal Ludvig
Homepage	s3cmd
Documentation	http://s3tools.org/s3cmd
Discussion	forum
Download	downloads
Source	subversion
Version	0.9.8.3
License	GPL version 2

Example Usage

Setup s3cmd

Pre-requisites: python and python-openssl.

To use s3cmd with Walrus, download version 0.9.8.3 from the s3cmd sourceforge site and untar it.

Please copy and paste the patch from using the following link (into a file called s3cmd-0.9.8.3.patch).

s3cmd version 0.9.8.3 patch

Change dir to s3cmd-0.9.8.3. Apply the patch (patch -p1 < s3cmd-0.9.8.3.patch).

Configuration

Create a config file for Walrus (called s3cfg.walrus in the examples below). Please use the following sample config as a guideline.

Sample config for Walrus

If you also wish to interact with Amazon's S3, you will need another config file. Please use the following example.

Sample config for S3

Examples

- Create a bucket `s3cmd -c s3cfg.walrus mb s3://testbucket`
- Delete a bucket `s3cmd -c s3cfg.walrus rb s3://testbucket`
- List all buckets `s3cmd -c s3cfg.walrus ls`
- List the contents of a bucket `s3cmd -c s3cfg.walrus ls s3://testbucket`
- Put an object `s3cmd -c s3cfg.walrus put <filename> s3://testbucket`
- Get an object `s3cmd -c s3cfg.walrus get s3://testbucket/<key> <filename>`
- Delete an object `s3cmd -c s3cfg.walrus del s3://testbucket/<key>`

Please use **s3cmd -h** for more options.

s3curl

S3 Curl is a tool that allows users to interact with Walrus by adding security parameters as curl headers.

You may create, delete, list buckets, put, get, delete objects, set access control policies, etc. Please refer to the Amazon S3 documentation for the S3 interface specification.

You will need the perl-Digest-HMAC package for S3 Curl, if it is not already installed on your system.

You will need to modify the file s3curl.pl in your favorite editor to change the hostname endpoint.

For example, change

```
my @endpoints = ( 's3.amazonaws.com' );
```

to

```
my @endpoints = ( 'your-host' );
```

where, *your-host* is the IP or the hostname on which Walrus runs. For authentication to succeed, it is **crucial** that *your-host* is the same as the host portion of the `$S3_URL` environment variable set by `eucarc`.

When modifying the file **s3curl.pl**, please specify just the hostname and not the port number or the entire path (e.g., you would specify **xyz.com** instead of **xyz.com:8773/services/Walrus**. However, you need to use the entire path in the command-line when using the tool).

Be sure to source your 'eucarc' file before running the commands below.

```
s3curl.pl --id $EC2_ACCESS_KEY --key $EC2_SECRET_KEY <curl options>
```

It is preferable to add your credentials to the ".s3curl" config file. Please read the README file bundled with s3curl for details.

- For example, to create a bucket,

```
s3curl.pl --id $EC2_ACCESS_KEY --key $EC2_SECRET_KEY --put /dev/null -- -s -v $S3_URL/bucketName
```

where bucketName is the name of the bucket that you want to create.

- To put an object,

```
s3curl.pl --id $EC2_ACCESS_KEY --key $EC2_SECRET_KEY --put <filename> -- -s -v $S3_URL/bucketName/objectName
```

where objectName is the name of the object that you want to create.

- To get the MD5 checksum, size and last modification time for an object.

```
s3curl.pl --id $EC2_ACCESS_KEY --key $EC2_SECRET_KEY --head -- -s -v $S3_URL/bucketName/objectName > object.head
cat object.head
```

- To get an object

```
s3curl.pl --id $EC2_ACCESS_KEY --key $EC2_SECRET_KEY --get -- -s -v $S3_URL/bucketName/objectName > object
```

- To delete an object,

```
s3curl.pl --id $EC2_ACCESS_KEY --key $EC2_SECRET_KEY --del -- -s -v $S3_URL/bucketName/objectName
```

- To delete a bucket,

```
s3curl.pl --id $EC2_ACCESS_KEY --key $EC2_SECRET_KEY --del -- -s -v $S3_URL/bucketName
```

Note that, according to the S3 specification, a bucket needs to be empty before it can be deleted.

You may pipe the output through "xmllindent" (you might need to install it, if it is not already installed on your system).

- For example, to get the access control policy for a bucket,

```
s3curl.pl --id $EC2_ACCESS_KEY --key $EC2_SECRET_KEY --get -- -s -v $S3_URL/bucketName?acl | xmllindent > bucket.acl
```

- To get the contents of a bucket.

```
s3curl.pl --id $EC2_ACCESS_KEY --key $EC2_SECRET_KEY --get -- -s -v $S3_URL/bucketName | xmllindent > bucket.list
```

- To list all buckets for a user.

```
s3curl.pl --id $EC2_ACCESS_KEY --key $EC2_SECRET_KEY --get -- -s -v $S3_URL | xmllindent > bucketlisting
```

s3fs

Author	rrizun
Homepage	s3fs
Documentation	wiki
Discussion	group
Download	download
Source	svn info
Version	r177
License	GNU General Public License v2

Installation

- Download: s3fs-r177-source.tar.gz.
- Save the following patch to a file called s3fs-r177-patch: s3fs-r177 patch
- Untar s3fs-r177-source.tar.gz and patch it

```
patch -p1 < s3fs-r177-patch
```

Compiling

Pre-requisites: **libfuse-dev libcurl4-openssl-dev libxml2-dev**

Run **make** (you might get a bunch of warnings).

Usage

```
mkdir test
./s3fs <bucket name> -o accessKeyId=<query id> -o secretAccessKey=<secret key> -o servicepath=/services/Walrus -o host=http://localhost:8773 test
```

where <bucket name> is an existing bucket name.

```
cd test; touch foo; ls ; rm foo, etc.
```

Kill the s3fs daemon to unmount the bucket.

Debugging

Add "-d" to the end of the s3fs command-line. This will produce debugging output on the command line and will not daemonize s3fs.

Look for errors in cloud-error.log and/or cloud-debug.log.

Interacting with Block Storage (2.0)

The Block Storage Service in Eucalyptus is interface-compatible with Amazon's Elastic Block Store. You can therefore use either EC2 commands or euca2ools commands to control it.

The instructions below rely on the euca2ools command-line tools distributed by the Eucalyptus Team. Please, install them if you haven't done so already.

The following operations are possible,

1. Creating volumes

You may create a volume either from scratch or from an existing snapshot.

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

where <size> is the size in GB and <zone> is the availability zones you wish to create the volume in (use euca-describe-availability-zones to discover zones).

For instance,

```
euca-create-volume --size 1 --zone myzone
```

will create a 1GB volume in the availability zone "myzone"

To create a volume from a snapshot,

```
euca-create-volume --snapshot <snapshot id> --zone <zone>
```

where <snapshot id> is the unique identifier for a snapshot and <zone> is the availability zone you wish to create the volume in.

For instance,

```
euca-create-volume --snapshot --zone myzone snap-EF4323
```

will create a volume from the snapshot "snap-EF4323" in the zone "myzone"

2. Query the status of volumes

```
euca-describe-volumes
```

Volumes marked "available" are ready for use.

3. Attaching a volume

You can attach volumes to existing instances (that have been started with euca-run-instances). You may attach a volume to only one instance at a time.

```
euca-attach-volume -i <instance id> -d <local device name> <volume id>
```

where <volume id> is the unique identifier for a volume (vol-XXXX), <instance id> is a unique instance identifier and <local device name> is the name of the local device in the guest VM.

For instance,

```
euca-attach-volume -i i-345678 -d /dev/sdb vol-FG6578
```

will attach the previously unattached volume "vol-FG6578" to instance "i-345678" with the local device name "/dev/sdb"

4. Detaching a volume

```
euca-detach-volume <volume id>
```

where <volume id> is the unique identifier for a previously attached volume (vol-XXXX).

For instance,

```
euca-detach-volume vol-FG6578
```

will detach volume "vol-FG6578"

Important! The user of the instance is responsible for making sure that the block device is unmounted before a detach. Detach cannot

ensure the consistency of user data if the user detaches a volume that is in use.

5. Deleting a volume

```
euca-delete-volume <volume id>
```

where <volume id> is the unique identifier for a volume (vol-XXXX).

6. Creating a snapshot from a volume

You can snapshot a volume so that you can create volumes in the future from the snapshot.

```
euca-create-snapshot <volume id>
```

where <volume id> is the unique identifier for a volume (vol-XXXX).

For instance,

```
euca-create-snapshot vol-GH4342
```

will snapshot the volume "vol-GH4342"

The volume to be snapshot needs to be "available" or "in-use." You cannot snapshot a volume that is in the "creating" state.

7. Querying the status of snapshots

```
euca-describe-snapshots
```

You may create volumes from snapshots that are marked "completed."

8. Deleting a snapshot

```
euca-delete-snapshot <snapshot id>
```

where <snapshot id> is the unique identifier for a snapshot.