Eucalyptus (1.2 and 1.3)

Supported ¶

- Installation
  - Rocks-based "almost-one-button" binary install
  - Experts-only "you-are-on-your-own" source install
  - RPM packages for "non-Rocks" RPM based systems
- Administration
  - Adding/approving/disabling/deleting users (via the Web interface)
  - Adding/listing/disabling images (Web interface with command line use)
  - Adding/deleting nodes and clusters (via edit of configuration files)
- Amazon's EC2 compatibility:
  - In terms of command-line tools:
    - ec2-describe-images (currently lists admin-provided images)
    - ec2-add-keypair (key pair generation & upload) and ec2-describe-keypair
    - ec2-run-instances and ec2-terminate-instances
    - ec2-describe-instances
    - ec2-availability-zones

Unsupported ¶

- Administration
  - Easy image deletion
- Amazon's EC2 incompatibilities:
  - User-selectable images (bundle & upload)
  - User-selectable kernels
  - Elastic IP addresses
  - In terms of command-line tools:
    - ec2-authorize (firewall control)
    - ec2-bundle-vol (image creation)
    - ec2-register and ec2-deregister (image registration with EC2)
    - ec2-upload-bundle & ec2-delete-bundle (image upload/removal to S3)

ChangeLog

Version 1.3 (2008-08-27) ¶

- Added support for the new ec2 tools (1.3-24159 and newer)

Version 1.2 (2008-07-29) ¶

- Added stand-alone RPM packages for non-rocks installation
- Added image caching to reduce instance creation time
- Added instance networking configuration options to eucalyptus.conf
- Bug Fixes
  - Improved installation-time error checking
  - Removed possibility of instance ID collision
  - Improved VDE runtime management
  - Improved VDE cleanup
  - Resolved occasional NC instance loss problem
  - Resolved EC2 client timing issue that resulted in parsing errors on client

Version 1.1 (2008-07-01) ¶

- Added WS-security for internal communication
- Added URL Query Interface for interacting with Eucalyptus
- Cluster Controller improvements:
  - Instance caching added to improve performance under certain conditions
  - Thread locks removed to improve performance
  - NC resource information gathered asynchronously to improve scheduler performance
- Network control improvements:
  - Added ability to configure 'public' instance interface network parameters (instead of hardcoded 10. network)
  - Lots of reliability changes
- Cloud Controller improvements:
  - Pure in-memory database
  - Image registration over WS interface
  - Improved build procedure
- Web interface improvements:
  - For all users (query interface credentials, listing of available images)
  - For the administrator (addition, approval, disabling, and deletion of users; disabling of images)
- Numerous bug fixes, improving stability and performance. In particular, but not limited to:
  - Recovering Cloud Controller system state
  - Timeout-related error reporting
  - Slimmer log files, with timestamps

Version 1.0 (2008-05-29) ¶

- First public version (limited-feature binary-only beta)

Known Bugs

Some users have run into problems while installing Rocks or Eucalyptus

- Solution: We are working on making Eucalyptus installation more robust and easier to diagnose. Please, consult Rocks support resources for help with Rocks installation.
- Instances with identical IDs may appear in different reservations
  - Solution: Will be fixed in v1.2.
- VDE creates many temporary subdirectories and files
  - Solution: Delete the files manually. Will be fixed in v1.2.
- Eucalyptus loses track of instances that are terminated before they fully booted.
  - Solution: Restart the node controller on the compute nodes with runaway instances. The problem will be fixed in v1.2.
- Parsing of large config files (with many nodes) fails.
  - Solution: Will be fixed in v1.2.
- Startup of large instances takes long
  - Solution: The problem is alleviated in v1.2 with introduction of disk image caching
- Some instances may appear to be stuck indefinitely in a non-running state, such as "pending" or "terminated".
  - Solution: Execute (as administrator or as the owner of the instance) `ec2-terminate-instance` on it.
- Rarely, eucalyptus processes may be left running even after the init script has finished stopping the system (eucalyptus stop).
  - Solution: Search for eucalyptus processes (any process with the string 'euca' contained in the command line) and kill them manually (e.g., `pkill -9 -f euca`).
- Eucalyptus log-files can grow large (we estimate several hundred megabytes per week).
  - Solution: Stop eucalyptus, remove or back up the log files (/opt/eucalyptus-1.1/var/log/eucalyptus/*), and restart eucalyptus.
- Rarely, a 'vdekey' is created that causes the vde_cryptcab listener process to fail, which will prevent private network instance connectivity from functioning properly.
  - Solution: Stop eucalyptus (on all front-end and node systems), regenerate the vdekey on the front-end using the command (ssh-keygen -t rsa1 -b 768

-f /opt/eucalyptus-1.1/var/eucalyptus/keys/vdekey -N ""), restart eucalyptus on the front-end, and finally restart eucalyptus on the nodes.

- Deploying Eucalyptus on a single machine does not work.
  - Solution: Select a different port for NC and CC. Will be fixed in v1.2.
- ec2 command-line tools fail with "Server: An error was discovered processing the <wsse:Security> header. (WSSecurityEngine: Invalid timestamp The security semantics of message have expired)"
  - Solution: Ensure that the clocks on the client and server machines are synchronized. This is not a Eucalyptus bug, but a consequence of the security policy enforced by the ec2 command-line tools.

Frequently Asked Questions

Is Eucalyptus a precise implementation of Amazon's EC2? ¶

No. Eucalyptus supports Amazon's interface syntactically and it implements the same functionality (with a few exceptions), but internally it is almost certainly different. Eucalyptus is designed to be extensible and easy to install and maintain, particularly in a research environment, where system administrator time is the most expensive commodity. While we can't be certain, Amazon's main design goal almost has to be scalability. Put another way, if we were to design a commercial software venue for cloud services where scalability is paramount and we could mandate how all clusters within the cloud were initially configured (instead of an open-source software tool for community distribution) we would have designed Eucalyptus differently.

Why choose Amazon's EC2 interface for Eucalyptus? ¶

The intention is to be able to support multiple cloud computing interfaces using the same "back end" infrastructure. EC2 seemed to be the best documented of the available choices at the time we began development and also the most commercially successful so we chose to implement it first. The interface module, however, can be replaced without changing the rest of the system (we hope).

Is Eucalyptus the product of a company? ¶

No. Eucalyptus has been developed in the MAYHEM Lab within the Computer Science Department at the University of California, Santa Barbara primarily as a tool for cloud-computing research. It is distributed as open source with a FreeBSD-style license that does not restrict its usage much. We hope that it fosters further interest and development, both scientific and commercial, in cloud computing.

Do Eucalyptus users need a credit card to access the system? ¶

No. In a commercial cloud, clients pay for their use with a credit card. Eucalyptus is designed to work in an environment where machines are available to a user community that accesses them via logins. Because user accountability usually must be ensured by the system administrators in such an environment, we have developed a "cloud administrator" interface for Eucalyptus that is more analogous to common practice in a setting without fees.

Cloud users must register with the cloud administrator using a Web-based sign-up page. The cloud administrator receives an email message requesting approval whenever a potential user requests access. The message contains links for approving or denying the request. The decision of the administrator is communicated to the requester via email. In the case of approval, the user will be able to obtain from a password-protected Web page the cryptographic certificates necessary for using Eucalyptus.

What software environments are supported? ¶

Eucalyptus targets Linux systems that use Xen (versions 3.*) for virtualization. It is packaged for Rocks V as a "roll" -- an ISO disk image with RPMs and some metadata that Rocks uses for cluster-wide installation. Additionally, we provide binary RPMS for i386 and x86_64 RPM-based systems, and now offer a source package as well for installation on other common Linux distributions.

What happens when all virtual machine "slots" are allocated? ¶

The SLAs (Service Level Agreements) implemented by the cloud controller are bare-bones simple. Our intention is for Eucalyptus sites and development teams to come up with their own (we have a few in mind we'd like to try) but in the current release, the simple SLAs supplied enforce no restrictions on occupancy duration. This type of SLA is comparable to the Amazon SLA with the exception that Amazon's will "time out" when charges to the registered credit card are no longer postable.

We have implemented no such time out analogy at present, so if users or the cloud administrator do not terminate running instances, eventually Eucalyptus will run out of VM slots to allocate. In this case, user requests fail and the EC2 tools report no instances available when the command "ec2-describe-availability-zones" is invoked.

What is an "Availability Zone" in Eucalyptus? ¶

Amazon implements "availability zones" to allow users some degree of control over instance placement. Specifically, EC2 users can choose to host images in different availability zones if they wish to try and ensure independent failure probabilities. Amazon, presumably, takes steps to insulate instances in separate availability zones from correlated failure (e.g., a single power outage that takes out a data center).

Under Eucalyptus, the abstraction is slightly different. Each availability zone

corresponds to a separate cluster within the Eucalyptus cloud. The advantage is that the networking within a single availability zone can be made much faster (i.e., it uses the cluster's private network in native mode). For allocations that span clusters, the technology Eucalyptus uses to implement a private network for each allocation imposes a substantial performance penalty.

Thus the two are similar in that cloud allocations to separate availability zones do reduce the chance of correlated failure. They are different, however, in that under Eucalyptus, each availability zone is restricted to a single "machine" (e.g., cluster) where at Amazon, the zones are much broader.

Can I help develop Eucalyptus? ¶

For the moment, we are restricting external development contributions for Eucalyptus internals to bug fixes. It is just too complicated to try and keep the code base stable with external developers when we are in this early phase of development. But we will gladly accept patches that fix bugs.

Furthermore, any tools you'd like to develop that use Eucalyptus without modifying it are welcome, as well. We will create a page with a short description of your project and a link to it.

Eucalyptus Installation Quick Start (1.2 and 1.3)

Assuming you have

- a cluster with Rocks V (including **xen** and **java** rolls),
- a few machines that you are willing to dedicate to Eucalyptus, and
- familiarity with Rocks commands,

installation of Eucalyptus, proceeds roughly as follows:

1. Add Eucalyptus to Rocks ¶

Download Eucalyptus rolls here: we have both 32-bit and 64-bit versions. Then:

```
# rocks add roll clean=1 /path/to/eucalyptus-5.0-1.*.disk1.iso
# rocks enable roll eucalyptus
# cd /home/install && rocks-dist dist
```

For more information try section 2.2 of Administrator's Guide

2. Install Eucalyptus on nodes ¶

- **OPTION A:** Add new nodes (choose 'VM Container' and wait for nodes to pop up):

```
# insert-ethers
```

- **OPTION B:** Re-target existing nodes:

```
# rocks set host pxeboot vm-container-0-0 action=install
# rocks set host pxeboot vm-container-0-1 action=install
...
# ssh vc0-0 reboot
# ssh vc0-1 reboot
...
```

For more information try section 2.3 of Administrator's Guide

3. Install Eucalyptus on the front-end ¶

Once the nodes have rebooted, do:

```
# kroll eucalyptus > build.sh
# sh ./build.sh
```

For more information try section 2.4 of Administrator's Guide

4. Configure Eucalyptus ¶

Install at least one runnable VM image (you can download the following test image here) that users can select:

```
$ tar zxvf euca-ttylinux.tgz
$ cd ttylinux
# /opt/eucalyptus-1.1/usr/sbin/euca add_image \
  --disk-image ttylinux.img \
  --kernel-image vmlinuz-2.6.16.33-xen \
  --image-name ttylinux
```

Last, but not least, log in to the Eucalyptus Web-site (login: **admin**, password: **admin**) to set the administrator's email:

https://your.front.end.hostname:8443/

That is it!

For more information try section 2.5 of Administrator's Guide

What next? ¶

At this point you'd probably want to see if it works. As an administrator, you can use the system exactly as all other users would. So, go ahead and download your certificates and use them with EC2 command-line tools as described in the User's Guide.

If it works for you, tell others to sign up for an account using the "Apply" link on the front page of the Web site. When they do so, you will receive email messages asking you to either accept or reject the request. More about this in section 4 of Administrator's Guide.

Eucalyptus Administrator's Guide (1.2 and 1.3)

Eucalyptus can be installed from source, RPM, or a ROCKS "roll". The current version of Eucalyptus is **1.3**. If you wish to patch an existing Eucalyptus **1.2** installation to restore compatibility with the new EC2 tools without upgrading entirely to **1.3**, see the patching page. For a clean install or to upgrade entirely to **1.3**, see the following instructions:

ROCKS	RPM	Source
Install	Install	Install
	Configure	Setup
Deploy		
Manage		

Eucalyptus User's Guide

This guide is meant for people interested in using an existing installation of Eucalyptus. (If you have a cluster that you would like to install Eucalyptus on, then take a look at the Administrator's Guide first.) These instructions apply to versions 1.1, 1.2, and 1.3 of Eucalyptus.

We will guide you through getting access to a Eucalyptus-based cloud, as well as installing and using tools for controlling virtual instances. Those familiar with Amazon's EC2 system will find most of these instructions familiar because Eucalyptus

can be used with EC2's command-line tools.

1. Sign up ¶

If you are using the Eucalyptus Public Cloud, use mayhem9.cs.ucsb.edu 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.)

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.

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.

2. Generate keys ¶

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.

```
unzip name-of-the-key-zip.zip
mkdir ~/.euca
mv euca2-*.pem ~/.euca
chmod 0700 ~/.euca
chmod 0600 ~/.euca/*.pem
```

3. Install EC2 command-line tools ¶

Eucalyptus v1.2 and lower is not compatible with the more recent EC2 command-line tools (1.3-24159 and up). See the discussion board for more information. The solution is either to use the previous version of EC2 tools (1.3-19403) or to upgrade Eucalyptus to v1.3.

Download the EC2 command-line tools from Amazon.

```
wget http://s3.amazonaws.com/ec2-downloads/ec2-api-tools-1.3-24159.zip
unzip ec2-api-tools.zip
```

If you are using the Eucalyptus Public Cloud, use mayhem9.cs.ucsb.edu instead of **your.cloud.server**.

Set the following environment variables (note that the base URL is different from the Web site's URL, both in terms of transport protocol - http instead of https - and the port number):

```
export EC2_HOME=/path/to/installed/ec2-commandline-tools
export PATH=$PATH:$EC2_HOME/bin
export EC2_PRIVATE_KEY=~/.euca/euca2-*-pk.pem
export EC2_CERT=~/.euca/euca2-*-cert.pem
export EC2_URL=http://your.cloud.server:8773/services/Eucalyptus
```

Now you *should* be ready to start using the tools. To test if the tools work (and if the cloud server is running properly), execute the following EC2 command:

If you get an **Invalid timestamp** error when running any of the ec2 commands, make sure the clock on your client machine (and the server, if you are in charge of it) is accurate.

```
ec2-describe-availability-zones
```

In the output of the above command, you should see the cluster's front-end hostname displayed along with the status of 'UP' and a short description of how many 'small' instance types the cloud can execute (002/002 means 2 available out of 2 total). If the response is different or if the number of available instances is 000/000, then contact your cloud administrator about the problem.

4. Quick Start ¶

Now you can begin running VM instances on the Eucalyptus cloud. Using the EC2

command-line tools, you can learn about installed images, start VM instances using those images, describe the running instances, and terminate them when you're finished with them.

The following EC2 commands will allow you to query the system:

```
ec2-describe-images
IMAGE <emi-id> ...

ec2-describe-instances
(will be empty until you start an instance, as shown below)

ec2-describe-availability-zones

ec2-describe-keypairs
(will be empty until you add key pairs, as shown below)
```

Before starting a VM, you need to create at least one key pair. This key pair will be injected into the VM, allowing you to SSH into the instance. Below we will use *mykey* as a handle, but you may choose any string you like instead:

```
ec2-add-keypair mykey
KEYPAIR keyname ...
-----BEGIN RSA PRIVATE KEY-----
<key contents>
-----END RSA PRIVATE KEY-----

cat >mykey.private
(paste the key contents, including BEGIN and END lines, printed out by the previous command)

chmod 0600 mykey.private

ec2-run-instances <emi-id> -k mykey -n <number of instances to start>

ec2-describe-instances
(should now show the instance)
```

Once the instance is shown as 'Running', it will also show two IP addresses assigned to it. You may log into it with the SSH key that you created:

```
ssh -i mykey.private root@one-of-the-ip-addresses
```

To terminate instances, use:

```
ec2-terminate-instances <instance-id1> <instance-id2> ... <instance-idn>
```

Please, see Amazon's EC2 Getting Started Guide for more information about these command-line tools. Keep in mind that not all tools are currently supported.