Eucalyptus (1.5.1)

Eucalyptus version 1.5 incorporates several new features and improvements. Following is a summary of major changes from version 1.4:

- Elastic Block Store (EBS) compatible storage service
- Walrus improvements:
  - Support for groups in ACLS
  - Fixed issues with meta data support and key names
  - Web browser form-based uploads via HTTP POST
  - Object copying
  - Query string authentication
  - Compressed image downloads and fixes to image caching
  - Reduced memory requirement
- Network improvement: new `MANAGED-NOVLAN` mode
- Node-side improvements:
  - Support for the KVM hypervisor
  - Compression and better failure handling on image downloads
  - Reworked caching (now with configurable limit)
- Web UI improvements:
  - Cloud registration with Rightscale (from admin's 'Credentials' tab)
  - New configuration options for Walrus
  - Better screening of usernames
  - Fixed account confirmation glitches
- Building and installation improvements
  - Better Java installation checking
  - New command-line administration: `euca_conf -addcluster ... -addnode ...`
  - Non-root user deployment of Eucalyptus
  - Binary packages for more distributions: Ubuntu, Debian, openSUSE and CentOS

For a more detailed list, see the 1.5 Changelog:

> http://open.eucalyptus.com/wiki/ChangeLog_v1.5

For a Eucalyptus compatibility matrix of supported Amazon features:

> http://open.eucalyptus.com/wiki/API_v1.5

Eucalyptus Administrator's Guide (1.5.1)

This guide is meant for people interested in installing Eucalyptus on their resources: anything from a laptop to a set of clusters (If you are trying to use an existing Eucalyptus installation, you may be more interested in the User's Guide).

1. Prerequisites
2. Installation Overview
   1. From source
   2. From distribution-specific binary packages
      - CentOS 5.3
      - OpenSUSE 11
      - Debian Lenny 5.0
      - Debian Squeeze/sid
      - Ubuntu Jaunty 9.04
   3. On a Rocks cluster
   4. As an upgrade from 1.3 or 1.4 (source or RPM)
3. Configuration
   - System configuration
   - Network configuration
4. Management
   - Managing Images
   - Managing Eucalyptus
5. Troubleshooting

Eucalyptus Prerequisites (1.5.1)

What follows is a comprehensive list of dependencies that must be satisfied before building Eucalyptus or running it. Later sections of this Administrator's Gude *may* have detailed instructions on how to satisfy these dependencies on specific distributions.

1. For compiling from source ¶

- C compilers
- Java Developer Kit (SDK) version 1.6 or above
- Apache ant 1.6.5 or above
- libc development files
- pthreads development files
- libvirt development files
- Axis2C and rampart development files (included with Eucalyptus)
- Curl development files
- openssl development files
- Optional: zlib development files

2. For running Eucalyptus ¶

There are a few different Eucalyptus components that run on either a cluster 'front-end', or on a cluster 'node'. There are different run-time dependencies for 'front-end' and 'node' components. One physical machine can play the role of the front-end and the node.

Front-end run-time dependencies

- **Java 6** is needed by the Eucalyptus components running on the front end. Note that GNU Compiler for Java (gcj), included by default with some Linux distributions, is **not** sufficient. Make sure that your JAVA_HOME environment variable is set to the location of your JDK.
- **Apache ant** is needed to run the Cloud Controller.
- **Perl** is used by helper scripts
- The head node must run a **server on port 25** that can deliver or relay email messages to cloud users' email addresses. This can be Sendmail, Exim, or postfix, or even something simpler, given that this server does not have to be able to receive incoming mail. Many Linux distributions satisfy this requirement out of the box. To test whether you have a properly functioning mail relay for localhost, try to send email to yourself from the terminal using "mail".
- Dependencies for network support differ depending on the mode used (see Eucalyptus Networking for details). For full functionality satisfy all of them:
  - For all modes:
    - `iproute` and `iptables` packages (`ip` and `iptables` commands must work)
  - For all modes except SYSTEM:
    - DHCP Server compatible with ISC DHCP Daemon version 3.0.X (dhcp3-server)
  - For MANAGED and MANAGED-NOVLAN modes:
    - `bridge-utils` package (`brctl` command must work)
  - Additionally, for MANAGED mode:
    - `vlan` package (`vconfig` command must work)
- For persistent dynamic block storage (aka EBS) to work, the front end will need to have the following software packages installed:
  - `lvm2` package (e.g., command `lvm` should work)
  - `aoetools` package. The `aoe` module needs to be loaded on the front end as well as all nodes (`modprobe aoe`). If your kernel does not have ATA-over-Ethernet support, you will have to add that.
  - `vblade` package

Node run-time dependencies

- **Perl** scripts are invoked by the Node Controller
- Two hypervisors are supported:
  1. **Xen** (version >= 3.0.x)
     - Furthermore, `xen-utils` package is needed (`xm` command must work)
  2. **KVM**
- Dependencies for network support differ depending on the mode used (see Eucalyptus Networking for details). For full functionality satisfy all of them:
  - For all modes:
    - `iproute` and `iptables` packages (`ip` and `iptables` commands must work)
  - For MANAGED and MANAGED-NOVLAN modes:
    - `bridge-utils` package (`brctl` command must work)
  - Additionally, for MANAGED mode:
    - `vlan` package (`vconfig` command must work)
- `libvirt` package (potentially with `libvirtd`, depending on hypervisor configuration)

All Eucalyptus components

- You *must* be **root** to install and start Eucalyptus components (they may be configured to run as a different user on most distributions). This document assumes that all commands will be executed as root.

Attention Rocks users: Eucalyptus 1.5.1 can be installed on a Rocks-based cluster of version 5 or higher. To satisfy the prerequisites, please, install Java on the front-end and the **xen** roll in each of your virtual machine containers. The JDK installed by the **java** roll of the current version of Rocks is unfortunately insufficient, so you will need to install JDK 1.6.0 "manually". For our testing we used Sun's JDK, which can be found at http://java.sun.com/javase/downloads/index.jsp.

3. Distribution specific examples ¶

For **Opensuse 11.1**, run the following command to install all required dependency packages:

```
yast2 -i bzr python-paramiko make gcc ant apache2 apache2-devel java-1_6_0-openjdk java-1_6_0-openjdk-devel libvirt-devel libcurl-devel vlan dhcp-server bridge-utils ant-contrib ant-nodeps curl libvirt
```

For **Ubuntu 9.04**, run the following command to install all required dependency packages:

```
apt-get install bzr gcc make apache2-threaded-dev ant openjdk-6-jdk libvirt-dev libcurl4-dev dhcp3-server vblade apache2 ruby unzip libopenssl-ruby curl vlan bridge-utils libvirt-bin kvm libcurl3
```

For **CentOS 5.3**, run the following command to install all required dependency packages:

```
yum install -y java-1.6.0-openjdk-devel ant ant-nodeps libvirt-devel curl-devel httpd httpd-devel apr-devel openssl-devel dhcp
```

4. For interacting with Eucalyptus ¶

To interact with Eucalyptus, you need to install functioning EC2 command-line tools from Amazon. The latest version of these tools that we support are ec2-api-tools-1.3-30349 and ec2-ami-tools-1.3-26357. Note that the tools have their own prerequisites (Java for API tools and Ruby for AMI tools).

Installing Eucalyptus (1.5.1)

Before you proceed with the installation, be sure to take a look at the list of Eucalyptus' prerequisites.

A Eucalyptus cloud setup consists of three components -- the cloud controller, the cluster controller(s) and node controller(s). The cloud controller is a Java program that, in addition to high-level resource scheduling and system accounting, offers a Web services interface and a Web interface to the outside world. Cluster controller and node controller are written in C and deployed as Web services inside Apache.

Communication among these three types of components goes over SOAP with WS-security. There is one cluster controller per cluster, running on the head node; there is one node controller per each compute node. So, if you are installing Eucalyptus on one cluster, then one cloud and one cluster controller should be deployed on the head node and one node controller should be deployed on each compute node.

If you are upgrading from a previous version of Eucalyptus, please follow the instructions in the Upgrade Document.

Eucalyptus can be installed from source or using a set of packages (RPM and DEB). The former method is more general and should work on practically any Linux system, the latter should work on distribution which we support (at the moment Ubuntu 9.04, Debian squeeze/lenny, CentOS 5.3 and openSUSE 11). Furthermore, we have special advice for those using Rocks-based clusters.

If run into any problems, be sure to check the troubleshooting guide for solutions to commonly encountered problems.

Installing Eucalyptus from source (1.5.1)

NOTE - If you are upgrading from a Eucalyptus 1.4 or older installation, please consult the Upgrade Documentation for instructions that will explain how to preserve user account information and re-import the images.

1. Download Eucalyptus ¶

Download either

- eucalyptus-1.5.1-src.tar.gz (Eucalyptus source with included java libraries)

or

- eucalyptus-1.5.1-src-online.tar.gz (Eucalyptus source that will download java libraries at build-time)

and for both

- eucalyptus-1.5.1-src-deps.tar.gz (Eucalyptus C library dependency packages)

All packages can be found on the Eucalyptus Web site:

- http://open.eucalyptus.com/downloads

Unpack the Eucalyptus source:

```
tar zvxf eucalyptus-1.5.1-src.tar.gz
```

Now you should have a directory eucalyptus-1.5.1. To simplify the remainder of the installation, define EUCALYPTUS_SRC environment variable to be the top of the source tree of eucalyptus and the variable EUCALYPTUS to be the directory where eucalyptus will be installed (we recommend using `/opt/eucalyptus/`):

```
cd eucalyptus-1.5.1
export EUCALYPTUS_SRC=`pwd`
export EUCALYPTUS=/opt/eucalyptus
```

2. Dependencies ¶

To install Eucalyptus, you need to build packages that Eucalyptus depends on, which we provide in the above-mentioned package eucalyptus-1.5.1-src-deps.tar.gz. For the sake of this discussion, we are going to assume that all packages have been untarred inside "$EUCALYPTUS_SRC/eucalyptus-src-deps/" as above and will be installed in "$EUCALYPTUS/packages".

Unpack the dependencies and create the directory you'll use to install them:

```
cd $EUCALYPTUS_SRC
tar zvxf ../eucalyptus-1.5.1-src-deps.tar.gz
mkdir -p $EUCALYPTUS/packages/
```

Build and install the dependencies. The following instructions work on some Linux distributions, but aren't universal. *Please, consult the documentation for the specific packages for help with building them on your distribution.*

a. Axis2 ¶

```
cd $EUCALYPTUS/packages
tar zxvf $EUCALYPTUS_SRC/eucalyptus-src-deps/axis2-1.4.tgz
```

b. Axis2/C ¶

```
export AXIS2C_HOME=$EUCALYPTUS/packages/axis2c-1.5.0
cd $EUCALYPTUS_SRC/eucalyptus-src-deps/
tar zvxf axis2c-src-1.5.0.tar.gz
cd axis2c-src-1.5.0
CFLAGS="-w" ./configure --prefix=${AXIS2C_HOME} --with-apache2=/usr/include/apache2 --with-apr=/usr/include/apr-1.0
make ; make install
```

c. Rampart/C ¶

```
export AXIS2C_HOME=$EUCALYPTUS/packages/axis2c-1.5.0
export LD_LIBRARY_PATH=${AXIS2C_HOME}/lib:$LD_LIBRARY_PATH
cd $EUCALYPTUS_SRC/eucalyptus-src-deps/
tar zvxf rampartc-src-1.2.0.tar.gz
cd rampartc-src-1.2.0
./configure --prefix=${AXIS2C_HOME} --enable-static=no --with-axis2=${AXIS2C_HOME}/include/axis2-1.5.0
make ; make install
```

Now edit the file $AXIS2C_HOME/axis2.xml: search for "Security" and change

```
  <!--phase name="Security"/-->
```

to

```
  <phase name="Security"/>
```

and save the file.

d. Other software ¶

Please, consult the prerequisites page for additional software, not included by us, that is required for building Eucalyptus.

Furthermore, you will need functioning EC2 command-line tools from Amazon. The latest version of these tools that we support are ec2-api-tools-1.3-30349 and ec2-ami-tools-1.3-26357.

3. Building Eucalyptus ¶

```
cd $EUCALYPTUS_SRC
./configure --with-axis2=$EUCALYPTUS/packages/axis2-1.4 --with-axis2c=$EUCALYPTUS/packages/axis2c-1.5.0 --enable-debug --prefix=$EUCALYPTUS
cd clc/; make deps; cd ..
make ; make install
```

4. Deploying Eucalyptus to multiple machines ¶

To configure Eucalyptus you need to specify where Eucalyptus is installed. Moreover, for security reasons you need to specify the Unix user that Eucalyptus's services will run as. We suggest using `eucalyptus` as such user.

These and other configuration options are stored in the file called `$EUCALYPTUS/etc/eucalyptus/eucalytpus.conf` on all nodes. You may edit it manually or you may use the `euca_conf` script that we provide. For instance, the minimal required configuration that should be the same on all nodes can be recorded as follows:

```
$EUCALYPTUS/usr/sbin/euca_conf -d $EUCALYPTUS -user eucalyptus $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf
```

At this point, if you plan to use Eucalyptus on more than one node, you're ready to push the software out to the other nodes. If you installed Eucalyptus in its own directory, you can just sync the entire package to all of the hosts listed above using whatever mechanism you typically use to push changes to nodes (rsync, for instance)

```
rsync -a $EUCALYPTUS/ hostname1:$EUCALYPTUS/
rsync -a $EUCALYPTUS/ hostname2:$EUCALYPTUS/
...
```

This would also be a good time to ensure that all of your nodes have the Unix user for running Eucalyptus (e.g., `eucalyptus` is in `/etc/passwd` on all nodes).

Installing Eucalyptus (1.5.1) on CentOS 5.3

Eucalyptus can be installed on CentOS 5.3 using binary RPM packages.

Download RPMs ¶

WARNING: Uninstalling an old RPM package (1.4 and earlier) will completely remove the $EUCALYPTUS directory, thus removing all the uploaded buckets and registered users! If you want to keep them you should **upgrade** instead, as explained below.

Eucalyptus binary installation is broken up into several packages: the cloud controller (-cloud package), the cluster controller (-cc package), and the node controller (-nc package). To simplify installation, everything is bundled into a single "tarball" for convenience, wherein we include copies of third-party packages that Eucalyptus depends on (Rampart, Axis2C, many Java libraries).

Download the appropriate tarball from http://open.eucalyptus.com/downloads

- For 32-bit machines, get eucalyptus-1.5.1-centos-i386.tar.gz
- For 64-bit machines, get eucalyptus-1.5.1-centos-x86_64.tar.gz

Untar the bundle in a temporary location:

```
tar zxvf eucalyptus-1.5.1-*.tar.gz
cd eucalyptus-1.5.1-*
```

All others dependencies are present in the standard repositories.

Prerequisites ¶

If you start with a standard CentOS installation, you will satisfy all Eucalyptus prerequsites with the following steps:

1. Front-end, node and client machine system clocks are syncronized (i.e., using NTP).
2. Node has a fully installed and configured installation of Xen.
   - RECOMMENDED: verify your Xen installation by manually bringing up a VM and testing that it has network connectivity using bridged networking.
3. Firewall rules must permit the Eucalyptus components to communicate with one another, and clients to communicate with Eucalyptus.
   - NOTE: on the front-end, ports 8443, 8773, 8774 must be open. On the node, port 8775 must be open

Install RPMs on the front end ¶

In the examples below we use `x86_64`, which should be replaced with `i386` or, in some cases, `i586` on 32-bit architectures.

Because of a bug in the packaging of euca-axis2c and euca-httpd in 1.4 you have to use --nopostun flag when upgrading those packages (this option is harmless for a first-time install):

```
cd eucalyptus-1.5.1-rpm-deps-x86_64
rpm -Uvh --nopostun euca-axis2c-1.5-1.x86_64.rpm \
                    euca-httpd-1.5-1.x86_64.rpm
```

The rest of the third-party dependencies can be installed normally:

```
rpm -Uvh aoetools-21-1.el4.x86_64.rpm \
         vblade-14-1mdv2008.1.x86_64.rpm
cd ..
```

On the front end, where cloud controller and cluster controller will run, install the -cloud and -cc RPMs:

```
rpm -Uvh eucalyptus-1.5.1-1.x86_64.rpm \
         eucalyptus-cloud-1.5.1-1.x86_64.rpm \
         eucalyptus-gl-1.5.1-1.x86_64.rpm \
         eucalyptus-cc-1.5.1-1.x86_64.rpm
```

Install RPMs on the nodes ¶

Again, because of a bug in the packaging of euca-axis2c and euca-httpd in 1.4 you have to use --nopostun flag when upgrading those packages (this option is harmless for a first-time install):

```
cd eucalyptus-1.5.1-rpm-deps-x86_64
rpm -Uvh --nopostun euca-axis2c-1.5-1.x86_64.rpm \
                    euca-httpd-1.5-1.x86_64.rpm
```

The rest of the third-party dependencies can be installed normally:

```
rpm -Uvh aoetools-21-1.el4.x86_64.rpm \
         euca-libvirt-1.5-1.x86_64.rpm \
         vblade-14-1mdv2008.1.x86_64.rpm
cd ..
```

On the compute nodes, install the node controller RPM with dependencies:

```
rpm -Uvh eucalyptus-1.5.1-1.x86_64.rpm \
         eucalyptus-gl-1.5.1-1.x86_64.rpm \
         eucalyptus-nc-1.5.1-1.x86_64.rpm
```

OPTIONAL: Other Configurations ¶

If you would like to run Eucalyptus as a non-root user, perform the following procedure:

1. Stop all Eucalyptus services (if you've started them already, as described First-time next)
2. On both front-end and node(s):
    - add new user 'eucalyptus'
    - change EUCA_USER="eucalyptus"' in /opt/eucalyptus/etc/eucalyptus/eucalyptus.conf
3. On node(s):
    - enable '(xend-http-server yes)' in /etc/xend/xend-config.sxp
    - restart xend
4. Start all Eucalyptus services in the following order (order is important!)
    - on node(s): /etc/init.d/eucalyptus-nc start
    - on front-end: /etc/init.d/eucalyptus-cc start
    - on front-end: /etc/init.d/eucalyptus-cloud start

Installing Eucalyptus (1.5.1) on openSUSE 11

Eucalyptus can be installed on openSUSE 11 using binary RPM packages.

Download RPMs ¶

Eucalyptus binary installation is broken up into several packages: the cloud controller (-cloud package), the cluster controller (-cc package), and the node controller (-nc package). To simplify installation, everything is bundled into a single "tarball" for convenience, wherein we include copies of third-party packages that Eucalyptus depends on (Rampart, Axis2C, many Java libraries).

Download the appropriate tarball from http://open.eucalyptus.com/downloads

- For 32-bit machines, get eucalyptus-1.5.1-opensuse-i386.tar.gz
- For 64-bit machines, get eucalyptus-1.5.1-opensuse-x86_64.tar.gz

Untar the bundle in a temporary location:

```
tar zxvf eucalyptus-1.5.1-*.tar.gz
cd eucalyptus-1.5.1-*
```

All others dependencies are present in the standard repositories.

Prerequisites ¶

If you start with a standard openSUSE installation, you will satisfy all Eucalyptus prerequsites with the following steps:

1. Front-end, node and client machine system clocks are syncronized (i.e. using NTP).
2. Node has a fully installed and configured installation of Xen.
    - RECOMMENDED: verify your Xen installation by manually bringing up a VM and testing that it has network connectivity using bridged networking.
3. Firewall rules must permit the Eucalyptus components to communicate with one another, and clients to communicate with Eucalyptus.
    - NOTE: on the front-end, ports 8443, 8773, 8774 must be open. On the node, port 8775 must be open

Install RPMs on the front end ¶

In the examples below we use `x86_64`, which should be replaced with `i586` on 32-bit architectures.

Install the third-party dependency RPMs:

```
cd eucalyptus-1.5.1-rpm-deps-x86_64
rpm -Uvh aoetools-25-2.49.x86_64.rpm \
         euca-axis2c-1.5-1.x86_64.rpm \
         euca-httpd-1.5-1.x86_64.rpm \
         vblade-15-2.49.x86_64.rpm
cd ..
```

On front end, where cloud controller and cluster controller will run, install the -cloud and -cc RPMs:

```
rpm -Uvh eucalyptus-1.5.1-1.x86_64.rpm \
         eucalyptus-cloud-1.5.1-1.x86_64.rpm \
         eucalyptus-gl-1.5.1-1.x86_64.rpm \
         eucalyptus-cc-1.5.1-1.x86_64.rpm
```

Install RPMs on the nodes ¶

Install the dependency packages on each node:

```
cd eucalyptus-1.5.1-rpm-deps-x86_64
rpm -Uvh aoetools-25-2.49.x86_64.rpm \
         euca-axis2c-1.5-1.x86_64.rpm \
         euca-httpd-1.5-1.x86_64.rpm \
         vblade-15-2.49.x86_64.rpm
cd ..
```

On the compute nodes, install the node controller RPM with dependencies:

```
rpm -Uvh eucalyptus-1.5.1-1.x86_64.rpm \
         eucalyptus-gl-1.5.1-1.x86_64.rpm \
         eucalyptus-nc-1.5.1-1.x86_64.rpm
```

OPTIONAL: Other Configurations ¶

If you would like to run Eucalyptus as a non-root user, perform the following procedure:

1. Stop all Eucalyptus services
2. On both front-end and node(s):
   - add new user 'eucalyptus'
   - change EUCA_USER="eucalyptus"' in /opt/eucalyptus/etc/eucalyptus/eucalyptus.conf
3. On node(s):
   - enable '(xend-http-server yes)' in /etc/xend/xend-config.sxp
   - restart xend
4. Start all Eucalyptus services in the following order (order is important!)
   - on node(s): /etc/init.d/eucalyptus-nc start
   - on front-end: /etc/init.d/eucalyptus-cc start
   - on front-end: /etc/init.d/eucalyptus-cloud start

Installing Eucalyptus (1.5.1) on Debian Lenny (5.0.1)

Eucalyptus can be installed on Debian Lenny using binary DEB packages.

Download DEBs ¶

Eucalyptus binary installation is broken up into several packages: the cloud controller (-cloud package), the cluster controller (-cc package), and the node controller (-nc package). To simplify installation, everything is bundled into a single "tarball" for convenience, wherein we include copies of third-party packages that Eucalyptus depends on (Rampart, Axis2C, and Java libraries).

Download the appropriate tarball from http://open.eucalyptus.com/downloads

- For 32-bit machines, get eucalyptus-1.5.1-debian-i386.tar.gz
- For 64-bit machines, get eucalyptus-1.5.1-debian-amd64.tar.gz

Untar the bundle in a temporary location and add the directory to your `sources.list`

```
tar zxvf eucalyptus-1.5.1-*.tar.gz
cd eucalyptus-1.5.1-*
su
echo deb file://${PWD} ./ >> /etc/apt/sources.list
apt-get update
```

NOTE: After installation feel free to remove the entry from `sources.list`

Prerequisites ¶

If you start with a standard Debian Lenny installation, you will satisfy all Eucalyptus prerequsites with the following steps:

1. Front-end, node and client machine system clocks are syncronized (i.e., using NTP).
2. Firewall rules must permit the Eucalyptus components to communicate with one another, and clients to communicate with Eucalyptus.
    - NOTE: on the front-end, ports 8443, 8773, 8774 must be open. On the node, port 8775 must be open
3. Ensure that `locales` are properly configured (use `dpkg-reconfigure locales`

Node Controllers: eucalyptus-nc

4. Node has a fully installed and configured installation of Xen.
    - RECOMMENDED: verify your Xen installation by manually bringing up a VM and testing that it has network connectivity using bridged networking.
    - EXAMPLE: `/etc/xend/xend-config.sxp`

    ```
    (xend-http-server yes)
    (xend-unix-server yes)
    (xend-unix-path /var/lib/xend/xend-socket)
    (xend-address localhost)
    (network-script network-bridge)
    (vif-script vif-bridge)
    (dom0-min-mem 196)
    (dom0-cpus 0)
    (vncpasswd '')
    ```

Cloud Controller: eucalyptus-cloud

5. Fix `cacerts` for `openjdk-6-jdk` (missing from the package).
    - Add `non-free` to your apt sources file `/etc/apt/sources.list`, for example:

    ```
    su -
    echo deb http://debian.osuosl.org/debian lenny non-free >> /etc/apt/sources.list
    apt-get update
    ```

    - Install `sun-java6-jre` and create link for `cacerts`

    ```
    su -
    apt-get install ca-certificates sun-java6-jre
    mkdir -p /etc/ssl/certs/java/
    ln -sf /etc/java-6-sun/security/cacerts /etc/ssl/certs/java/cacerts
    ```

Install DEBs on the front end ¶

On front end, where cloud controller and cluster controller will run, install the -cloud and -cc DEBs:

```
aptitude install eucalyptus-cc eucalyptus-cloud
```

Install DEBs on the nodes ¶

On the compute nodes, install the node controller DEB:

```
aptitude install eucalyptus-nc
```

Installing Eucalyptus (1.5.1) on Debian Squeeze (sid)

Eucalyptus can be installed on Debian Squeeze using binary DEB packages.

Download DEBs ¶

Eucalyptus binary installation is broken up into several packages: the cloud controller (-cloud package), the cluster controller (-cc package), and the node controller (-nc package). To simplify installation, everything is bundled into a single "tarball" for convenience, wherein we include copies of third-party packages that Eucalyptus depends on (Rampart, Axis2C, and Java libraries).

Download the appropriate tarball from http://open.eucalyptus.com/downloads

- For 32-bit machines, get eucalyptus-1.5.1-debian-i386.tar.gz
- For 64-bit machines, get eucalyptus-1.5.1-debian-amd64.tar.gz

Untar the bundle in a temporary location and add the directory to your `sources.list`

```
tar zxvf eucalyptus-1.5.1-*.tar.gz
cd eucalyptus-1.5.1-*
su
echo deb file:///${PWD} ./ >> /etc/apt/sources.list
apt-get update
```

NOTE: After installation feel free to remove the entry from `sources.list`

Prerequisites ¶

If you start with a standard Debian Squeeze installation, you will satisfy all Eucalyptus prerequsites with the following steps:

1. Front-end, node and client machine system clocks are syncronized (i.e., using NTP).
2. Firewall rules must permit the Eucalyptus components to communicate with one another, and clients to communicate with Eucalyptus.
    - NOTE: on the front-end, ports 8443, 8773, 8774 must be open. On the node, port 8775 must be open
3. Ensure that `locales` are properly configured (use `dpkg-reconfigure locales`

Node Controllers: eucalyptus-nc

4. Node has a fully installed and configured installation of Xen.
    - RECOMMENDED: verify your Xen installation by manually bringing up a VM and testing that it has network connectivity using bridged networking.
    - EXAMPLE: `/etc/xend/xend-config.sxp`

```
(xend-http-server yes)
(xend-unix-server yes)
(xend-unix-path /var/lib/xend/xend-socket)
(xend-address localhost)
(network-script network-bridge)
(vif-script vif-bridge)
(dom0-min-mem 196)
(dom0-cpus 0)
(vncpasswd '')
```

Install DEBs on the front end ¶

On front end, where cloud controller and cluster controller will run, install the -cloud and -cc DEBs:

```
aptitude install eucalyptus-cc eucalyptus-cloud
```

Install DEBs on the nodes ¶

On the compute nodes, install the node controller DEB:

```
aptitude install eucalyptus-nc
```

Installing Eucalyptus (1.5.1) on Ubuntu Jaunty (9.04)

Eucalyptus can be installed on Ubuntu Jaunty using binary DEB packages.

Download DEBs ¶

Eucalyptus binary installation is broken up into several packages: the cloud controller (-cloud package), the cluster controller (-cc package), and the node controller (-nc package). To simplify installation, everything is bundled into a single "tarball" for convenience.

Download the appropriate tarball from http://open.eucalyptus.com/downloads

- For 32-bit machines, get eucalyptus-1.5.1-ubuntu-i386.tar.gz
- For 64-bit machines, get eucalyptus-1.5.1-ubuntu-amd64.tar.gz

Untar the bundle in a temporary location and add the directory to your `sources.list`

```
tar zxvf eucalyptus-1.5.1-*.tar.gz
cd eucalyptus-1.5.1-*
sudo -s
echo deb file://${PWD} ./ >> /etc/apt/sources.list
apt-get update
```

NOTE: After installation feel free to remove the entry from `sources.list`

Prerequisites ¶

If you start with a standard Ubuntu Jaunty installation, you will satisfy all Eucalyptus prerequsites with the following steps:

1. Front-end, node and client machine system clocks are syncronized (i.e., using NTP).
2. Firewall rules must permit the Eucalyptus components to communicate with one another, and clients to communicate with Eucalyptus.
    - NOTE: on the front-end, ports 8443, 8773, 8774 must be open. On the node, port 8775 must be open

3. Your node machine(s) must be configured with a bridge as the primary interface, if running in SYSTEM networking mode (default). You must first uninstall or disable Network Manager (default with Ubuntu Desktop), then follow the procedure below (example):

```
sudo vi /etc/network/interfaces
```

Comment out any entry for your existing interfaces (eth0/eth1 etc) and add a bridge entry with your interfaces attached. For example, to have your bridge come up with all physical ethernet devices added to it, and have DHCP assign an address to the bridge:

```
auto br0
iface br0 inet dhcp
      bridge_ports all
```

For a static configuration with just eth0 attached (substitute your actual network parameters):

```
auto br0
iface br0 inet static
      address 192.168.12.20
      netmask 255.255.255.0
      network 192.168.12.0
      broadcast 192.168.12.255
      gateway 192.168.12.1
      dns-nameservers 192.168.12.1
      dns-search foobar foobar.com
      bridge_ports eth0
```

Install DEBs on the front end ¶

On front end, where cloud controller and cluster controller will run, install the -cloud and -cc DEBs:

```
aptitude install eucalyptus-cc eucalyptus-cloud
```

Install DEBs on the nodes ¶

On the compute nodes, install the node controller DEB:

```
aptitude install eucalyptus-nc
```

Eucalyptus on a Rocks cluster

If you want to install Eucalyptus on a Rocks cluster, you can now follow the regular CentOS RPM instructions. The Cloud Controller and the Cluster Controller will need to be installed on your Rocks front-end, and the Node Controller will need to be installed on any of your Rocks nodes that have been configured as 'VM Containers'.

Eucalyptus will **not** run on a Rocks virtual cluster!

If you have previously installed Eucalyptus on your Rocks cluster, you should disable the old Eucalyptus rolls:

```
rocks disable roll eucalyptus
```

rebuild the rocks distribution:

```
cd /home/install
rocks-dist dist
```

and finally reinstall the nodes as 'VM Containers'. You can then follow the CentOS RPM instructions to install and configure Eucalyptus.

Keep in mind that the `java` roll in Rocks V (and V.I) includes JDK version 1.5 which is **not** enough to run Eucalyptus. You have to install the 1.6 JDK. For our testing we used Sun's JDK, which can be found at http://java.sun.com/javase/downloads/index.jsp.

Cloud Project : Centos 5.3x64, Rocks V.I and Eucalyptus 1.5.1 x64

Want to build a Rocks/Centos(RedHat Enterprise)/Eucalyptus Cloud - http://hackdaddy.wordpress.com/

Here is an example of my development and deployment of a Cloud Computing Cluster using Centos5.3x86_64, Rock V.Ix86_64 and Eucalyptus v1.5.1x86_64 on a AMD Opteron Dual x64 / HeadNode and Intel Core 2 Quad-Core / Compute Nodes. Please read this entire tutorial first because you may just need to run a script that would handle some configuration.

Requirements: (for 1 headnode and 2 compute node environment)

Intel or AMD Hyper-V Compatible multicore Processors x86_64

4gb + RAM

100 gb harddrive (/export mount point and / (root) must be atleast 40 gb)

HeadNode: Must have 2 network interface card (preferably gigabit ethernet)

ComputeNode: At Least 1 gigabit ethernet card

Gigabit Ethernet Switch

Resources:

* Rocks V.I x86_64 CD Set (Kernel/Boot Roll, Base Roll, WebServer Roll, Ganglia Roll)
* CentOS 5.3 x86_64 CD Set all discs (replace the Rocks OS set all Centos Discs)
* Eucalyptus v1.5.1 x86_64 (Get the generic package)

Components (some of these will be added as a part of the Rocks implementation)

* Package manager (Yum, Apt , RPM (synaptic or up2date optional))
* DHCP Server (dhcp3)
* Xen 3+ (I recommend Xen 3.4)
* libvirt 6+ (the install comes with libvirt 3, plus there are issue with the one in the package on centos)
* Python 5 (not 6)
* Perl 1.8.5+
* Java 6+
* Apache 1.6+ (I used 1.7.1)
* rsync
* Firewall (iptables)
* sshd
* Text Editor (nano, kedit, gedit)
* KVM
* Qemu
* vlan
* sendmail
* iproute
* ntp
* bridge-utils
* vblade
* aoetools
* lvm2
* nmap (to see what's running on open ports)
* wget
* unzip
* Eucalyptus 1.5.1 (eucalyptus-1.5.1-centos-x86_64.tar.gz)
* Sample Image (euca-ttylinux.tgz)
* ec2-api-tools-1.3-30349
* ec2-ami-tools-1.3-26357

Configurations

check if hyper-v or paravirtualization compatible if no output not compatible:

Check on Physical Address Extensions – grep pae /proc/cpuinfo

Check Intel-VT or AMD-V for hardware virtualization.

Intel

grep vmx /proc/cpuinfo

AMD

grep svm /proc/cpuinfo

open ports: Headnode = 22, 25, 80, 443, 5900, 5901,8443, 8773, 8774
ComputeNode = 8775

also 5900, 5901 if you want to use vncviewer/vncserver there is also a

remote desktop utility for Centos, I believe vino.

Be sure selinux is disabled.

Use SYSTEM managed or default Networking configuration.

Instructions assume that you understand how to use linux and text editors. Also assumes that you understand networking, virtual machines and system configuration. Please do the configurations logged in with a root account. Please connect eth0 on internal lan to switch plus the nodes and eth1 on public wan. Be sure that the compute nodes can boot from network cards (PXE boot). * Don't forget to turn off PXE Boot after you have installed a vm-container-X-X.

First: Install the headnode by using the rocks documentation (Please Read entirely). You will need to begin by inserting your boot/kernel roll. Then type (omit any < > seen in this tutorial, unless stated otherwise) Follow Rocks Documentation however replace the OS Rolls with the entire CD set for CentOS 5.3 x86_64. Record your information but keep it secure, normally the defaults are sufficient.

Plan your network configuration: I put a 192.168.42.95 for my static ip on eth1 (public wan) because this is in my network configuration talk to your network admin and get your info you will need the dns and gateway. I put 10.1.1.1 on eth0 (cloud lan) because this is what I want. If it recommends a default use it.

Use manual partition for the system with atleast:

Use ext3 file system.

40 gb / (force primary)

Ram * 3 swap (force primary)

40 gb /export

the rest if necessary as you see fit. By followng rocks documentation install should go by smoothly. Reboot into Rocks/Centos.

Share internet on compute nodes through your head node. You can put the following in a shell script and run it change your configurations to match your own.

linux-network-share.sh

```
#!/bin/bash
# Created by nixCraft – www.cyberciti.biz
# Edited by Micah Rowland
IPT=”/sbin/iptables”
MOD=”/sbin/modprobe”

# set wan interface such as eth1 or ppp0
SHARE_IF=”eth1!

# clean old fw
echo “Clearing old firewall rules…”
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT

# Get some kernel modules
echo “Loading kernel modules…”
$MOD ip_tables
$MOD iptable_filter
$MOD iptable_nat
$MOD ip_conntrack
$MOD ipt_MASQUERADE
$MOD ip_nat_ftp
$MOD ip_nat_irc
$MOD ip_conntrack_ftp
$MOD ip_conntrack_irc

# Clean old rules if any, rhel specific but above will take care of everything
# service iptables stop

# unlimited traffic via loopback device
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT

echo “Setting ${SHARE_IF} as router interface…”
$IPT –table nat –append POSTROUTING –out-interface ${SHARE_IF} -j MASQUERADE

# Start other custom rules
#$IPT
# End other custom rules

echo “*** Instructions on TCP/IP On The Windows / Mac / Linux Masqueraded Client ***”
echo “1. Login to your other LAN node computers”
echo “2. Use a text editor to modify your network setting with the info below modified
for your network.”
echo “3. Set DNS (NS1 and NS2) to 192.168.43.80 and 192.168.43.81!
echo “4. Create a route-eth0 file in /etc/sysconfig/network-scripts with appropriate permissions”
echo “5. Enter $(ifconfig ${SHARE_IF} | grep ‘inet addr:'| grep -v '127.0.0.1" | cut -d: -f2 | awk ‘{ print $1}') as the default gateway.”
```

This adds a router for each subnet.
It goes via dev . Here is my route-eth0 file. change yours accordingly.

```
# Added by Micah 7/7/09
default 10.1.1.1 dev eth0
10.1.0.0/24 via 10.1.1.1 dev eth0
192.168.42.0/24 via 192.168.42.95 dev eth0
192.168.0.0/24 via 192.168.122.1 dev eth0
```

Second: Let's begin by configuring our package managers. We will be adding apt support considering most of the documentation is written in hopes of using Ubuntu 9.04 Server

In your yum.conf file enter this:

```
[dag]
name=Dag RPM Repository for Red Hat Enterprise Linux
baseurl=http://apt.sw.be/redhat/el$releasever/en/$basearch/dag
#baseurl=http://ftp.riken.jp/Linux/dag/redhat/el$releasever/en/$basearch/dag
gpgcheck=1
```

then run:

```
rpm –import http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
rpm –import http://dag.wieers.com/rpm/packages/RPM-GPG-KEY.dag.txt
```

Then get the following repo file and add them here: /etc/yum.repos.d

CentOS-GITCO.repo

CentOS-GITCO_TESTING.repo

Now, yum install apt

After installation update the os.list file for apt for your CentOS Distro found here: /etc/apt/sources.list.d

edit os.list and add:

```
### CENTOS
repomd http://mirror.centos.org/ centos/$(VERSION)/os/$(ARCH)
repomd http://mirror.centos.org/ centos/$(VERSION)/updates/$(ARCH)
repomd http://mirror.centos.org/ centos/$(VERSION)/extras/$(ARCH)
```

also, uncomment any centos or redhat enterprise lines within reason. Modification may be needed. Please google.

I don't update, but if you don't have the version I mentioned above please update til you do. Please review the update components also.

Third: Eucalyptus pre-requisites. Now I used java.sun binary install for java 6 and apache foundation for ant 1.7.1 but Euclyptus team feels you should do this if using rocks then remove the <-y> cause you may have conflicts:

```
yum install -y java-1.6.0-openjdk-devel ant ant-nodeps libvirt-devel
curl-devel httpd httpd-devel apr-devel openssl-devel dhcp
```

This may not be necessary because you used rocks, but remember not to get rocks confused with what you are doing here which is a eucalyptus cloud, not rocks cluster, because you want the amazon tools. Try it couldn't hurt.

Fourth: Installation of Eucalyptus

Verify all node clocks are synchronized with ntp. Verify all required ports previously mentioned are open, Headnode = 22, 25, 80, 443, 5900, 5901, 8443, 8773, 8774 ComputeNode = 8775. This can be done through iptables or use the utility in Centos unders: System > Administration > Security Level and Firewall. Since the Compute nodes have no gui then do the iptables commands. see iptables or google or try. Later we will do dns masquerading so that your compute nodes get internet access. So, you may want to revisit this later to add a new rule.

/etc/sysconfig/iptables (verify you iptables file it may be different than this one)

```
iptables -I INPUT -p tcp –dport 8775 –syn -j ACCEPT
```

```
/etc/rc.d/init.d/iptables restart
```

Verify Xen get's network connectivity on the compute nodes by manual loading a xen image. Please see xen documentation or google.

All downloads should be in your /opt directory. Now run:

```
tar zxvf eucalyptus-1.5.1-*.tar.gz
cd eucalyptus-1.5.1-*
```

then

go into the created directory eucalyptus-1.5.1-* and go into deps
directory and run the rpms. Before running the deps or eucalyptus,
If you are upgrading or had a different version installed please see
the documentation at open.eucalyptus.com. Or is you installed a Eucalyptus roll.

Run your rpm's in order such as dependencies then libraries. see eucalyptus documentation.

In your /opt directory run:

```
unzip ec2-api-tools-1.3-30349.zip
```

```
unzip ec2-ami-tools-1.3-26357.zip
```

Fifth: Add your System Paths for Eucalyptus to access resources. You add this to the end of the /etc/profile or /etc/bashrc file. Like so:

```
export JAVA_HOME=/usr/java/jdk1.6.0_14
export EC2_HOME=/opt/ec2-api-tools-1.3-30349
export EC2_AMITOOL_HOME=/opt/ec2-ami-tools-1.3-26357
export EUCALYPTUS=/opt/eucalyptus
export PATH=$PATH:$HOME/.euca:$EUCALYPTUS:$JAVA_HOME/bin:/opt/apache-ant-
1.7.1/bin:$EC2_HOME/bin:$EC2_AMITOOL_HOME/bin
```

Reference Links:

Rocks Documentation – http://www.rocksclusters.org/wordpress/?page_id=4

LinuxStreet – http://www.linuxstreet.net/news/E/19987/Install-Xen-3-3-on-CentOS-5-2-vi...

Eucalyptus Doc – http://open.eucalyptus.com/wiki/EucalyptusPrerequisites_v1.5

Field Commander Wieers – http://dag.wieers.com/blog/using-apt-in-an-rpm-world

Gitco – http://www.gitco.de/linux/x86_64/centos/5/

CentOS / Redhat Linux Internet Connection Sharing – http://www.cyberciti.biz/faq/rhel-fedora-linux-internet-connection-shari...

Trouble Shoot and Support – http://forum.eucalyptus.com/forum/ (must sign up)

Downloads for this tutorial only.

wget http://s3.amazonaws.com/ec2-downloads/ec2-api-tools-1.3-30349.zip

wget http://s3.amazonaws.com/ec2-downloads/ec2-ami-tools-1.3-26357.zip

Eucalyptus v1.5.1x86_64 Centos 5.3 – http://open.eucalyptus.com/downloads/90

euca-ttylinux – http://open.eucalyptus.com/downloads/3

Upgrading to Eucalyptus 1.5.1 from 1.4

These are instructions for those who would like to upgrade to Eucalyptus 1.5 from source-based or RPM-based 1.4 installation. If you're still running 1.3, please, follow the instructions for upgrading to 1.4 before following these instructions.

These upgrade instructions involve making a backup of the key state. The last section of this document explains how to roll back to the previous installation using the backup.

Commands below assume that your **$EUCALYPTUS** variable points to the root of the (old) 1.4 installation. The typical 1.4 installation is in `/opt/eucalyptus`. After moving the 1.4 installation away or backing it up, the same location will be re-used for the 1.5 installation. If you want to install 1.5 somewhere else, adjust the commands accordingly.

1. Clean up Eucalyptus running state ¶

- Write down the value of the "Walrus path" listed under the "Configuration" tab of the Web interface. (That is where all uploaded images and user buckets are located.)
- Terminate **all** Eucalyptus instances

```
ec2-terminate-instances ...      # (as admin)
```

- Shut down Eucalyptus on **all** nodes.

  ```
  $EUCALYPTUS/etc/init.d/eucalyptus stop
  ```

- Check for errant Eucalyptus processes on **all** nodes and kill them

  ```
  ps aux | grep euca
  kill -9 ...
  ```

2. Install Eucalyptus 1.5 ¶

Both source- and RPM-based installations can be upgraded:

- **Option A:** Source-based installation upgrade:
  - Move away or back up the 1.4 installation:
    - Pick a backup location:

      ```
      export EUCALYPTUS_OLD=/opt/eucalyptus-1.4
      ```

    - If your 1.4 installation is in a separate directory, such as `/opt/eucalyptus`, you can just move it away on the head-node and all compute nodes. E.g.:

      ```
      mv $EUCALYPTUS $EUCALYPTUS_OLD
      ```

    - Else, if you installed 1.4 in a location shared with other software, such as `/` or `/usr/local`, you'll have to move away or tar up the following directories and files one-by-one:

      ```
      $EUCALYPTUS/etc/init.d/eucalyptus
      $EUCALYPTUS/etc/eucalyptus
      $EUCALYPTUS/usr/sbin/euca_*
      $EUCALYPTUS/usr/share/eucalyptus
      $EUCALYPTUS/var/eucalyptus/[^b]*      # (everything except the "bukkits" directory)
      ```

    - Also, if you'd like to clean up, you might want to remove the following two directories:

      ```
      $EUCALYPTUS/var/log/eucalyptus
      $EUCALYPTUS/var/run/eucalyptus
      ```

  - Follow the steps in the Source Code Installation section of the Administrator's Guide and, afterwards, **return here**.
  - Copy back the old database and keys (note that the location has changed slightly):

    ```
    cp $EUCALYPTUS_OLD/var/eucalyptus/db/eucalyptus.* $EUCALYPTUS/var/lib/eucalyptus/db/
    cp $EUCALYPTUS_OLD/var/eucalyptus/db/eucalyptus_volumes.* $EUCALYPTUS/var/lib/eucalyptus/db/
    rm -f $EUCALYPTUS/var/eucalyptus/db/*.lck
    cp $EUCALYPTUS_OLD/var/eucalyptus/keys/* $EUCALYPTUS/var/lib/eucalyptus/keys/
    ```

  - Since the "Walrus path" (`$EUCALYPTUS/var/eucalyptus/bukkits` by default) potentially contains a lot of data, we do not recommend copying it. If you were using the default location, just move it back:

    ```
    mv $EUCALYPTUS_OLD/var/eucalyptus/bukkits $EUCALYPTUS/var/eucalyptus
    ```

    (FYI: The official default Walrus path in 1.5 is `$EUCALYPTUS/var/lib/eucalyptus`. If you want to move the buckets there, for consistency with the official value, be sure to also change the Walrus Path setting under the Configuration tab of the Web interface.)

- **Option B:** RPM-based installation upgrade:
  - Follow the steps in the CentOS Installation section of the Administrator's Guide and, afterwards, **return here**.
  - The upgrade should leave *rollback* archives in /root which could be used to revert to 1.4.

3. Update the configuration ¶

- We provide a way to upgrade the configuration file

  ```
  $EUCALYPTUS/usr/sbin/euca_conf -upgrade-conf <old_config> <new_config>
  ```

  after which we strongly advice to go and check the generated configuration file: if you upgraded from an older RPM the above command was executed automatically during the upgrade.

- Edit `$EUCALYPTUS_OLD/etc/eucalyptus/eucalyptus.conf` file on head node. We suggest starting with the fresh one created during installation and copying over matching parts from the old one. (If you updated an RPM-based install, the new configuration is in `eucalyptus.conf.rpmnew` while `eucalyptus.conf` contains your old one.) Specifically, be sure to carry across the values of the following variables:
  - `*_PORT`
  - `NODES`
  - `INSTANCE_PATH`
  - `VNET_*`

- Set `HYPERVISOR="xen"` since your 1.4 installation was Xen-based. You can later switch to a KVM-based installation.

- Setting the EUCA_USER variable according to the type of installation that you performed in step 2 above (running as root is easier to configure, but it requires one to use a specially compiled Apache).
- Network parameters from 1.4 should continue to work. To learn about the new network mode (MANAGED-NOVLAN) see the Network Configuration section of the Administrator's Guide..
- On each compute node, go over the config file, comparing it with the old one, same as with the head-node file. (Naturally, you are free to create a new config file for compute-nodes and `rsync` it with all the nodes.)
- Synchronize node keys

  ```
  $EUCALYPTUS/usr/sbin/euca_sync_key -c $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf
  ```

- Start NCs, the CC, and the CLC. Each one of those services has its own startup script in 1.5, so perform some combination of these on the appropriate machines:

  ```
  $EUCALYPTUS/etc/init.d/eucalyptus-nc start
  $EUCALYPTUS/etc/init.d/eucalyptus-cc start
  $EUCALYPTUS/etc/init.d/eucalyptus-clc start
  ```

- In a Web browser, load `https://headnode:8443/` and log in as before. Verify that the user accounts and the images are there.
- Verify that the nodes are back up and that they can run your old instances (if not, see the Troubleshooting section.)

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

4. Clean up old disk state ¶

Once you are confident that the new installation is working, delete the old state on disk.

```
rm -rf $EUCALYPTUS_OLD
```

If you upgraded using RPM packages, delete on all nodes the backup of old state that was created during the upgrade:

```
rm /root/eucalyptus-pre-1.5-rollback.tar
```

The upgrade moved some directories location (in order to comply to FHS) so you may have to remove by hand some directory in particular

```
rm -rf $EUCALYPTUS/var/eucalyptus
```

6. Rolling back to an earlier installation ¶

- Stop Eucalyptus 1.5 processes, if any, on all nodes
- **Option A:** Rolling back source-based 1.5 upgrade:
  - Remove any files added during the failed upgrade. For example:

    ```
    rm -rf $EUCALYPTUS
    ```

  - Move back the old installation on all nodes:

    ```
    mv $EUCALYPTUS_OLD $EUCALYPTUS
    ```

- **Option B:** Rolling back RPM-based 1.5 upgrade:
  - Remove the failed RPMs on all affected nodes (depending on the failure, you might have to use the `--nopreun` option)

    ```
    rpm -e eucalyptus-cloud eucalyptus-cc euca-httpd euca-axis2c euca-libvirt eucalyptus
    ```

  - Download and install the 1.4 RPMs on all nodes as discussed in the Administrator's Guide. For example, on the front-end you can use:

    ```
    rpm -ivh eucalyptus-1.4-2.i386.rpm eucalyptus-cloud-1.4-2.i386.rpm eucalyptus-cc-1.4-2.i386.rpm eucalyptus-gl-1.4-2.i386.rpm
    ```

  - Copy the old state saved during the upgrade process:

    ```
    cd $EUCALYPTUS
    tar xf /root/eucalyptus-pre-1.5-rollback.tar
    ```

  - Copy over the old configuration file

    ```
    cd $EUCALYPTUS
    cp etc/eucalyptus/eucalyptus.conf.old etc/eucalyptus/eucalyptus.conf
    ```

- Start Eucalyptus, as before

Eucalyptus Configuration (1.5.1)

This document describes the steps for configuring Eucalyptus after the software has been installed on all nodes (either from source or using binary packages). Instructions below assume that you have variable `$EUCALYPTUS` set. For RPM-based installations, `$EUCALYPTUS` is `/opt/eucalyptus/` while for DEB-based installations, `$EUCALYPTUS` is '/'. So set it appropriately:

```
export EUCALYPTUS=....
```

Eucalyptus comes with the `euca_conf` script for setting up the configuration file located in '$EUCALYPTUS/etc/eucalyptus/eucalyptus.conf'. Alternatively, you can change that file in a text editor instead of using 'euca_conf'.

1. First-time Configuration ¶

Eucalyptus installation consists of three types of components: cloud controller (CLC), cluster controller (CC), and the node controller(s) (NCs). In following instructions we assume that CLC and CC are co-located on a machine that we will refer to as the *front end* and that NCs run on *compute nodes*. The instructions will also work if one physical machine fulfills the role of both the front end and a compute node.

IMPORTANT: Make sure that the cloud controller is up and running before going through the following steps.

a. Front-end Configuration ¶

To connect the Eucalyptus components together, you will need to register the Cluster with the Cloud, and register each Node with the Cluster. On the front-end, do:

```
$EUCALYPTUS/usr/sbin/euca_conf -addcluster <clustername> <clusterhost> $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf
```

where <clustername> is the name your would like to attach to your Cluster, and <clusterhost> is the hostname of the machine or the IP where the Cluster Controller is running.

Also on the front-end, add the hostnames on which you plan to run node controllers one-by-one (this involves connecting to the node via SSH to propagate the cryptographic keys, so you may be prompted for a password):

```
$EUCALYPTUS/usr/sbin/euca_conf -addnode <nodehost> $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf
```

where <nodehost> is the hostname or IP of your node.

Alternatively, you can add nodes all at once with the `-nodes` option, which requires you to explicitly propagate cryptographic keys afterwards:

```
$EUCALYPTUS/usr/sbin/euca_conf -nodes "<nodehost1> ... <nodehostN>" $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf
$EUCALYPTUS/usr/sbin/euca_sync_key -c $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf
```

OPTIONAL: Eucalyptus provides some options when it comes to configuring your VM virtual network. By default, we enable the simplest but least feature-ful networking mode, called SYSTEM in the eucalyptus.conf file: Eucalyptus will assume that you already have a DHCP server configured to serve IP addresses to VMs that start on cluster nodes. Please consult the the brief explanation in the comments of the configuration file and the Eucalyptus Networking document if you wish to try other modes that will enable more features (security groups, elastic IPs, etc.).

b. Compute-node Configuration ¶

If you installed from binary packages you can now skip to step 2 since the compute nodes should be appropriately configured. If you later decide to diverge from the default configuration, you might want to revisit these steps.

On each compute node, create a local directory where VM images are placed temporarily when VMs are running (images will be cached under the same path, too). Instruct the nodes to run the node controller, choose what hypervisor to use (`xen` or `kvm`), and specify the path for VM images. This path is used to store temporary VM images and it's important it's empty (everything in it will be removed!).

```
for x in hostname1 hostname2 ... hostnameN ; do \
        ssh $x "mkdir -p /usr/local/instances/; $EUCALYPTUS/usr/sbin/euca_conf -hypervisor kvm -instances /usr/local/instances $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf"
done
```

Make sure that the user you have decided to run eucalyptus as (`username='eucalyptus'` in the above example) has the ability to control VMs through the node controller machine's libvirt installation. A good test is to run the command `virsh list` as the eucalyptus user to see if that user has the appropriate rights.

Finally, ensure that the networking settings in 'eucalyptus.conf' on each of your nodes is configured properly. For instance, correct values for VNET_INTERFACE and VNET_BRIDGE may differ from your front-end. See Eucalyptus Networking for more details.

2. Running Eucalyptus ¶

First, make sure that you have all of the runtime dependencies of Eucalyptus installed, based on your chosen set of configuration parameters. If there is a problem with runtime dependencies (for instance, if Eucalyptus cannot find/interact with them), all errors will be reported in log files located in $EUCALYPTUS/var/log/eucalyptus.

Use the init-scripts to start each component on the appropriate host. Most likely, on the front-end you would run:

```
$EUCALYPTUS/etc/init.d/eucalyptus-cloud start
$EUCALYPTUS/etc/init.d/eucalyptus-cc start
```

And on each of the compute nodes you would run:

```
$EUCALYPTUS/etc/init.d/eucalyptus-nc start
```

To stop them you call the script with *stop* instead of start.

If you installed from binary packages you can now skip to step 3. If you installed from source and you want to have eucalyptus started automatically when your machines are (re)booted, you can add the following symlinks on the appropriate hosts

```
ln -sf $EUCALYPTUS/etc/init.d/eucalyptus-cloud /etc/init.d/eucalyptus-cloud
ln -sf $EUCALYPTUS/etc/init.d/eucalyptus-cc /etc/init.d/eucalyptus-cc
ln -sf $EUCALYPTUS/etc/init.d/eucalyptus-nc /etc/init.d/eucalyptus-nc
```

and then add the symlinks to the distribution's booting process. This process differs from distribution to distribution. For example if you have `update-rc.d` available you can run:

```
update-rc.d eucalyptus-cloud defaults
```

or if you have `chkconfig` available you can run:

```
chkconfig eucalyptus-cloud on
```

3. First-time Run-time Setup ¶

To configure eucalyptus, after you started all components, login to

> https://localhost:8443

where you should substitute localhost with the name of the host running the cloud controller. (WARNING: on some machines it may take few minutes after the starting of the Cloud Controller for the URL to be responsive the first time you run Eucalyptus.) You will be prompted for a user/password which are set to admin/admin. Upon logging in you will be guided through three first-time tasks:

1. You will be forced to change the admin password.
2. You will be asked to set the admin's email address.
3. You will be asked to confirm the URL of the Walrus service (the storage component of Eucalyptus) which should start with the hostname or IP address of the cluster head node where you are installing the ClC.

After completing the first-time tasks, you will see the 'Configuration' tab. To use the system with the EC2 client tools, you must generate user credentials. Click the 'Credentials' tab and download your certificates via the 'Download certificates' button. You will be able to use these x509 certificates with Amazon EC2 tools and third-party tools like rightscale.com.

Create a directory, for example $HOME/.euca,

```
mkdir $HOME/.euca
```

unpack the credentials into it, and source the included 'eucarc':

```
. $HOME/.euca/eucarc
```

Note that you will have to source this file every time you intend to use the EC2 command-line tools, or you may add it to your local default environment.

Eucalyptus Network Configuration (1.5.1)

Eucalyptus versions 1.5 and higher includes a highly configurable VM networking subsystem that can be adapted to a variety of network environments. There are four high level networking "modes", each with its own set of configuration parameters, features, benefits and in some cases restrictions placed on your local network setup. The administrator must select one of these four modes before starting Eucalyptus on the front-end and nodes via modification of the 'eucalyptus.conf' configuration file on each machine running a Eucalyptus component. Brief descriptions of each mode follows:

> **SYSTEM Mode** - This is the simplest networking mode, but also offers the smallest number of networking features. In this mode, Eucalyptus simply assigns a random MAC address to the VM instance before booting and attaches the VM instance's ethernet device to the physical ethernet through the node's local Xen bridge. VM instances typically obtain an IP address using DHCP, the same way any non-VM machine using DHCP would obtain an address. Note that in this mode, the Eucalyptus administrator (or the administrator that manages the network to which Eucalyptus components are attached) must set up a DHCP server that has a dynamic pool of IP addresses to hand out as VMs boot. In other words, if your laptop/desktop/server gets an IP address using DHCP on the same network as the Eucalyptus nodes, then your VMs should similarly obtain addresses. This mode is most useful for users who want to try out Eucalyptus on their laptops/desktops.
>
> **STATIC Mode** - This mode offers the Eucalyptus administrator more control over VM IP address assignment. Here, the

administrator configures Eucalyptus with a 'map' of MAC address/IP Address pairs. When a VM is instantiated, Eucalyptus sets up a static entry within a Eucalyptus controlled DHCP server, takes the next free MAC/IP pair, assigns it to a VM, and attaches the VMs ethernet device to the physical ethernet through the Xen bridge on the nodes (in a manner similar to SYSTEM mode). This mode is useful for administrators who have a pool of MAC/IP addresses that they wish to always assign to their VMs.

NOTE - Running Eucalyptus in SYSTEM or STATIC mode disables some key functionality such as the definition of ingress rules between collections of VMs (termed security groups in Amazon EC2), the user-controlled, dynamic assignment of IPs to instances at boot and run-time (elastic IPs in Amazon EC2), and isolation of network traffic between VMs (that is, the root user within VMs will be able to inspect and potentially interfere with network traffic from other VMs).

MANAGED Mode - This mode is the most featureful of the three modes, but also carries with it the most potential constraints on the setup of the Eucalyptus administrator's network. In MANAGED mode, the Eucalyptus administrator defines a large network (usually private, unroutable) from which VM instances will draw their IP addresses. As with STATIC mode, Eucalyptus will maintain a DHCP server with static mappings for each VM instance that is created. Eucalyptus users can define a number of 'named networks', or 'security groups', to which they can apply network ingress rules that apply to any VM that runs within that 'network'. When a user runs a VM instance, they specify the name of such a network that a VM is to be a member of, and Eucalyptus selects a subset of the entire range of IPs that other VMs in the same 'network' can reside. A user can specify ingress rules that apply to a given 'network', such as allowing ping (ICMP) or ssh (TCP, port 22) traffic to reach their VMs. This capability allows Eucalyptus expose a capability similar to Amazon's 'security groups'. In addition, the administrator can specify a pool of public IP addresses that users may allocate, then assign to VMs either at boot or dynamically at run-time. This capability is similar to Amazon's 'elastic IPs'. Eucalyptus administrators that require security groups, elastic IPs, and VM network isolation must use this mode.

MANAGED-NOVLAN Mode - This mode is identical to MANAGED mode in terms of features (dynamic IPs and security groups) but does not provide VM network isolation. Admins who want dynamic assignable IPs and the security groups, but are not running on a network that is 'VLAN clean' or don't care if their VMs are isolated from one another on the network should choose this mode.

Each Eucalyptus network mode has its own set of infrastructure requirements, configuration parameters, and caveats. These are described in more detail in the following sections.

SYSTEM Mode ¶

There is very little Eucalyptus configuration to use SYSTEM mode, as in this mode, Eucalyptus mostly stays 'out of the way' in terms of VM networking. The options in 'eucalyptus.conf' that must be configured correctly in 'SYSTEM' mode are as follows:

On the front-end:

```
VNET_MODE="SYSTEM"
```

On each node:

```
VNET_MODE="SYSTEM"
VNET_BRIDGE
```

In each Eucalyptus node controller's (NC) 'eucalyptus.conf' file, make sure that the parameter 'VNET_BRIDGE' is set to the name of the Xen bridge device that is connected to your local ethernet. In Xen 3.0 (and some other versions), the name of the bridge, by default, was 'xenbr0'. If you have such an installation, specify it like so:

```
VNET_BRIDGE="xenbr0"
```

In Xen 3.2 and higher, the name of the bridge (most of the time) is set to the name of your ethernet device (generally 'eth0'). If this is the case on your system, set configure Eucalyptus like so:

```
VNET_BRIDGE="eth0"
```

Make sure that what you are specifying in this field is actually a bridge, and that it is the bridge that is connected to an ethernet network that has a DHCP server running elsewhere that is configured to hand out IP addresses dynamically. Use the 'brctl show' command to inspect the status of your local bridges. Note that your front-end machine may not have any bridges if Xen is not installed (this is fine, as VNET_BRIDGE is only a relevant for node controllers, and will be safely ignored by the front-end components).

To test whether this mode is working properly at run-time, start an instance and log in to the node where the instance is running. Run 'xm list' to find the Xen ID that your instance is running under. Then, look at the output of 'brctl show', it should look something like this (assuming your VNET_BRIDGE is set to 'eth0', and the Xen ID of your instance was '18'):

```
; brctl show eth0
bridge name  bridge id          STP enabled     interfaces
eth0         8000.000c29369858  no              peth0
                                                vif18.0
```

note that Eucalyptus has correctly attached the VM's 'eth0' interface (vif18.0) to the bridge ('eth0') that is being used to attach VMs to the local ethernet ('peth0'). At this point, the VM should be sending DHCP requests to the local ethernet, and the DHCP server on the network should be sending a reply.

CAVEATS - In this mode, as mentioned previously, VMs are simply started with their ethernet interfaces attached to the local ethernet without any isolation. Practically, this means that you should treat a VM the same way that you would treat a non-VM machine running on the network. Eucalyptus does it's best to discover the IP address that was assigned to a running VM via a third-party DHCP server, but can be unsuccessful depending on the specifics of your network (switch types/configuration, location of CC on the network, etc.). Practically, if Eucalyptus cannot determine the VM's IP, then the user will see '0.0.0.0' in the output of 'describe-instances' in both the private and public address fields. The best workaround for this condition is to instrument your VMs to send some network traffic to your front end on boot (after they obtain an IP address). For instance, setting up your VM to ping the front-end a few times on boot should allow Eucalyptus to be able to discover the VMs IP.

STATIC Mode ¶

In this mode, Eucalyptus will manage VM IP address assignment by maintaining its own DHCP server with one static entry per VM. The options in 'eucalyptus.conf' that must be configured correctly in 'STATIC' mode are as follows:

On the front-end (options annotated with a '*' may be required depending on your installation, see below for details):

```
VNET_MODE="STATIC"
VNET_INTERFACE
VNET_DHCPDAEMON
*VNET_DHCPUSER
VNET_SUBNET
VNET_NETMASK
VNET_BROADCAST
VNET_ROUTER
VNET_DNS
VNET_MACMAP
```

On each node:

```
VNET_MODE="STATIC"
VNET_BRIDGE
```

The Eucalyptus administrator must configure the front-end's 'eucalyptus.conf' first with a valid, configured ethernet device that is attached to the same physical ethernet as the Eucalyptus nodes:

```
VNET_INTERFACE="eth0"
```

Next, the admin ust ensure that there is a DHCP server binary installed on the front-end and Eucalyptus knows where it is located:

```
VNET_DHCPDAEMON="/usr/sbin/dhcpd3"
```

If your DHCP daemon binary is configured to run as 'non-root' (say, as the user 'dhcpd' as is the case in Ubuntu >= 8.10), then you must configure Eucalyptus to be aware of that user:

```
VNET_DHCPUSER="<dhcpusername>"
```

Then, the admin must input IP subnet information for that device. For example, if the front-end's 'eth0' interface has the IP address '192.168.1.254' on the '192.168.1.0/24' network, with a gateway at '192.168.1.1' and a DNS at '192.168.1.2', the values in 'eucalyptus.conf' would look like so:

```
VNET_SUBNET="192.168.1.0"
VNET_NETMASK="255.255.255.0"
VNET_BROADCAST="192.168.1.255"
VNET_ROUTER="192.168.1.1"
VNET_DNS="192.168.1.2"
```

Finally, the administrator must supply a list of static MAC/IP mappings that will be assigned, first come first served, to VM instances. Note that each IP must reside in the subnet defined above, and must not be in use by any other machine on the network.

```
VNET_MACMAP="AA:DD:11:CE:FF:ED=192.168.1.3 AA:DD:CE:FF:EE=192.168.1.4"
```

On the nodes, you must ensure that the bridge is entered (typically 'xenbr0' on Xen 3.0 and 'eth0' on Xen >= 3.2). Run the command 'brctl show' to inspect your nodes' bridge setup.

```
VNET_BRIDGE="xenbr0"
```

Once you have configured Eucalyptus properly, start up the node controllers and the front-end components. To test whether this mode is working properly at run-time, start an instance and log in to the node where the instance is running. Run 'xm list' to find the Xen ID that your instance is running under. Then, look at the output of 'brctl show', it should look something like this (assuming your VNET_BRIDGE is set to 'eth0', and the Xen ID of your instance was '18'):

```
; brctl show eth0
bridge name  bridge id          STP enabled     interfaces
eth0         8000.000c29369858  no              peth0
                                                vif18.0
```

note that Eucalyptus has correctly attached the VM's 'eth0' interface (vif18.0) to the bridge ('eth0') that is being used to attach VMs to the local ethernet ('peth0'). Make sure that the DHCP server has been started properly on the front-end ('ps axww | grep -i dhcpd | grep -i euca'). At this point, the VM should be sending DHCP requests to the local ethernet, and the DHCP server on the front-end should be sending a reply with one of the static MAC/IP mappings the admin has defined in 'eucalyptus.conf'.

CAVEATS - In this mode, as mentioned previously, VMs are started with their ethernet interfaces attached to the local ethernet without any isolation. Practically, this means that you should treat a VM the same way that you would treat a non-VM machine running on the network. Eucalyptus does not verify that your settings are valid, thus, you must enter them correctly in order for your VMs to obtain IP addresses. Finally, we assume that the installed DHCP daemon is, or is compatible with, ISC DHCP Daemon version 3.0.X. If it is not, we recommend either installing a version that is (common in most distributions) or writing a wrapper script around your installed DHCP server and point Eucalyptus at it (via VNET_DHCPDAEMON in 'eucalyptus.conf').

MANAGED Mode ¶

In this mode, Eucalyptus will fully manage the local VM instance network and provides all of the networking features Eucalyptus current supports (VM network isolation, user controllable VM firewalls (ingress rules/security groups), dynamic public IP assignment). The options in 'eucalyptus.conf' that must be configured correctly in 'MANAGED' mode are as follows:

On the front-end (options annotated with a '*' may be required depending on your installation, see below for details):

```
VNET_MODE="MANAGED"
VNET_INTERFACE
VNET_DHCPDAEMON
*VNET_DHCPUSER
VNET_SUBNET
VNET_NETMASK
VNET_DNS
VNET_ADDRSPERNET
*VNET_PUBLICIPS
```

On each node:

```
VNET_MODE="MANAGED"
VNET_INTERFACE
```

Be advised that this mode requires that your local network/configuration conforms to certain requirements that Eucalyptus depends upon.

Requirements for MANAGED mode ¶

Before using 'MANAGED' mode, you must confirm that:

1.) there is an available range of iP addresses that is completely unused on the network (192.168..., 10....., other).

2.) your network is 'VLAN clean', meaning that all switch ports that Eucalyptus components are connected to will allow and forward VLAN tagged packets.

3.) you are not running a firewall on the front-end (CC) or your firewall is compatible with the dynamic changes that Eucalyptus will make to the front-end's netfilter rules.

All three of these requirements must be met before MANAGED mode should be attempted. Failure to verify the above will, at least, result VM instances being unavailable on the network.

For requirement '1', choose a IP range that you know is completely unused on your network. Choose a range that is as large as possible. Typical examples are:

if the network 10.0.0.0 - 10.255.255.255 is completely unused:

```
VNET_MODE="MANAGED"
VNET_SUBNET="10.0.0.0"
VNET_NETMASK="255.0.0.0"
VNET_DNS="<your DNS>"
VNET_ADDRSPERNET="128"
```

or if the network 192.168.0.0 - 192.168.255.255 is completely unused:

```
VNET_MODE="MANAGED"
VNET_SUBNET="192.168.0.0"
VNET_NETMASK="255.255.0.0"
VNET_DNS="<your DNS>"
VNET_ADDRSPERNET="64"
```

Next, the admin must verify that the local network will allow/forward VLAN tagged packets between machines running Eucalyptus components. To verify, perform the following test:

on the front-end, choose the interface that is on the local ethernet (and will be set in eucalyptus.conf as VNET_INTERFACE), and run:

```
vconfig add <interface> 10
ifconfig <interface>.10 192.168.1.1 up
```

replace '192.168.1.1' with an IP from the range you selected above.

On the node, choose the interface on the local network (will be set in eucalyptus.conf as VNET_INTERFACE), and run:

```
vconfig add <interface> 10
ifconfig <interface>.10 192.168.1.2 up
```

again, replace '192.168.1.2' with another IP in the range you selected above.

Then, try a ping between hosts. On the front-end:

```
ping 192.168.1.2
```

on the node:

```
ping 192.168.1.1
```

If this does not work, then your switch needs to be configured to forward VLAN tagged packets (if it is a managed switch, see your switch's documentation to determine how to do this).

Finally, you need to carefully inspect the firewall on the front-end to make sure that it will not interfere with Eucalyptus, or vice-versa. Eucalyptus will flush the 'filter' and 'nat' tables upon boot in MANAGED mode, but provides a way for the administrator to define special rules that are loaded when Eucalyptus starts (see below for details).

Configuring MANAGED mode ¶

The Eucalyptus administrator must configure the front-end's 'eucalyptus.conf' first with a valid, configured ethernet device that is attached to the same physical ethernet as the Eucalyptus nodes:

```
VNET_INTERFACE="eth0"
```

Next, the admin ust ensure that there is a DHCP server binary installed on the front-end and Eucalyptus knows where it is located:

```
VNET_DHCPDAEMON="/usr/sbin/dhcpd3"
```

If your DHCP daemon binary is configured to run as 'non-root' (say, as the user 'dhcpd' as is the case in Ubuntu >= 8.10), then you must configure Eucalyptus to be aware of that user:

```
VNET_DHCPUSER="<dhcpusername>"
```

Nodes must have VNET_INTERFACE set properly. For example, with current Xen versions, this parameter (when your node's Xen bridge is 'eth0') is typically:

```
VNET_INTERFACE="peth0"
```

Once you have verified that your network configuration meets the requirements for running in MANAGED mode, the rest of the configuration is fairly simple. For example, if the 192.168.0.0/16 network is free and unused on your network:

```
VNET_MODE="MANAGED"
VNET_SUBNET="192.168.0.0"
VNET_NETMASK="255.255.0.0"
VNET_DNS="<your dns>"
VNET_ADDRSPERNET="64"
VNET_PUBLICIPS="<publicIPa> <publicIPb> ... <publicIPn>"
```

SUBNET, NETMASK, and DNS have been described previously. VNET_ADDRSPERNET is used to control how many VM instances may simultaneously be part of an individual user's named network (called a 'security group' in Amazon EC2). Choosing the right value for this parameter depends on how many IPs you have made available using VNET_SUBNET/VNET_NETMASK, how many VLANs your network supports simultaneously, and how many concurrent active user networks the administrator wishes to support. In the above example, there are 65536 addresses available (192.168.0.0/16). If we divide by the number of addresses per network (set to 64 above), we find the maximum number of simultaneous active named networks that can be in use at any one point in time (65536 / 64 == 1024). If your eucalyptus installation has 100 users, then each user could have at most 10 active security groups in operation at any point in time (of course, they can define as many as they wish, but can only have sets of running VMs residing in at most 10 networks). Each security group could support up to 61 instances (64 addresses minus 1 address for the subnet, broadcast, and router IPs). If your installation favors more VMs per network and fewer active security groups per user, the administrator may adjust the VNET_ADDRSPERNET parameter accordingly. Setting it to '256' would result in each active user's security group supporting up to 253 VM instances, and each of 100 users could simultaneously have 2 active security groups.

If you would like users to log in to their instances from outside the cluster/cluster front-end, you must find a set of public IP addresses, that are not in use, and allow Eucalyptus to dynamically route them to VM instances at instance boot time or dynamically at run time. For each IP address you choose, your front-end must be capable of being configured with that IP address. To test, choose some free public IP addresses and perform the following test for each one:

on the front-end:

```
ip addr add <publicIP>/32 dev <interface>
```

on some external machine representative of where users will wish to log into their VM instances:

```
ping <publicIP>
```

if this works, then dynamic IP assignment to VM instances will work. Remove the assigned address with the following command:

```
ip addr del <publicIP>/32 dev <interface>
```

Once you have compiled a list of available public IP addresses, allow Eucalyptus to use them by listing the IPs in 'eucalyptus.conf':

```
VNET_PUBLICIPS="<publicIPa> <publicIPb> ... <publicIPn>"
```

CAVEATS - When Eucalyptus is running in MANAGED mode, you cannot currently run an entire eucalyptus installation o n a single machine as this mode depends upon traffic between named networks passing through a front-end router (instead of going through the loopback device). If you wish to run Eucalyptus on a single machine (laptop), you must use SYSTEM or STATIC mode. In MANAGED mode, Eucalyptus will flush the front-end's iptables rules for both 'filter' and 'nat'. Next, it will set the default policy for the 'FORWARD' chain in 'filter' to 'DROP'. At run time, the front-end will be adding and removing rules from 'FORWARD' as users add/remove ingress rules from their active security groups. In addition, the 'nat' table will be configured to allow VMs access to the external network using IP masquerading, and will dynamically add/remove rules in the 'nat' table as users assign/unassign public IPs to VMs at instance boot or run-time. If the administrator has some rules that they wish to apply o the front-end, they should perform the following procedure on the front-end, before eucalyptus is started or while eucalyptus is not running. **WARNING** if the admin chooses to perform this operation to define special iptables rules that are loaded when Eucalyptus starts, they could inadvertently cause Eucalyptus VM networking to fail. It is suggested that you only do this only if you are completely sure that it will not interfere with the operation of Eucalyptus.

```
<use iptables to set up your iptables rules>
iptables-save > $EUCALYPTUS/var/run/eucalyptus/net/iptables-preload
```

Troubleshooting MANAGED Mode ¶

If you start an instance believe that it is running but is not available on the network, here are some things to check.

First, verify that the requirements of MANAGED mode have been met as described above (unused range of IPs, VLAN capable network, no interfering firewall rules on the nodes or front-end). Test whether you can get to the instance from the front-end using it's private address (from the range you specified). If you cannot, next, inspect the interfaces on the front-end and nodes:

on front-end:

```
ifconfig -a
```

You should see an interface '<interface>.<vlan>' with an IP address that is up and running. For instance, if may be 'eth0.10'. If it is not, check your VNET_INTERFACE parameter and inspect the eucalyptus log files for errors.

on the node:

```
brctl show
```

You should see a number of bridges called 'eucabr<vlan>', where '<vlan>' is a number that typically starts from '10'. The output should be similar (if VNET_INTERFACE="peth0") to:

```
; brctl show eucabr10
bridge name  bridge id          STP enabled     interfaces
eucabr10     8000.000c29369858  no              peth0.10
                                                vif18.0
```

If this is not the case, check your VNET_INTERFACE setting, and inspect the logfiles for details.

Back on the front-end, make sure that 'dhcpd' is running:

```
ps axww | grep <dhcpd>
```

where '<dhcpd>' is what you have set for VNET_DHCPDAEMON. Make sure that, in the output of 'ps', you see that the daemon is listening on the vlan tagged interface from above (<interface>.<vlan>). If it is not running, check the eucalyptus logs for the reason why (if the command failed, you will see this information in 'cc.log', if the daemon failed at runtime, you can inspect the reason in the daemon's output itself in 'http-cc_error_log'.

If you can access the private IP of the instance from the front-end, but public IPs are not being forwarded properly, first confirm that the user's security group is set up properly by having them run 'ec2-describe-group <group of instance>'. '<group of instance>' is set to 'default' by default or if unspecified when the instance was started. If the group has appropriate ingress rules set, check that the rules have been implemented on the front-end:

```
iptables -L <username>-<groupname>
```

If there are no rules here, check the 'cc.log' for errors applying the table rules for more insight. Next, check the 'nat' table:

```
iptables -L -t nat
```

You should see one DNAT rule for routing traffic from a public IP to the instance IP, and one SNAT rule for setting the source IP of outgoing packets from that instance. If you do not, check 'cc.log' to determine the cause.

If all of these checks pass and the instance still is experiencing network problems, please prepare the following information and send it along to the Eucalyptus discussion board:

on front-end and one representative node, capture the output of the following commands:

```
netstat -rn
ifconfig -a
brctl show
iptables-save
```

and send us 'cc.log', 'nc.log', 'httpd-cc_error_log' and 'httpd-nc_error_log'.

MANAGED-NOVLAN Mode ¶

In this mode, Eucalyptus will fully manage the local VM instance network and provides all of the networking features Eucalyptus current supports (user controllable VM firewalls (ingress rules/security groups), dynamic public IP assignment), but does not provide VM network isolation. The options in 'eucalyptus.conf' that must be configured correctly in 'MANAGED-NOVLAN' mode are as follows:

On the front-end (options annotated with a '*' may be required depending on your installation, see below for details):

```
VNET_MODE="MANAGED-NOVLAN"
VNET_INTERFACE
VNET_DHCPDAEMON
*VNET_DHCPUSER
VNET_SUBNET
VNET_NETMASK
VNET_DNS
VNET_ADDRSPERNET
*VNET_PUBLICIPS
```

On each node:

```
VNET_MODE="MANAGED-NOVLAN"
VNET_BRIDGE
```

Be advised that this mode requires that your local network/configuration conforms to certain requirements that Eucalyptus depends upon.

Requirements for MANAGED-NOVLAN mode ¶

Before using 'MANAGED-NOVLAN' mode, you must confirm that:

> 1.) there is an available range of iP addresses that is completely unused on the network (192.168..., 10....., other).
>
> 2.) you are not running a firewall on the front-end (CC) or your firewall is compatible with the dynamic changes that Eucalyptus will make to the front-end's netfilter rules.

Both of these requirements must be met before MANAGED-NOVLAN mode should be attempted. Failure to verify the above will, at least, result VM instances being unavailable on the network.

For requirement '1', choose a IP range that you know is completely unused on your network. Choose a range that is as large as possible. Typical examples are:

if the network 10.0.0.0 - 10.255.255.255 is completely unused:

```
VNET_MODE="MANAGED-NOVLAN"
VNET_SUBNET="10.0.0.0"
VNET_NETMASK="255.0.0.0"
VNET_DNS="<your DNS>"
VNET_ADDRSPERNET="128"
```

or if the network 192.168.0.0 - 192.168.255.255 is completely unused:

```
VNET_MODE="MANAGED-NOVLAN"
VNET_SUBNET="192.168.0.0"
VNET_NETMASK="255.255.0.0"
VNET_DNS="<your DNS>"
VNET_ADDRSPERNET="64"
```

You will need to carefully inspect the firewall on the front-end to make sure that it will not interfere with Eucalyptus, or vice-versa. Eucalyptus will flush the 'filter' and 'nat' tables upon boot in MANAGED-NOVLAN mode, but provides a way for the administrator to define special rules that are loaded when Eucalyptus starts (see below for details).

Configuring MANAGED-NOVLAN mode ¶

The Eucalyptus administrator must configure the front-end's 'eucalyptus.conf' first with a valid, configured ethernet device that is attached to the same physical ethernet as the Eucalyptus nodes:

```
VNET_INTERFACE="eth0"
```

Next, the admin ust ensure that there is a DHCP server binary installed on the front-end and Eucalyptus knows where it is located:

```
VNET_DHCPDAEMON="/usr/sbin/dhcpd3"
```

If your DHCP daemon binary is configured to run as 'non-root' (say, as the user 'dhcpd' as is the case in Ubuntu >= 8.10), then you must configure Eucalyptus to be aware of that user:

```
VNET_DHCPUSER="<dhcpusername>"
```

Nodes must have VNET_BRIDGE set properly. For example, with current Xen versions, this parameter (when your node's Xen bridge is 'xenbr0') is typically:

```
VNET_BRIDGE="xenbr0"
```

Once you have verified that your network configuration meets the requirements for running in MANAGED-NOVLAN mode, the rest of the configuration is fairly simple. For example, if the 192.168.0.0/16 network is free and unused on your network:

```
VNET_MODE="MANAGED-NOVLAN"
VNET_SUBNET="192.168.0.0"
VNET_NETMASK="255.255.0.0"
VNET_DNS="<your dns>"
VNET_ADDRSPERNET="64"
VNET_PUBLICIPS="<publicIPa> <publicIPb> ... <publicIPn>"
```

SUBNET, NETMASK, and DNS have been described previously. VNET_ADDRSPERNET is used to control how many VM instances may simultaneously be part of an individual user's named network (called a 'security group' in Amazon EC2). Choosing the right value for this parameter depends on how many IPs you have made available using VNET_SUBNET/VNET_NETMASK, how many VLANs your network supports simultaneously, and how many concurrent active user networks the administrator wishes to support. In the above example, there are 65536 addresses available (192.168.0.0/16). If we divide by the number of addresses per network (set to 64 above), we find the maximum number of simultaneous active named networks that can be in use at any one point in time (65536 / 64 == 1024). If your eucalyptus installation has 100 users, then each user could have at most 10 active security groups in operation at any point in time (of course, they can define as many as they wish, but can only have sets of running VMs residing in at most 10 networks). Each security group could support up to 61 instances (64 addresses minus 1 address for the subnet, broadcast, and router IPs). If your installation favors more VMs per network and fewer active security groups per user, the administrator may adjust the VNET_ADDRSPERNET parameter accordingly. Setting it to '256' would result in each active user's security group supporting up to 253 VM instances, and each of 100 users could simultaneously have 2 active security groups.

If you would like users to log in to their instances from outside the cluster/cluster front-end, you must find a set of public IP addresses, that are not in use, and allow Eucalyptus to dynamically route them to VM instances at instance boot time or dynamically at run time. For each IP address you choose, your front-end must be capable of being configured with that IP address. To test, choose some free public IP addresses and perform the following test for each one:

on the front-end:

```
ip addr add <publicIP>/32 dev <interface>
```

on some external machine representative of where users will wish to log into their VM instances:

```
ping <publicIP>
```

if this works, then dynamic IP assignment to VM instances will work. Remove the assigned address with the following command:

```
ip addr del <publicIP>/32 dev <interface>
```

Once you have compiled a list of available public IP addresses, allow Eucalyptus to use them by listing the IPs in 'eucalyptus.conf':

```
VNET_PUBLICIPS="<publicIPa> <publicIPb> ... <publicIPn>"
```

CAVEATS - When Eucalyptus is running in MANAGED-NOVLAN mode, you cannot currently run an entire eucalyptus installation on a single machine as this mode depends upon traffic between named networks passing through a front-end router (instead of going through the loopback device). If you wish to run Eucalyptus on a single machine (laptop), you must use SYSTEM or STATIC mode. In MANAGED-NOVLAN mode, Eucalyptus will flush the front-end's iptables rules for both 'filter' and 'nat'. Next, it will set the default policy for the 'FORWARD' chain in 'filter' to 'DROP'. At run time, the front-end will be adding and removing rules from 'FORWARD' as users add/remove ingress rules from their active security groups. In addition, the 'nat' table will be configured to allow VMs access to the external network using IP masquerading, and will dynamically add/remove rules in the 'nat' table as users assign/unassign public IPs to VMs at instance boot or run-time. If the administrator has some rules that they wish to apply o the front-end, they should perform the following procedure on the front-end, before eucalyptus is started or while eucalyptus is not running. **WARNING** if the admin chooses to perform this operation to define special iptables rules that are loaded when Eucalyptus starts, they could inadvertently cause Eucalyptus VM networking to fail. It is suggested that you only do this only if you are completely sure that it will not interfere with the operation of Eucalyptus.

```
<use iptables to set up your iptables rules>
iptables-save > $EUCALYPTUS/var/run/eucalyptus/net/iptables-preload
```

Managing Eucalyptus Images (1.5.1)

First, be sure to source your 'eucarc' file before running the commands below. Note that all users may upload and register images (depending on access granted to them by the Eucalyptus administrator), but only the admin user may ever upload/register kernels or ramdisks.

The latest version of EC2 API and AMI tools that we support are ec2-api-tools-1.3-30349 and ec2-ami-tools-1.3-26357. Download BOTH before proceeding, if you have not done so already.

1. Adding Images ¶

To enable a VM image as an executable entity, a user/admin must add a root disk image, a kernel/ramdisk pair (ramdisk may be optional) to Walrus and register the uploaded data with Eucalyptus. Each is added to Walrus and registered with Eucalyptus separately, using three EC2 commands. The following example uses the test image that we provide. Unpack it to any directory:

Add the kernel to Walrus, and register it with Eucalyptus (**WARNING**: your bucket names must not end with a slash!):

```
ec2-bundle-image -i <kernel file> --kernel true
ec2-upload-bundle -b <kernel bucket> -m /tmp/<kernel file>.manifest.xml
ec2-register <kernel-bucket>/<kernel file>.manifest.xml
```

Next, add the root filesystem image to Walrus:

```
ec2-bundle-image -i <vm image file>
ec2-upload-bundle -b <image bucket> -m /tmp/<vm image file>.manifest.xml
ec2-register <image bucket>/<vm image file>.manifest.xml
```

Our test kernel does not require a ramdisk to boot. If the administrator would like to upload/register a kernel/ramdisk pair, the procedure is similar to the above:

```
ec2-bundle-image -i <initrd file> --ramdisk true
ec2-upload-bundle -b <initrd bucket> -m <initrd file>.manifest.xml
ec2-register <initrd bucket>/<initrd file>.manifest.xml
```

2. Associating kernels and ramdisks with instances ¶

There are three ways that one can associate a kernel (and ramdisk) with a VM instance.

1. A user may associate a specific kernel/ramdisk identifier with an image at the 'ec2-bundle-image' step

   ```
   ec2-bundle-image -i <vm image file> --kernel <eki-XXXXXXXX> --ramdisk <eri-XXXXXXXX>
   ```

2. A user may choose a specific kernel/ramdisk at instance run time as an option to 'ec2-run-instances'

   ```
   ec2-run-instances <emi-XXXXXXXX> --kernel <eki-XXXXXXXX> --ramdisk <eri-XXXXXXXX>
   ```

3. The administrator can set 'default' registered kernel/ramdisk identifiers that will be used if a kernel/ramdisk is unspecified by either of the above options. This is accomplished by logging in to the administrative interface (https://your.cloud.server:8443), clicking on the 'Configuration' tab and adding an <eki-xxxxxxxx> and optionally an <eri-xxxxxxxx> as the defaults kernel/ramdisk to be used.

3. Deleting Images ¶

In order to delete an image, you must first de-register the image:

```
ec2-deregister <emi-XXXXXXXX>
```

Then, you can remove the files stored in your bucket. Assuming you have sourced your 'eucarc' to set up EC2 client tools:

```
ec2-delete-bundle -a $EC2_ACCESS_KEY -s $EC2_SECRET_KEY --url $S3_URL -b <bucket> -p <file prefix>
```

If you would like to remove the image and the bucket, add the '--clear' option:

```
ec2-delete-bundle -a $EC2_ACCESS_KEY -s $EC2_SECRET_KEY --url $S3_URL -b <bucket> -p <file prefix> --clear
```

Examples ¶

Following is an example using the 'ttylinux' image for Xen:

```
cd $EUCALYPTUS_SRC/eucalyptus-src-deps
tar zxvf euca-ttylinux.tgz

ec2-bundle-image -i ttylinux/vmlinuz-2.6.16.33-xen --kernel true
ec2-upload-bundle -b kernel-bucket -m /tmp/vmlinuz-2.6.16.33-xen.manifest.xml
ec2-register kernel-bucket/vmlinuz-2.6.16.33-xen.manifest.xml

ec2-bundle-image -i ttylinux/ttylinux.img
ec2-upload-bundle -b image-bucket -m /tmp/ttylinux.img.manifest.xml
ec2-register image-bucket/ttylinux.img.manifest.xml
```

Next is an example using the Ubuntu pre-packaged image that we provide using the included KVM compatible kernel/ramdisk (a Xen compatible kernel/ramdisk is also included). See this page to get more pre-packaged images.

```
tar zxvf euca-ubuntu-9.04-x86_64.tar.gz

ec2-bundle-image -i euca-ubuntu-9.04-x86_64/kvm-kernel/vmlinuz-2.6.28-11-generic --kernel true
ec2-upload-bundle -b ubuntu-kernel-bucket -m /tmp/vmlinuz-2.6.28-11-generic.manifest.xml
ec2-register ubuntu-kernel-bucket/vmlinuz-2.6.28-11-generic.manifest.xml
(set the printed eki to $EKI)

ec2-bundle-image -i euca-ubuntu-9.04-x86_64/kvm-kernel/initrd.img-2.6.28-11-generic --ramdisk true
ec2-upload-bundle -b ubuntu-ramdisk-bucket -m /tmp/initrd.img-2.6.28-11-generic.manifest.xml
ec2-register ubuntu-ramdisk-bucket/initrd.img-2.6.28-11-generic.manifest.xml
(set the printed eri to $ERI)

ec2-bundle-image -i euca-ubuntu-9.04-x86_64/ubuntu.9-04.x86-64.img --kernel $EKI --ramdisk $ERI
ec2-upload-bundle -b ubuntu-image-bucket -m /tmp/ubuntu.9-04.x86-64.img.manifest.xml
ec2-register ubuntu-image-bucket/ubuntu.9-04.x86-64.img.manifest.xml
```

Now, the newly uploaded image(s) should be ready to start using (see User's Guide for more information on using Eucalyptus).

Eucalyptus Management (1.5.1)

This part of the Administrator's Guide describes tasks that can be performed on a completed Eucalyptus installation, whether it was installed from source, from an RPM package, or from a Rocks roll.

1. Administrator's environment ¶

Currently, some administrative tasks can only be done through the command-line interface. When running commands, environment variable EUCALYPTUS must be pointing to the root of Eucalyptus installation for them to work properly.

If you installed Eucalyptus using RPMs, the value of $EUCALYPTUS is **/opt/eucalyptus**

```
export EUCALYPTUS=/path/to/eucalyptus
```

Adding this command to administrator's shell startup script, such as .profile, is probably a good idea.

2. Image Management ¶

To use Eucalyptus, Xen images must be added and registered with the system. We have a document detailing the steps of this process in Image Management.

3. Node Management ¶

Once you have a running Eucalyptus system you can add and remove nodes (systems running Node Controllers) by editing `$EUCALYPTUS/etc/eucalyptus/eucalyptus.conf` and editing the list of `NODES`:

```
NODES="vm-container-0-0 vm-container-0-1"
```

Whenever you add new nodes into the system, be sure to immediately propagate cryptographic keys to them as follows:

```
${EUCALYPTUS}/usr/sbin/euca_sync_key
```

4. User Management ¶

4.1 User sign-up ¶

Users interested in joining the cloud should be directed to the front-end Web page (note the **https** prefix!):

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

As soon as the administrator logs in for the first time and enters the email address to be used for application requests, thus activating the Web site for use by others, the login box of the Web site will have an "Apply for account" link underneath it. After a user fills out the application form, an email is sent to the administrator, containing two URLs, one for accepting and one for rejecting the user.

Note that there is no authentication performed on the people who fill out the form. It is up to the administrator to perform this authentication! The only "guarantee" the administrator has is that the account will not be active unless the person who requested the account (and, hence, knows the password) can read email at the submitted address. Therefore, if the administrator is willing to give the account to the person behind the email address, it is safe to approve the account. Otherwise, the administrator may use the additional information submitted (such as the telephone number, project PI, etc.) to make the decision.

Accepting or rejecting a signup request causes an email message to be sent to the user who made the request. In the case of an acceptance notification, the user will see a link for activating the account. Before activating the account, the user will have to log in with the username and password that they chose at signup.

4.2 Adding users ¶

Users can be added by the administrator explicitly by logging into the Eucalyptus web interface, as an administrative user, clicking the 'Users' tab, clicking on the 'Add User' button, and filling out the same user form that a user would fill out if they applied themselves. The user will be automatically 'approved' using this method, but their account will not be active until the user clicks the link that is sent via email similar to the above method.

4.3 Managing users ¶

If the administrator wishes to disable or delete a user, they can do so through the web interface, as an administrative user, clicking the 'Users' tab, and clicking either the 'disable' or 'delete' link respectively.

Eucalyptus Troubleshooting (1.5.1)

Eucalyptus cloud admins are encouraged to consult the Known Bugs page before diving into the investigation of unexpected behavior.

1. Restarting ¶

If an administrator ever needs to stop/start a Eucalyptus front-end because of a configuration change, or if the machine on which the front-end is running reboots unexpectedly, the administrator must terminate all running instances in the system before bringing Eucalyptus back online. (It is possible to restart the cloud controller using `/etc/init.d/eucalyptus restart` on the head-node without affecting the rest of the system, but then some of the configuration is not reloaded. Doing `stop` followed by `start` on the head-node will reload the configuration, but will also destroy the virtual network setup among the running VMs, making them inaccessible.)

If the restart is planned, the administrator can use the client tools to terminate all users instances before stopping/reconfiguring/starting Eucalyptus. If the restart was unplanned (front-end machine crashes), the admin can try to start Eucalyptus and immediately terminate all running instances, or can manually stop all eucalyptus components, destroy all running Xen instances using 'xm shutdown' or 'xm destroy' on the nodes, and starting all Eucalyptus components.

2. Diagnostics ¶

Installation/Discovering resources ¶

If something is not working right with your Eucalyptus installation, the best first step (after making sure that you have followed the installation/configuration/networking documents faithfully) is to make sure that your cloud is up and running, that all of the components are communicating properly, and that there are resources available to run instances. After you have set up and configured Eucalyptus, set up your environment properly with your admin credentials, and use the following command to see the 'status' of your cloud:

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

You should see output similar to the following:

```
AVAILABILITYZONE        cluster <hostname of your front-end>
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small     0128 / 0128   1    128    10
AVAILABILITYZONE        |- c1.medium    0128 / 0128   1    256    10
AVAILABILITYZONE        |- m1.large     0064 / 0064   2    512    10
AVAILABILITYZONE        |- m1.xlarge    0064 / 0064   2   1024    20
AVAILABILITYZONE        |- c1.xlarge    0032 / 0032   4   2048    20
AVAILABILITYZONE        |- <node-hostname-a>        certs[cc=true,nc=true] @ Sun Jan 04 15:13:30 PST 2009
AVAILABILITYZONE        |- <node-hostname-b>        certs[cc=true,nc=true] @ Sun Jan 04 15:13:30 PST 2009
AVAILABILITYZONE        |- <node-hostname-c>        certs[cc=true,nc=true] @ Sun Jan 04 15:13:30 PST 2009
AVAILABILITYZONE        |- <node-hostname-d>        certs[cc=true,nc=true] @ Sun Jan 04 15:13:30 PST 2009
AVAILABILITYZONE        |- <node-hostname-e>        certs[cc=true,nc=true] @ Sun Jan 04 15:13:30 PST 2009
AVAILABILITYZONE        |- <node-hostname-f>        certs[cc=true,nc=true] @ Sun Jan 04 15:13:30 PST 2009
...
```

Next, the administrator should consult the Eucalyptus logfiles. On each machine running a Eucalyptus component, the logfiles are located in:

```
$EUCALYPTUS/var/log/eucalyptus/
```

On the front-end, the Cloud Controller (CLC) logs primarily to 'cloud-output.log' and 'cloud-debug.log'. Consult these files if your client tool (ec2 API tools) output contains exception messages, or if you suspect that none of your operations are ever being executed (never see Xen activity on the nodes, network configuration activity on the front-end, etc.).

The Cluster Controller (CC) also resides on the front-end, and logs to 'cc.log' and 'httpd-cc_error_log'. Consult these logfile in general, but especially if you suspect there is a problem with networking. 'cc.log' will contain log entries from the CC itself, and 'httpd-cc_error_log' will contain the STDERR/STDOUT from any external commands that the CC executes at runtime.

A Node Controller (NC) will run on every machine in the system that you have configured to run VM instances. The NC logs to 'nc.log' and 'httpd-nc_error_log'. Consult these files in general, but especially if you believe that there is a problem with VM instances actually running (i.e., it appears as if instances are trying to run - get submitted, go into 'pending' state, then go into 'terminated' directly - but fail to stay running).

Node Controller troubleshooting ¶

- If nc.log reports "Failed to connect to hypervisor," xen/kvm + libvirt is not functioning correctly.

Walrus troubleshooting ¶

- "ec2-upload-bundle" will report a "409" error when uploading to a bucket that already exists. This is a known compatibility issue when using ec2 tools with Eucalyptus. The workaround is to use ec2-delete-bundle with the "--clear" option to delete the bundle and the bucket, before uploading to a bucket with the same name, or to use a different bucket name.
- When using "ec2-upload-bundle," make sure that there is no "/" at the end of the bucket name.

Block storage troubleshooting ¶

- Unable to attach volumes when the front end and the NC are running on the same machine. This is a known issue with ATA over Ethernet (AoE). AoE will not export to the same machine that the server is running on. The workaround is to run the front end and the node controller on different hosts.
- Volume ends up in "deleted" state when created, instead of showing up as "available." Look for error messages in $EUCALYPTUS/var/log/eucalyptus/cloud-error.log. A common problem is that ATA-over-Ethernet may not be able to export the created volume (this will appear as a "Could not export..." message in cloud-error.log). Make sure that "VNET_INTERFACE" in eucalyptus.conf on the front end is correct.
- Failure to create volume/snapshot. Make sure you have enough loopback devices. If you are installing from packages, you will get a warning. On most distributions, the loopback driver is installed as a module. The following will increase the number of loopback devices available,

  ```
  rmmod loop ; modprobe loop max_loop=256
  ```
- If block devices do not automatically appear in your VMs, make sure that you have the "udev" package installed.
- I'm running gentoo. I get "which: no vblade in ((null))." Try compiling "su" without pam.

Eucalyptus User's Guide (1.5.1)

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.)

Getting Started Using Eucalyptus (1.5.1)

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. 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/*
```

3. Install EC2 command-line tools ¶

Download the EC2 command-line tools from Amazon.

```
wget http://s3.amazonaws.com/ec2-downloads/ec2-api-tools-1.3-30349.zip
unzip ec2-api-tools-1.3-30349.zip
```

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

Set the following environment variables and source the 'eucarc' file that came with your credentials to set other crucial Eucalyptus environment variables:

```
export EC2_HOME=/path/to/installed/ec2-commandline-tools
export PATH=$PATH:$EC2_HOME/bin
source ~/.euca/eucarc
```

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.

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 >mykey.private
('mykey' is the name for the key in Eucalyptus, 'mykey.private' is the file to be used by ssh)

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)
```

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 'ec2-describe-addresses' as a reminder), and assign it to your running instance:

Allow 'ssh' connections from the Internet:

```
ec2-authorize default -P tcp -p 22 -s 0.0.0.0/0
```

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

```
ec2-allocate-address
```

Associate an allocated IP with your running instance:

```
ec2-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. You may log into it with the SSH key that you created:

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

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, depending on how the administrator has configured Eucalyptus, not all tools/operations are necessarily supported (security groups/elastic IPs). Consult your administrator for more information.

Interacting with Walrus (1.5.1)

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.

Interacting with Block Storage (1.5.1)

The Block Storage Service in Eucalyptus is interface compatible with Amazon's Elastic Block Store. You can therefore use ec2 commands to interact with Eucalyptus' Block Storage. You will need to download and install EC2 API tools in order to use Block Storage.

The following operations are possible,

1. Creating volumes

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

```
ec2-create-volume -s <size> -z <zone>
```

where <size> is the size in GB and <zone> is the availability zones you wish to create the volume in (use ec2-describe-availability-zones to discover zones).

For instance,

```
ec2-create-volume -s 1 -z myzone
```

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

To create a volume from a snapshot,

```
ec2-create-volume --snapshot <snapshot id> -z <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,

```
ec2-create-volume --snapshot snap-EF4323 -z myzone
```

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

2. Query the status of volumes

```
ec2-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 ec2-run-instances). You may attach a volume to only one instance at a time.

```
ec2-attach-volume <volume id> -i <instance id> -d <local device name>
```

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,

```
ec2-attach-volume vol-FG6578 -i i-345678 -d /dev/sdb
```

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

4. Detaching a volume

```
ec2-detach-volume <volume id>
```

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

For instance,

```
ec2-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

```
ec2-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.

```
ec2-create-snapshot <volume id>
```

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

For instance,

```
ec2-create-snapshot vol-GH4342
```

will snapshot the volume "vol-GH4342"

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

7. Querying the status of snapshots

```
ec2-describe-snapshots
```

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

8. Deleting a snapshot

```
ec2-delete-snapshot <snapshot id>
```

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

Using Pre-packaged Images With Eucalyptus 1.5.1

To help get you started with Eucalyptus, we have provided links to pre-packaged virtual machines that are ready to run in your

Eucalyptus cloud. Clicking the link will download a package that contains a VM image (*.img), a Xen compatible kernel/ramdisk pair (xen-kernel/vmlinuz* and xen-kernel/initrd*) and a KVM compatible kernel/ramdisk pair (kvm-kernel/vmlinuz* and kvm-kernel/initrd*). Once you have downloaded an image, you can bundle, upload and register it for use in your Eucalyptus cloud. Please refer to this guide for more instructions.

- **Ubuntu 9.04 64bit**
- **CentOS 5.3 64bit**
- **Debian 5.0 64bit**
- **Fedora 10 64bit**

Once you've selected and downloaded the image(s) you plan to use, visit the Eucalyptus Image Management guide for details on how to bundle, upload and register the images with your Eucalyptus cloud.

Known problems with Eucalyptus 1.5.1

- The file `axis2c.log` contains errors:

  ```
  [error] rampart_handler_util.c(241) [rampart][rampart_handler_utils] 0 parameter is not set.
  [error] error.c(94) OXS ERROR [x509.c:284 in openssl_x509_get_subject_key_identifier] oxs defualt error , The extenension index of NID_subject_key_identifier is not valid
  ```

  These errors are benign and can be ignored.

- 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.

ChangeLog (1.5.1)

Version 1.5.1 (2009-05-08) ¶

- Elastic Block Store (EBS) support (volumes & snapshots)
- Walrus improvements:
  - Support for groups in ACLS
  - Fixed issues with meta data support
  - Web browser form-based uploads via HTTP POST
  - Object copying
  - Query string authentication
  - Support for arbitrary key names
  - Compressed image downloads and fixes to image caching
  - Reduced memory requirement
- Network improvement: new `MANAGED-NOVLAN` mode
- Node-side improvements:
  - Support for the KVM hypervisor
  - Compression & retries on Walrus downloads
  - Reworked caching (now with configurable limit)
- Web UI improvements:
  - Cloud registration with Rightscale (from admin's 'Credentials' tab)
  - New configuration options for Walrus
  - Better screening of usernames
  - Fixed account confirmation glitches
- Building and installation improvements
  - Better Java installation checking
  - New command-line administration: `euca_conf -addcluster ... -addnode ...`
  - Non-root user deployment of Eucalyptus
  - Binary packages for more distributions (Ubuntu et al)

Version 1.4 (2009-01-05) ¶

- Added new networking subsystem that no longer depends on VDE
- Added support for elastic IP assignment and security using the 'MANAGED' networking mode
- Cluster controller scheduling policy can now be configured in eucalyptus.conf (ROUNDROBIN and GREEDY currently supported)
- Cluster controller now handles concurrent requests (no longer have to restrict apache to allow only one connection at a time)
- Added Walrus: an Amazon S3 interface compatible storage manager. Walrus handles storage of user data as well as filesystem images, kernels, and ramdisks.
- Node Controller improvements:
  - Retrieval of images from Walrus instead of NFS-mounted file system
  - New caching and cleanup code, including start-time integrity checks
  - On-the-fly script-based generation of libvirt XML configuration
  - Script-based discovery of node resources (no assumptions about stat)

  - MAX_CORES overrides actual number of cores both down and up
  - Moved libvirt errors to nc.log and suppressed harmless ones
  - Serialized some Xen invocations to guard against non-determinism
  - Added proper swap creation, also "ephemeral" disk support
  - More robust instance state reporting to Cluster Controller
- Web interface improvements:
  - Added cloud/Walrus configuration, including clusters and VM types
  - Revamped 'credentials' tab with new options to edit user information and hide/show "secret" strings
  - Editing of user information for the administrator, including confirmation dialog for deletion of users
  - User-initiated password recovery mechanism
  - Fixed a couple of bugs: ' ' in username, spurious double additions
- Cloud Controller:
  - User, Cluster, and System keys are now stored in PKCS12 keystores and have moved to var/eucalyptus/keys
  - Configuration is handled entirely through the Web interface
  - Clusters dynamically added/started/stopped
  - Webservices operations complete up to EC2 2008-05-05 (w/o EBS):
  - "Elastic IP" address support
  - Image registration and attribute manipulation
  - GetConsole and RebootInstances support
  - Working Security Groups support for clusters in MANAGED network mode
  - See website for additional details, extensions, and caveats: http://eucalyptus.cs.ucsb.edu/wiki/API_v1.4
  - Instance Metadata service (including userData)
  - Workaround to use Amazon's tools for registering kernels & ramdisks
- Revamped logging throughout, with five levels a la log4j
- More standard build procedure: configure; make; make install
- More robust start-time checking

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)