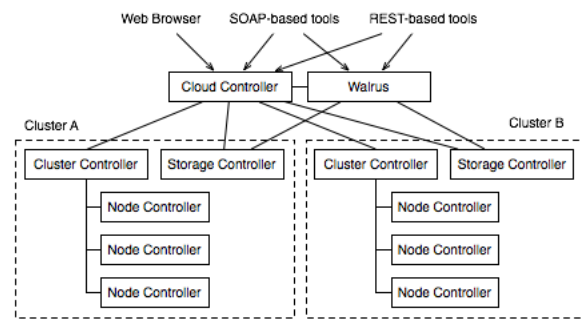


Eucalyptus Administrator's Guide (2.0)

This guide is for anyone wishing to install, configure, and manage Eucalyptus on their resources—from a set of clusters in a data center to a personal laptop. For help using an existing Eucalyptus installation, please see the Eucalyptus 2.0 User's Guide.

Installing Eucalyptus 2.0



A Eucalyptus cloud setup consists of five types of components. The *cloud controller* (CLC) and "Walrus" are top-level components, with one of each in a cloud installation. The cloud controller is a Java program that offers EC2-compatible SOAP and "Query" interfaces, as well as a Web interface to the outside world. In addition to handling incoming requests, the cloud controller performs high-level resource scheduling and system accounting. Walrus, also written in Java, implements bucket-based storage, which is available outside and inside a cloud through S3-compatible SOAP and REST interfaces.

Top-level components can aggregate resources from multiple clusters (i.e., collections of nodes sharing a LAN segment, possibly residing behind a firewall). Each cluster needs a *cluster controller* (CC) for cluster-level scheduling and network control and a "storage controller" (SC) for EBS-style block-based storage. The two cluster-level components would typically be deployed on the head-node of a cluster. Finally, every node with a hypervisor will need a *node controller* (NC) for controlling the hypervisor. CC and NC are written in C and deployed as Web services inside Apache; the SC is written in Java. Communication among these components takes place over SOAP with WS-security.

Many instructions in this guide refer to a *single-cluster installation*, in which all components except NC are co-located on one machine, which we refer to as **front-end**. All other machines, running only NCs, will be referred to as **nodes**. In more advanced configurations, such as those with multiple CCs or with Walrus deployed separately, the *front-end* will refer to just the machine running the CLC.

Eucalyptus can be installed from source or using a set of packages (RPM and DEB). Installing Eucalyptus from source is a more general method and should work on practically any Linux system; installing from packages is easier but will only work on the distributions that we support. Eucalyptus currently supports installation from binary packages on these Linux distributions:

- CentOS 5
- Debian squeeze
- OpenSUSE 11
- Fedora 12

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

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

Installing Eucalyptus from Source (2.0)

1. Prerequisites

What follows is a comprehensive list of dependencies that must be satisfied before building Eucalyptus. Here we provide examples for installing prerequisite packages for the distributions supported by Eucalyptus. **NOTE** - If you are upgrading from a Eucalyptus 1.6.2 or older installation, please consult the Upgrade Documentation for instructions that will explain how to preserve user account information, images, volumes and snapshots.

Prerequisites 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

Prerequisites for running Eucalyptus

There are a few different Eucalyptus components that run on either the 'front-end or '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.
- **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 Network Configuration 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)
 - vtun package, for multi-cluster configurations
 - 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)
 - aoe-tools 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**
- Disk utilities: dd and parted commands
- Dependencies for network support differ depending on the mode used (see Eucalyptus Network configuration 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 libvirt-d, depending on hypervisor configuration)

All Eucalyptus components

- You *must* be **root** to install and start Eucalyptus components (by default they will run under a different user after start). This document assumes that all commands will be executed as root.

Attention CentOS users: The version of OpenJDK that is bundled with CentOS-5 cannot compile the version of GWT that comes with Eucalyptus as a dependency. You will need to install JDK 1.6.0 "manually". We use Sun's JDK, which can be found at <http://java.sun.com/javase/downloads/index.jsp>. Be sure to set your JAVA_HOME and PATH properly before running the Eucalyptus 'configure' script.

Distribution-specific examples

What follows is a superset of all packages necessary for building and running Eucalyptus on each supported distribution:

- For **Opensuse 11.2**, download and install RPMs the appropriate OpenSUSE RPM dependency package from the Eucalyptus website, then run the following command to install all required dependency packages:

```
zypper -n install curl bzip python-paramiko make gcc ant apache2 apache2-prefork 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
```

- For **Ubuntu 10.04**, run the following command to install all required dependency packages:

```
apt-get install bzip2 gcc make apache2-threaded-dev ant openjdk-6-jdk\
libvirt-dev libcurl4-openssl-dev dhcp3-server vblade apache2 unzip curl vlan\
bridge-utils libvirt-bin kvm vtun
```

- For **CentOS 5** and **Fedora 12**, download and install RPMs the appropriate CentOS or Fedora RPM dependency package from the Eucalyptus website, then 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 libxml2 libxml2-devel gnutls gnutls-devel xen-devel libgcrypt-devel zlib-devel perl-Conver
```

- For **Debian**, run the following command to install all required dependency packages:

```
apt-get install gcc make apache2-threaded-dev ant openjdk-6-jdk\
libvirt-dev libcurl4-dev dhcp3-server vblade apache2 unzip curl vlan\
bridge-utils libvirt-bin kvm sudo vtun
```

Please, consult the distribution-specific pages for detailed installation instructions.

Tools for interacting with Eucalyptus

To interact with Eucalyptus, you need to install EC2-compatible command-line tools. The instructions in Eucalyptus documentation rely on the euca2ools command-line tools distributed by the Eucalyptus Team. Many other third-party tools can also be used for some of the tasks, as described on the ecosystem page.

2. Download Eucalyptus and supplied dependencies

In what follows substitute the desired version of Eucalyptus (e.g., 2.0.3) for \$VERSION. You can do this manually or by setting a shell variable:

```
export VERSION=2.0.3
```

Download either:

- eucalyptus-\$VERSION-src.tar.gz (Eucalyptus source with included java libraries)

or

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

and for both

- eucalyptus-\$VERSION-src-deps.tar.gz (Eucalyptus C library dependency packages)

All packages can be found on the Eucalyptus Web site:

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

Unpack the Eucalyptus source:

```
tar zxvf eucalyptus-$VERSION-src.tar.gz
```

Now you should have a directory eucalyptus-\$VERSION. 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-$VERSION
export EUCALYPTUS_SRC=`pwd`
export EUCALYPTUS=/opt/eucalyptus
```

3. Build Dependencies

To install Eucalyptus, you need to build packages that Eucalyptus depends on, which we provide in the above-mentioned package eucalyptus-\$VERSION-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 $EUCALYPTUS_SRC
tar zxvf ../eucalyptus-$VERSION-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

To compile Axis2/C, you will need to locate development headers for Apache and for APR. On some distributions (e.g., Ubuntu and Debian) the following values should work:

```
export APACHE_INCLUDES=/usr/include/apache2
export APR_INCLUDES=/usr/include/apr-1.0
```

On CentOS 5, the headers should be in the following location:

```
export APACHE_INCLUDES=/usr/include/httpd/
export APR_INCLUDES=/usr/include/apr-1/
```

while on OpenSuse 11 you may find them at:

```
export APACHE_INCLUDES=/usr/include/apache2/
export APR_INCLUDES=/usr/include/apr-1/
```

With the two environment variables set, you can build and install Axis2/C as follows:

```
export AXIS2C_HOME=$EUCALYPTUS/packages/axis2c-1.6.0
cd $EUCALYPTUS_SRC/eucalyptus-src-deps/
tar zxvf axis2c-src-1.6.0.tar.gz
cd axis2c-src-1.6.0
CFLAGS="-w" ./configure --prefix=${AXIS2C_HOME} --with-apache2=$APACHE_INCLUDES --with-apr=$APR_INCLUDES --enable-multi-thread=no
make ; make install
```

c. Rampart/C

```
export AXIS2C_HOME=$EUCALYPTUS/packages/axis2c-1.6.0
export LD_LIBRARY_PATH=${AXIS2C_HOME}/lib:$LD_LIBRARY_PATH
cd $EUCALYPTUS_SRC/eucalyptus-src-deps/
tar zxvf rampartc-src-1.3.0-0euca2.tar.gz
cd rampartc-src-1.3.0
./configure --prefix=${AXIS2C_HOME} --enable-static=no --with-axis2=${AXIS2C_HOME}/include/axis2-1.6.0
make ; make install
```

Next, change the following in \$AXIS2C_HOME/axis2.xml. In the 'inflow' section, change:

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

to

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

In the 'outflow' section, change:

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

to

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

4. Building Eucalyptus

First, make sure JAVA_HOME is defined. For example, on Centos 5:

```
export JAVA_HOME="/usr/lib/jvm/java-openjdk/"
export JAVA="$JAVA_HOME/jre/bin/java"
```

then, build Eucalyptus:

```
cd $EUCALYPTUS_SRC
./configure --with-axis2=$EUCALYPTUS/packages/axis2-1.4 --with-axis2c=$EUCALYPTUS/packages/axis2c-1.6.0 --enable-debug --prefix=$EUCALYPTUS
make ; make install
```

5. Deploying Eucalyptus

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 (although not all software components are required on all nodes, it is simpler to just mirror everything and selectively enable components via start-up scripts). 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 $EUCALYPTUS/ root@{node-host-1}:$EUCALYPTUS/
rsync -a $EUCALYPTUS/ root@{node-host-1}:$EUCALYPTUS/
...
```

On installations without a root user, such as Ubuntu, it may be easier to pull the software from each node one at a time:

```
node1# rsync -a {user}@{front-end}:$EUCALYPTUS/ $EUCALYPTUS/
node2# rsync -a {user}@{front-end}:$EUCALYPTUS/ $EUCALYPTUS/
...
```

NOTE: Installing Eucalyptus in the same directory on all nodes will make it easier to manage it, so we strongly advise you to do so.

6. Configure Hosts

a. Set up a 'eucalyptus' user on all machines

Eucalyptus will run as regular user on your systems, which you'll need to add before running Eucalyptus (we will use `eucalyptus`) on **all machines**. For most distributions, this task is accomplished by running the command:

```
useradd eucalyptus
```

For OpenSUSE, use:

```
groupadd eucalyptus
useradd eucalyptus -m -g eucalyptus
```

b. Configure your hypervisor

Ensure that this user can control your hypervisor through libvirt on **all compute nodes**. On some distributions, this can be accomplished by adding `eucalyptus` to group `libvirt` OR `libvirt` in file `/etc/group`. Please consult the documentation for libvirt on your distribution for instructions. See Hypervisor Configuration for more detailed information.

c. Configure your network

Eucalyptus provides several networking modes from which to choose, depending on your local network setup, capabilities, and the networking features that you wish to take advantage of within Eucalyptus. Most networking options require that, on your node controllers, the primary interface is configured to be a bridge (this is the default configuration with some distribution's Xen hypervisor configuration). See Network Configuration for more information and set-up instructions. Once you have decided which network mode you will be using, you may be required to set up ethernet bridges on Eucalyptus component machines. Example bridge configuration steps can be found [here](#).

d. Configure Eucalyptus components

On your **compute nodes**, create a local directory where VM images are to be placed temporarily while VMs are running (images will be cached under the same path, too). It is important that the directory is empty as *everything in it will be removed*. Be sure to pick a partition with ample disk space as VM images can be large. We use `/usr/local/eucalyptus` in the example below.

Place the mandatory parameters (including the type of hypervisor you plan to use) into the configuration file and set up the permissions on Eucalyptus files appropriately on **all nodes**. Both tasks can be accomplished with flags to `euca_conf` tool:

- **-d** specifies the root of Eucalyptus installation (`$EUCALYPTUS`)
- **--hypervisor** specifies the hypervisor ('xen' or 'kvm')
- **--instances** specifies where, on compute nodes, instance files will be stored
- **--user** specifies the user that you created for running Eucalyptus
- **--setup** invokes the first-time setup tasks

```
$EUCALYPTUS/usr/sbin/euca_conf -d $EUCALYPTUS --hypervisor kvm --instances /usr/local/eucalyptus --user eucalyptus --setup
```

e. Distribution-specific post configuration steps

Some linux distributions require that the admin perform a few extra steps before bringing up Eucalyptus. This section details some of these steps:

For Ubuntu, apparmor needs to be configured to allow dhcpd3 to write to the filesystem. Add the following lines to `/etc/apparmor.d/usr.sbin.dhcpd3`:

```
/opt/eucalyptus/var/run/eucalyptus/net/ r,
/opt/eucalyptus/var/run/eucalyptus/net/** r,
/opt/eucalyptus/var/run/eucalyptus/net/*.pid lrw,
/opt/eucalyptus/var/run/eucalyptus/net/*.leases* lrw,
/opt/eucalyptus/var/run/eucalyptus/net/*.trace lrw,
```

where you substitute `/opt/eucalyptus` with the path to where you have chosen to install Eucalyptus. Then, restart apparmor (NOTE: sometimes changes don't take effect right away - either wait or reboot the system to be sure):

```
/etc/init.d/apparmor stop
/etc/init.d/apparmor start
```

Also, since Ubuntu DHCP daemon is configured to run as 'dhcpd' and not root, ensure that the following two variables are set as follows in the `$EUCALYPTUS/etc/eucalyptus/eucalyptus.conf` file **on the Cluster head-node**:

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

```
VNET_DHCPUSER="dhcpd"
```

At this point you should be ready to start Eucalyptus processes on all nodes but before doing so you may want to configure the Eucalyptus network: you can read more about it at Network Configuration.

f. Configure your startup scripts

If you want to have eucalyptus started automatically when your machines are (re)booted, you can add the following symlinks on the appropriate hosts: add `eucalyptus-cloud` on the Cloud head-node, add `eucalyptus-cc` on the Cluster head-node(s), and add `eucalyptus-nc` on the compute node(s)

```
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
```

7. Running Eucalyptus

Eucalyptus comes with the `euca_conf` script for configuring Eucalyptus. For some requests it modifies the configuration file located in '`$EUCALYPTUS/etc/eucalyptus/eucalyptus.conf`' (which can also be edited manually), for other requests it modifies the database maintained by the Cloud Controller (much of that functionality is duplicated in the Web interface, to be described later).

In addition to modifying the configuration, `euca_conf` attempts to synchronize x509 credentials across the nodes of a Eucalyptus installation by relying on `rsync` and `scp`. *We highly recommend setting up password-less SSH access for the `root` user across all nodes of your Eucalyptus installation* (otherwise, `euca_conf` will prompt you for remote system passwords).

As explained in the overview, a Eucalyptus installation consists of five types of components: cloud controller (CLC), Walrus, cluster controller (CC), storage controller (SC), and the node controller(s) (NCs). In following instructions we assume that all components except the NCs are co-located on one machine that we refer to as the *front end* and that NCs run on one or more other machines referred to as *compute nodes*.

To run 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`. For more information on Eucalyptus log files and error reports, please see Troubleshooting Eucalyptus.

Next, inspect the contents of `$EUCALYPTUS/etc/eucalyptus/eucalyptus.conf` carefully, on each machine, to make sure that the settings are appropriate for your environment. Once you have confirmed that everything is configured properly, enable the cloud services on the front-end:

```
# enable services on the front-end
$EUCALYPTUS/usr/sbin/euca_conf -d $EUCALYPTUS --setup
$EUCALYPTUS/usr/sbin/euca_conf -d $EUCALYPTUS --enable cloud --enable walrus --enable sc
```

BEFORE STARTING EUCALYPTUS SERVICES! If you are upgrading from Eucalyptus 1.6.2 to Eucalyptus 2.0, return now to the Upgrade Instructions, and proceed with running the upgrade commands for the front-end and nodes as specified. If you are performing a first-time installation, you may proceed with the following steps and start Eucalyptus services:

Start each component on the appropriate host.

```
# start enabled front-end services
$EUCALYPTUS/etc/init.d/eucalyptus-cloud start

# start the cluster controller
$EUCALYPTUS/etc/init.d/eucalyptus-cc start
```

And on each of the compute nodes run:

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

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

NOTE: if you later decide to make changes to `$EUCALYPTUS/etc/eucalyptus/eucalyptus.conf` that will effect the cluster-controller, make sure to use the 'cleanstart', 'cleanstop', and/or 'cleanrestart' directives to the init scripts (as opposed to start/stop/restart). This will both remove all existing CC state, and will cause it to re-read the configuration file.

Installing Eucalyptus 2.0 from binary packages

You can install Eucalyptus 2.0.* from binary packages on these Linux operating systems:

- Centos 5.5
- openSuse 11.2
- Debian "squeeze"
- Fedora 12

Installing Eucalyptus (2.0) on Centos 5.5

Eucalyptus can be installed on CentOS 5 from source or by using binary RPM packages. This document details the steps required to install Eucalyptus from RPMs. In what follows, the value of **\$VERSION** must be set to the version of Eucalyptus you wish to install. For example, you can set the value to 2.0.3 using bash:

```
export VERSION=2.0.3
```

Notice: Before you begin, please ensure that you have an up-to-date CentOS installation on your target machine(s).

Prerequisites

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

- Front-end, node(s), and client machine system clocks are synchronized (e.g., using NTP).

```
yum install -y ntp
ntpdate pool.ntp.org
```
- Front end needs java, command to manipulate a bridge, and the binaries for dhcp server (do not configure or run dhcp server on the CC):

```
yum install -y java-1.6.0-openjdk ant ant-nodeps dhcp \
bridge-utils perl-Convert-ASN1.noarch \
scsi-target-utils httpd
```
- Node has a fully installed and configured installation of Xen that allows controlling the hypervisor via HTTP from localhost.

```
yum install -y xen
sed --in-place 's/#(xend-http-server no)/(xend-http-server yes)/' /etc/xen/xend-config.sxp
sed --in-place 's/#(xend-address localhost)/(xend-address localhost)/' /etc/xen/xend-config.sxp
/etc/init.d/xend restart
```
- Firewall rules must permit the Eucalyptus components to communicate with one another, and clients to communicate with Eucalyptus. On the front-end, ports 8443, 8773, 8774 and 9001 must be open; on the node, port 8775 must be open. If you are planning on using Elastic IPs and/or Security Groups, consider disabling the firewall and use Eucalyptus facilities for enabling custom firewall rules (see Network configuration for more information). To do so, on both the front-end and the nodes:
 - `run system-config-securitylevel-tui`
 - select Security Level: Disabled
 - select OK

Download and Install RPMs

Eucalyptus binary installation is broken up into several packages: one for each of the components (CLC, Walrus, CC, etc.), as well as a couple of common packages.

There are two options for downloading and installing the packages:

Yum option

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

```
[euca]
name=Eucalyptus
baseurl=http://www.eucalyptussoftware.com/downloads/repo/eucalyptus/${VERSION}/yum/centos/${BASEARCH}
gpgcheck=0
```

Replace `${VERSION}` and `${BASEARCH}` in the `baseurl=` line above.

Now install Eucalyptus on the front-end:

```
yum install eucalyptus-cloud eucalyptus-cc eucalyptus-walrus \
eucalyptus-sc
```

and on the node:

```
yum install eucalyptus-nc
```

Tarball option

The packages are available in a single tarball, wherein we also include copies of third-party CentOS packages that Eucalyptus depends on (Rampart, Axis2C, many Java libraries), at <http://www.eucalyptus.com/download/eucalyptus> (look for a CentOS tarball of the right Eucalyptus version and architecture).

Untar the bundle in a temporary location:

```
tar zxvf eucalyptus-$VERSION-*.tar.gz
cd eucalyptus-$VERSION-*
```

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

Install RPMs on the front end

First, on the front end, install third-party dependency RPMs:

```
cd eucalyptus-$VERSION*-rpm-deps-x86_64

rpm -Uvh aoetools-21-1.el4.x86_64.rpm \
  euca-axis2c-1.6.0-1.x86_64.rpm \
  euca-rampartc-1.3.0-1.x86_64.rpm \
  vblade-14-lmdv2008.1.x86_64.rpm \
  vtun-3.0.2-1.el5.rf.x86_64.rpm \
  lzo2-2.02-3.el5.rf.x86_64.rpm \
  perl-Crypt-OpenSSL-Random-0.04-1.el5.rf.x86_64.rpm \
  perl-Crypt-OpenSSL-RSA-0.25-1.el5.rf.x86_64.rpm \
  perl-Crypt-X509-0.32-1.el5.rf.noarch.rpm \
  python25-2.5.1-bashton1.x86_64.rpm \
  python25-devel-2.5.1-bashton1.x86_64.rpm \
  python25-libs-2.5.1-bashton1.x86_64.rpm
cd ..
```

then install the `-cloud`, `-walrus`, `-cc` and `-sc` RPMs:

```
rpm -Uvh eucalyptus-$VERSION-*.x86_64.rpm \
  eucalyptus-common-java-$VERSION-*.x86_64.rpm \
  eucalyptus-cloud-$VERSION-*.x86_64.rpm \
  eucalyptus-walrus-$VERSION-*.x86_64.rpm \
  eucalyptus-sc-$VERSION-*.x86_64.rpm \
  eucalyptus-cc-$VERSION-*.x86_64.rpm \
  eucalyptus-gl-$VERSION-*.x86_64.rpm
```

Install RPMs on the nodes

Next, on each node install the dependency packages:

```
cd eucalyptus-$VERSION*-rpm-deps-x86_64
rpm -Uvh aoetools-21-1.el4.x86_64.rpm \
  euca-axis2c-1.6.0-1.x86_64.rpm \
  euca-rampartc-1.3.0-1.x86_64.rpm \
  perl-Crypt-OpenSSL-Random-0.04-1.el5.rf.x86_64.rpm \
  perl-Crypt-OpenSSL-RSA-0.25-1.el5.rf.x86_64.rpm \
  perl-Crypt-X509-0.32-1.el5.rf.noarch.rpm \
  python25-2.5.1-bashton1.x86_64.rpm \
  python25-devel-2.5.1-bashton1.x86_64.rpm \
  python25-libs-2.5.1-bashton1.x86_64.rpm
cd ..
```

then install the node controller RPM with dependencies:

```
rpm -Uvh eucalyptus-$VERSION-*.x86_64.rpm \
  eucalyptus-gl-$VERSION-*.x86_64.rpm \
  eucalyptus-nc-$VERSION-*.x86_64.rpm
```

Post-Install Steps

The last step in the installation is to make sure that the user 'eucalyptus', which is created at RPM installation time, is configured to interact with the hypervisor through libvirt on all of your compute nodes. On each node, access the libvirtd configuration, `/etc/libvirt/libvirtd.conf`, and confirm that the following lines are uncommented:

```
unix_sock_group = "libvirt"
unix_sock_ro_perms = "0777"
unix_sock_rw_perms = "0770"
```

To check that libvirt is configured and interacting properly with the hypervisor, run the following command on each node:

```
# on XEN
su eucalyptus -c "virsh list"
# on KVM
su eucalyptus -c "virsh qemu:///system list"
```


The output of that command *may* include error messages (failed to connect to xend), but as long as it includes a listing of all domains (at least `Domain-0`), the configuration is in order.

Now start up your Eucalyptus services. On the front-end:

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

On the node:

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

At this point you should be ready to proceed with first-time configuration.

Installing Eucalyptus (2.0) on openSUSE 11.2

Eucalyptus can be installed on openSUSE 11 from source, or by using binary RPM packages. This document details the steps required to install Eucalyptus from RPMs. In what follows, the value of **\$VERSION** must be set to the version of Eucalyptus you wish to install. For example, we can set the value to 2.0.3 using bash:

```
export VERSION=2.0.3
```

Prerequisites

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

- Front-end, node and client machine system clocks are synchronized (i.e. using NTP).

```
sntp -P no -r pool.ntp.org
yast2 -i ntp
/etc/init.d/ntp restart
```

- Install all other dependency packages that are required for Eucalyptus to run on the front end

```
zypper install apache2 apache2-prefork java-1_6_0-openjdk \
    java-1_6_0-openjdk-devel mozilla-nss libvirt curl \
    vlan dhcp-server bridge-utils \
    perl-Crypt-OpenSSL-RSA perl-Crypt-OpenSSL-Random tgt
```

and on the node

```
zypper install vlan apache2 perl-Crypt-OpenSSL-RSA \
    perl-Crypt-OpenSSL-Random tgt
```

- Install Xen packages and network bridge, using the 'yast2' command and following these steps:
 - Virtualization
 - Install Hypervisor and Tools
 - Select 'OK'

This creates the network bridge for you, so there is no need to create it yourself.

- Node has a fully installed and configured installation of Xen.

```
sed --in-place \
's/#(xend-http-server no)/(xend-http-server yes)/' \
/etc/xen/xend-config.sxp
sed --in-place \
's/#(xend-address localhost)/(xend-address localhost)/'\
/etc/xen/xend-config.sxp
/etc/init.d/xend restart
```

- We recommend that you verify your Xen installation by manually bringing up a VM and testing that it has network connectivity using bridged networking.
- 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 and 9001 must be open. On the node, port 8775 must be open
 - If you are planning on using Elastic IPs and/or Security Groups, you may want to consider disabling the firewall and use Eucalyptus facilities for enabling custom firewall rules (see Network configuration for more information).

```
yast2 firewall startup manual
/etc/init.d/SuSEfirewall2_init stop
reboot
```

Download and Install RPMs

Eucalyptus binary installation is broken up into several packages: one for each of the components (CLC, Walrus, CC, etc.), as well as a couple of common packages.

There are two options for downloading and installing the packages:

Zypper option

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

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

answer question about trusting packages from this repository then refresh it

```
zypper refresh Eucalyptus
```

now install eucalyptus on the front-end

```
zypper install eucalyptus-cloud eucalyptus-cc \
    eucalyptus-walrus eucalyptus-sc
```

and on the node

```
zypper install eucalyptus-nc
```

Tarball option

These packages are available in a single tarball, wherein we also include copies of third-party openSUSE packages that Eucalyptus depends on (Rampart, Axis2C, many Java libraries), at <http://www.eucalyptus.com/download/eucalyptus> (look for a openSUSE tarball of the right Eucalyptus version and architecture).

Untar the bundle in a temporary location:

```
tar zxvf eucalyptus-$VERSION-<em>.tar.gz
cd eucalyptus-$VERSION-</em>
```

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

Install RPMs on the front end

First, on the front end, install third-party dependency RPMs:

```
cd eucalyptus-$VERSION*-rpm-deps-x86_64
rpm -Uvh aoetools-21-1.el4.x86_64.rpm \
    euca-axis2c-1.6.0-1.x86_64.rpm \
    euca-rampartc-1.3.0-1.x86_64.rpm \
    vblade-14-1mdv2008.1.x86_64.rpm \
    vtun-3.0.1-1.x86_64.rpm
cd ..
```

then install the -cloud, -walrus, -cc and -sc RPMs:

```
rpm -Uvh eucalyptus-$VERSION-<em>.x86_64.rpm \
    eucalyptus-common-java-$VERSION-</em>.x86_64.rpm \
    eucalyptus-cloud-$VERSION-<em>.x86_64.rpm \
    eucalyptus-sc-$VERSION-</em>.x86_64.rpm \
    eucalyptus-walrus-$VERSION-<em>.x86_64.rpm \
    eucalyptus-cc-$VERSION-</em>.x86_64.rpm \
    eucalyptus-gl-$VERSION-*.x86_64.rpm
```

Install RPMs on the nodes

Next, on each node, install the dependency packages:

```
cd eucalyptus-$VERSION*-rpm-deps-x86_64
rpm -Uvh aoetools-25-2.49.x86_64.rpm \
    euca-axis2c-1.6.0-1.x86_64.rpm \
    euca-rampartc-1.3.0-1.x86_64.rpm \
    vblade-15-2.49.x86_64.rpm
cd ..
```

then install the node controller RPM with dependencies:

```
rpm -Uvh eucalyptus-$VERSION-<em>.x86_64.rpm \
    eucalyptus-gl-$VERSION-</em>.x86_64.rpm \
    eucalyptus-nc-$VERSION-*.x86_64.rpm
```

Regardless of the download and installation option used, make sure that the libvirt daemon (libvirtd) is running and configured to start at boot.

- `/etc/init.d/libvirtd start`

- check eucalyptus can interact with libvirt `su eucalyptus -c "virsh list"`

On the node, uncomment these lines in `/etc/libvirt/libvirtd.conf`:

```
unix_sock_group = "libvirt"
unix_sock_ro_perms = "0777"
unix_sock_rw_perms = "0770"
auth_unix_ro = "none"
auth_unix_rw = "none"
```

Post-Install Steps

Now start up your Eucalyptus services. On the front-end:

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

On the node:

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

At this point you should be ready to proceed with first-time configuration.

Installing Eucalyptus (2.0) on Fedora 12

Eucalyptus can be installed on Fedora 12 from source or by using binary RPM packages. This document details the steps required to install Eucalyptus from RPMs. In what follows, the value of **\$VERSION** must be set to the version of Eucalyptus you wish to install. For example:

```
export VERSION=2.0.3
```

Notice: Before you begin, please ensure that you have an up-to-date Fedora 12 installation on your target machine(s).

Prerequisites

If you start with a standard Fedora installation, you will satisfy all prerequisites with the following steps:

- Front-end, node and client machine system clocks are synchronized (e.g., using NTP).

```
yum install -y ntp
ntpdate pool.ntp.org
```

- Front-end needs java to manipulate a bridge and the binaries for dhcp server (do not configure it nor run it on the CC):

```
yum install -y java-1.6.0-openjdk java-devel ant \
               ant-nodeps dhcp httpd boto
yum install -y aoetools vblade vtun bridge-utils
```

- Install KVM and other related tools for the nodes

```
yum install -y qemu-kvm libvirt aoetools vblade
```

- Firewall rules must permit the Eucalyptus components to communicate with one another, and clients to communicate with Eucalyptus. On the front-end, ports 8443, 8773, 8774 and 9001 must be open; on the node, port 8775 must be open. If you are planning on using Elastic IPs and/or Security Groups, you may want to consider disabling the firewall and use Eucalyptus facilities for enabling custom firewall rules (see Network configuration for more information). On both the front-end and the nodes:

For example, from a text console:

- `run system-config-firewall-tui`
- select Firewall to be Disabled
- select OK

- Disable SELINUX using your favorite editor to open the SELINUX configuration file. For example:

```
vi /etc/selinux/config
```

Edit the configuration file so that SELINUX has the value of "disabled" (`SELINUX="disabled"`).

Download and Install RPMs

Eucalyptus binary installation is broken up into several packages: one for each of the components (CLC, Walrus, CC, etc.), as well as a couple of common packages.

There are two options for downloading and installing the packages:

Yum option:

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

```
[euca]
name=Eucalyptus
baseurl=http://www.eucalyptussoftware.com/downloads/repo/eucalyptus/{{VERSION}}/yum/fedora/{{BASEARCH}}
gpgcheck=0
```

Replace {{VERSION}} and {{BASEARCH}} in the baseurl= line above.

Now install eucalyptus on the front-end,

```
yum install eucalyptus-cloud eucalyptus-cc \
            eucalyptus-walrus eucalyptus-sc
```

and on the node

```
yum install eucalyptus-nc
```

Tarball option

These packages are available in a single tarball, wherein we also include copies of third-party Fedora packages that Eucalyptus depends on (Rampart, Axis2C, many Java libraries), at <http://www.eucalyptus.com/download/eucalyptus> (look for a Fedora tarball of the correct Eucalyptus version and architecture).

Untar the bundle in a temporary location:

```
tar zxvf eucalyptus-$VERSION-*.tar.gz
cd eucalyptus-$VERSION-*
```

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

Install RPMs on the front end

First, on the front end, install third-party dependency RPMs:

```
yum install -y perl-Crypt-OpenSSL-RSA perl-Crypt-OpenSSL-Random \
            scsi-target-utils wget
```

```
cd eucalyptus-$VERSION*-rpm-deps-x86_64
rpm -Uvh euca-axis2c-1.6.0-1.x86_64.rpm \
        euca-rampartc-1.3.0-1.x86_64.rpm
cd ..
```

then install the -cloud, -walrus, -cc and -sc RPMs:

```
rpm -Uvh eucalyptus-$VERSION-*.x86_64.rpm \
        eucalyptus-common-java-$VERSION-*.x86_64.rpm \
        eucalyptus-cloud-$VERSION-*.x86_64.rpm \
        eucalyptus-sc-$VERSION-*.x86_64.rpm \
        eucalyptus-walrus-$VERSION-*.x86_64.rpm \
        eucalyptus-cc-$VERSION-*.x86_64.rpm \
        eucalyptus-gl-$VERSION-*.x86_64.rpm
```

Install RPMs on the nodes

Next, on each node, install the dependency packages:

```
yum install -y perl-Crypt-OpenSSL-RSA \
            perl-Crypt-OpenSSL-Random wget
cd eucalyptus-$VERSION*-rpm-deps-x86_64
rpm -Uvh euca-axis2c-1.6.0-1.x86_64.rpm \
        euca-rampartc-1.3.0-1.x86_64.rpm
cd ..
```

then install the node controller RPM with dependencies:

```
rpm -Uvh eucalyptus-$VERSION-*.x86_64.rpm \
        eucalyptus-gl-$VERSION-*.x86_64.rpm \
        eucalyptus-nc-$VERSION-*.x86_64.rpm
```

Post-Install Steps**Modify libvirtd.conf file**

On the node, /etc/libvirt/libvirtd.conf file needs to be modified:

```
unix_sock_group = "kvm"
unix_sock_ro_perms = "0777"
```

```
unix_sock_rw_perms = "0770"
auth_unix_ro = "none"
auth_unix_rw = "none"
```

Once you made the modification, stop and start the libvirt, and make sure the sockets belong to the correct group:

```
/etc/init.d/libvirtd stop
/etc/init.d/libvirtd start
chown root:kvm /var/run/libvirt/libvirt-sock
chown root:kvm /var/run/libvirt/libvirt-sock-ro
```

Now, the node should be ready to run.

Run Eucalyptus

Now start up your Eucalyptus services. On the front-end:

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

On the node:

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

At this point you should be ready to proceed with **first-time configuration**.

Installing Eucalyptus (2.0) on Debian "squeeze"

Eucalyptus can be installed on Debian squeeze using binary DEB packages. In what follows, the value of **\$VERSION** must be set to the version of Eucalyptus you wish to install. For example, we can set the value to 2.0.3 using bash:

```
export VERSION=2.0.3
```

Prerequisites

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

- Ensure that `locales` are properly configured (use `dpkg-reconfigure locales`).
- Synchronize clocks (e.g., using NTP: `ntpdate pool.ntp.org`) across all Eucalyptus machines and client machines.
- If using a firewall, permit the Eucalyptus components to communicate with one another, and permit clients to communicate with Eucalyptus. On the front-end, ports 8443, 8773, 8774 and 9001 must be open. On the node, port 8775 must be open.
- If running in SYSTEM mode, which is the default networking mode, your node machine(s) must be configured with a bridge as the primary interface. For example, you may try:

```
sudo apt-get install bridge-utils
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 to have DHCP assign an address to the bridge, use:

```
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
```

Finally, restart the network by rebooting the machine or entering the following command:

```
/etc/init.d/networking restart
```

Download DEBs

Eucalyptus binary installation is broken up into several packages: one for each of the components (CLC, Walrus, CC, etc.), as well as a couple of common packages.

There are two options for downloading the DEB packages:

Remote repository option

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

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

And then run:

```
apt-get update
```

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

Tarball (local repository) option

DEB packages are also available in a single "tarball", wherein we include copies of third-party packages that Eucalyptus depends on (Rampart, Axis2C, and Java libraries).

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

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

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

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

For 32-bit:

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

For 64-bit

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

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

**** Install DEBs**

On the front end, where cloud controller, Walrus, cluster controller, and storage controller will run, install the appropriate DEBs:

```
aptitude install eucalyptus-common eucalyptus-cloud \
eucalyptus-walrus eucalyptus-sc eucalyptus-cc
```

On the compute nodes, install required iscsi dependencies and the node-controller DEB :

```
aptitude install open-iscsi libcrypt-openssl-random-perl \
libcrypt-openssl-rsa-perl libcrypt-x509-perl \
eucalyptus-nc
```

(You may safely ignore the error `adduser: The group 'libvirt' does not exist.`)

Post-Install Steps

Modify `qemu.conf` file

On the node, make sure `libvirt` is configured to run as user "eucalyptus":

```
sudo vi /etc/libvirt/qemu.conf
# set the field user to be: user = "eucalyptus"
```

Modify `libvirtd.conf` file

On the node, uncomment the following lines in `/etc/libvirt/libvirtd.conf`:

```
unix_sock_group = "libvirt"
unix_sock_ro_perms = "0777"
unix_sock_rw_perms = "0770"
auth_unix_ro = "none"
auth_unix_rw = "none"
```

Restart `libvirtd`

Once you've made the modification, stop and start libvirt, and make sure the sockets belong to the correct group:

```
/etc/init.d/libvirt-bin stop
/etc/init.d/libvirt-bin start
chown root:libvirt /var/run/libvirt/libvirt-sock
chown root:libvirt /var/run/libvirt/libvirt-sock-ro
```

At this point you should be ready to proceed with **first-time configuration**.

Upgrading to Eucalyptus 2.0

These instructions are for those who would like to upgrade to Eucalyptus 2.0 series software from a previous source-based or package-based Eucalyptus installation. If you're running a version of Eucalyptus prior to version 1.6.2, please, follow the instructions for upgrading to 1.6. before following these instructions. The Instructions below assume that **\$EUCALYPTUS** points to the root of the new Eucalyptus installation and **\$OLD_EUCA** points to the root of the old installation (which can be the same as the new one).

1. Warning! Stop all Eucalyptus Services and Back Up Your Current Installation

We strongly recommend backing up your installation before performing an upgrade. To do so, you must first terminate all Eucalyptus instances; stop all Eucalyptus components; and "kill" all Eucalyptus processes. These steps and additional specific steps for backing up your current installation are provided in the first part of the Backup section.

2. Install Eucalyptus 2.0

Binary Packages

Note: If you are using our binary packages (RPMs or DEBs), `euca_upgrade` will be invoked automatically, creating backups in

```
/root/eucalyptus.backup.$TIMESTAMP.
```

- If upgrading using **binary packages**, follow the steps in the installation instructions for a specific distribution:
 - CentOS 5
 - OpenSUSE 11
 - Debian "Squeeze" (see warning)
 - Fedora 12

and, afterwards, proceed to section 3. *Start Eucalyptus and Verify the Upgrade.*

Source

- If upgrading a **source-based** installation, follow the steps in the Source Code Installation section of the Administrator's Guide and, afterwards, **return here**, and run the following upgrade commands:
To upgrade the front-end, run:
 - `$EUCALYPTUS/usr/share/eucalyptus/euca_upgrade --new $EUCALYPTUS --old $OLD_EUCA --conf --keys --db`and to upgrade the nodes, run:
 - `$EUCALYPTUS/usr/share/eucalyptus/euca_upgrade --new $EUCALYPTUS --old $OLD_EUCA --conf --keys`

3. Start Eucalyptus and Verify the Upgrade

- Start the services on the appropriate machines (after a DEB-based install they should already be running):

```
$EUCALYPTUS/etc/init.d/eucalyptus-nc start
$EUCALYPTUS/etc/init.d/eucalyptus-cc cleanstart
$EUCALYPTUS/etc/init.d/eucalyptus-cloud start
```
- In a Web browser, load `https://front-end:8443/` and log in as before. Verify that the user accounts and the images are there.
- **Important:** Verify that the Buckets Path and Volumes Path settings under the Configuration tab of the Web interface matches the actual locations of the buckets and volumes *before* running any instances or using buckets.
- Verify that the nodes are back up and that they can run your old instances (if not, see the Troubleshooting section.)

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

4. Optionally: Roll Back to an Earlier Installation

- Follow the steps in the second part of the Backup section, called "Restoration". If you are relying on the backup created by `euca_upgrade` during a package-based upgrade, then after re-installing the old packages, copy back the saved state (the backed up copies of `db/*`, `keys/*`, `etc/eucalyptus/eucalyptus.conf`) to your restored installation. Then, start Eucalytpus, as before.

Configuration

This section of the Administrator's Guide describes how to configure Eucalyptus, both during installation and after a decision to reconfigure parts of the system.

First-time Setup (2.0)

This document describes the steps for activating and possibly further configuring Eucalyptus after the software has been installed on all nodes (either from source or using binary packages).

After you've started all components, you will need to perform registration so that they can communicate with each other.

Registering Eucalyptus Components

This section will assume that you have installed all Eucalyptus components and they are up and running. We will assume that your Eucalyptus setup consists of one front end and one or more nodes.

First, you will need to register various front end components. To do this, run the following commands on the front end.

```
$EUCALYPTUS/usr/sbin/euca_conf --register-walrus <front end IP address>
$EUCALYPTUS/usr/sbin/euca_conf --register-cluster <clustername> <front end IP address>
$EUCALYPTUS/usr/sbin/euca_conf --register-sc <clustername> <front end IP address>
```

Finally, you need to register nodes with the front end. To do so, run the following command on the front end,

```
$EUCALYPTUS/usr/sbin/euca_conf --register-nodes "<Node 0 IP address> <Node 1 IP address> ... <Node N IP address>"
```

where "<Node X IP address>" is the IP address of host X that is running the Node Controller (NC).

At this point, you have successfully registered Eucalyptus components and are ready to begin your initial configuration.

Initial Configuration

Point your browser to,

```
https://<front end IP address>:8443
```

Since Eucalyptus is using a self-signed certificate, your browser is likely to prompt you to accept the certificate. 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 and password both of which are set to `admin` initially.

Upon logging in the first time you will be asked to

- 1. change the admin password,
- 2. set the admin's email address, and
- 3. confirm the IP of the Cloud Controller host.

After clicking 'Submit', you will see the 'Configuration' tab. Since you've used `euca_conf` to register Walrus and a cluster, they will be listed along with a few configurable parameters. Look over the parameters to see if any need adjustment.

Notice: in the 'Configuration' tab you can configure the 'VM Types'. There are 5 VM type available and you can configure the amount of memory, numbers of cores and default size of the root file system each VM will use. We suggest you make the 'Disk (GB)' 10 for all images, since this is the default value for EMIs. Making it too big it may slow down the instances start-up time.

For more information, see the Management section.

To use the system with client tools, you must obtain user credentials. From the 'Credentials' tab, Eucalyptus users can obtain two types of credentials: x509 certificates and query interface credentials. Use the 'Download Credentials' button to download a zip-file with both or click on the 'Show Keys' to see the query interface credentials. You will be able to use your credentials with Euca2ools, Amazon EC2 tools and third-party tools like rightscale.com. Create a directory to store your credentials, unpack the zip-file into it, and source the included 'eucarc':

```
mkdir $HOME/.euca
unzip euca2-admin-x509.zip -d $HOME/.euca
. $HOME/.euca/eucarc
```

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

Hypervisor Configuration

Eucalyptus deploys instances (i.e., virtual machines) on a hypervisor. Eucalyptus can use either `xen` or `kvm` hypervisors. To interact with them, Eucalyptus employs `libvirt` virtualization API. The best choice for the hypervisor depends on its support for your hardware, on the support for the hypervisor in your OS (some distros support KVM better, some support Xen better), as well as personal preferences.

Another consideration is support for Eucalyptus features in the hypervisor. Because Eucalyptus uses features that only recently have been added to hypervisors, some combinations of hypervisor and kernel do not function as intended. The most common problem we encounter has to do with support for attaching and removing block devices. On some kernels, for example, you may see a lot of `WARN_ON` messages in the logs (similar to kernel oops), with KVM you will not be able to specify the exact device block (it will be chosen by the system), and on some hypervisor-kernel combinations EBS will not work at all (e.g., Debian "squeeze" with 2.6.30-2-amd64 kernel and KVM v88).

Virtio Configuration

With a sufficiently recent version of the KVM hypervisor (60 or greater) and guest VMs with Virtio drivers (available for both Linux and Windows), using Virtio for I/O of guest VMs is an option. To enable the use of Virtio by Eucalyptus, set to "1" one or more of the Virtio options in `eucalyptus.conf` file on the NC hosts:

- **USE_VIRTIO_DISK** - if set to "1", Eucalyptus will use Virtio for EBS (elastic block store) volumes being attached to VMs running on the node
- **USE_VIRTIO_ROOT** - if set to "1", Eucalyptus will use Virtio for the root file system disk of all instances started on the node
- **USE_VIRTIO_NET** - if set to "1", Eucalyptus will use Virtio for the network card of all instances started on the node

For more information on Virtio, see <http://www.linux-kvm.org/page/Virtio>

Running a test VM with hypervisor tools

First of all, before even installing Eucalyptus, install a hypervisor of your choice and, based on the hypervisor's documentation, try to construct and run a test VM from the command line.(If you cannot run a VM *outside* Eucalyptus, you will not be able to run any VMs *through* Eucalyptus.)

Running a Xen VM usually involves creating a configuration file and passing it to the `xm create` command. Running a KVM VM usually involves invoking `kvm` with many parameters on the command-line.

If the hypervisor doesn't work out of the box on your distro, you may want to experiment with options. For Xen, the options are specified in:

```
/etc/xend/xend-config.sxp
```

We had good luck with these:

```
(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 '')
```

Running a test VM with libvirt's virsh

Since Eucalyptus interacts with hypervisors through `libvirt`, it is also a good idea to ensure that `libvirt` is set up properly, particularly for user "eucalyptus". A way to do so is to try

```
virsh list
```

as the "eucalyptus" user (root usually can always connect). If that fails, the solutions are distribution-specific: for example, on some Debian-based distros, the user "eucalyptus" needs to be in the group **libvirt** or **libvirtd**.

On distros using **PolicyKit**, you may want to ensure that in

```
/etc/PolicyKit/PolicyKit.conf
```

there is something like

```
<config version="0.1">
<match action="org.libvirt.unix.manage">
  <match user="eucalyptus">
    <return result="yes"/>
  </match>
</match>
</config>
```

As the last resort, you may want to look into

```
/etc/libvirt/libvirtd.conf
```

and keep an eye on logs in

```
/var/log/libvirt
```

Eucalyptus Network Configuration (2.0)

Eucalyptus provides the network infrastructure that enables access to and from VM instances. Typically, these VMs reside on *virtual subnets* that must be configured to be separate and distinct from the physical network on which Eucalyptus components interact. Eucalyptus creates these virtual subnets during instance creation and disposes of them when the last instance on a particular virtual subnet is terminated. Note that the administrator must ensure that the virtual subnet IP address space does not contain, is not contained by, and does not conflict with any part of the physical network IP address space.

Eucalyptus offers four different networking modes (MANAGED, MANAGED-NOVLAN, SYSTEM, and STATIC) that accommodate Eucalyptus deployment on a variety of physical network infrastructures. Before starting Eucalyptus, the administrator should select one of these four modes by modifying the `eucalyptus.conf` file on each machine running a Eucalyptus component.

WARNING: If you edit any networking related value in `eucalyptus.conf`, for the changes to take effect you must perform a "clean restart" of the CC using the following command (make sure to terminate all instances before performing the clean restart):

```
$EUCALYPTUS/etc/init.d/eucalyptus-cc cleanrestart
```

This document presents the networking features available in Eucalyptus, along with information and requirements pertaining to the underlying physical network, to help you select the appropriate networking mode for your needs and your infrastructure. We follow up with detailed instructions for configuring Eucalyptus using each Eucalyptus networking mode.

1. Networking Features and Requirements

Ultimately your selection of a Eucalyptus networking mode will depend on the networking features you want for your cloud and the degree of control you have over the configuration of the underlying physical network. Here we present a list of Eucalyptus networking features and physical network requirements for your reference when choosing and configuring a Eucalyptus networking mode.

Eucalyptus Networking Features:

Connectivity: All network modes provide IP-level connectivity for VMs started by Eucalyptus. Whether a VM is reachable via IP on a specific port from a host outside of a Eucalyptus installation depends on whether the VM received a public IP address and whether the VM's security group rules (described below) allow traffic on the port.

IP Control: In all modes except SYSTEM, Eucalyptus assigns IP addresses to VMs. In SYSTEM mode, one must allow a DHCP server outside of Eucalyptus to assign IPs to VMs that Eucalyptus starts.

Security Groups: Security groups are sets of networking rules (in effect a firewall) applied to all VM instances associated with a group. In practice a security group defines the access rules for all VM instances associated with a group. For example, a user can specify ingress rules, such as allowing ping (ICMP) or SSH (TCP, port 22) traffic to reach VMs in a specific security group. Note that when you create a VM instance, unless otherwise specified at instance run-time, it is assigned to a "default" security group that denies incoming network traffic from all sources. Thus, to allow login and usage of a new VM instance you must authorize network access to the default security group with the `euca-authorize` command. Security groups are available in MANAGED and MANAGED-NOVLAN Mode.

Elastic IPs: With elastic IPs the user gains control over a set of Public IP addresses. Once allocated to the user, those same IPs can be dynamically associated to running instances, thus overriding pre-assigned public IPs. This allows users to run well-known services (e.g., Web sites, etc.) within the Eucalyptus cloud. Elastic IPs are available in MANAGED and MANAGED-NOVLAN mode.

Metadata service: Eucalyptus provides an AWS-compatible metadata service that lets you access instance-specific information from inside a running VM. Data provided includes the instance's public and private IP address, reservation ID, user-data (specified with `euca-run-instance` command), and a launch index (useful for differentiating multiple instances launched with -n option of the `euca-run-instance` command).

The metadata service is available in all Eucalyptus networking modes. However, the way you access the service from within virtual machines differs depending on the networking mode used to configure your Eucalyptus installation: If your system is configured using MANAGED or MANAGED-NOVLAN modes, then the metadata service is available to VMs just as in Amazon EC2 as shown:

```
http://169.254.169.254/<specific meta-data information request>
```

If your system is configured using SYSTEM or STATIC networking modes, then retrieving data requires the IP (or hostname) and port number of the Eucalyptus Cloud Controller (CLC):

```
http://<IP or hostname of CLC>:8773/<specific meta-data information request>
```

(We recommend that you set up a DNS entry for each cloud controller. This way you can configure your images to access the metadata

service using a DNS name, thus avoiding the need to recreate pre-configured images in the event a specific IP address changes.)

VM Isolation: While network traffic between VM instances belonging to a security group is always open, Eucalyptus can enforce isolation of network traffic between different security groups. This is enforced using a VLAN tag per security group, thus, protecting VMs from unwanted eavesdropping by VM instances belonging to other security groups. VM isolation is available only in MANAGED mode. In MANAGED-NOVLAN mode, VM isolation is guaranteed by having different security groups on different subnets—this translates into Layer-3 only VM isolation.

Network Requirements:

VLAN clean: MANAGED mode uses a Virtual LAN (VLAN) to enforce network isolation between instances. If the underlying physical network is also using a VLAN, there can be conflicts that prevent instances from being network accessible. Thus, it is important to know if your network is 'VLAN clean' (that is, VLANs are usable by Eucalyptus).

Testing VLAN clean:

The admin can verify that the network is VLAN clean (allows/forwards VLAN tagged packets) between machines running Eucalyptus components by performing the following test:

First, choose two IP addresses from the subnet you plan to use with Eucalyptus. (In the following example, those IP addresses are 192.168.1.1 and 192.168.1.2). Next, on the front end machine, choose the interface that is on the local Ethernet (and is set in eucalyptus.conf as VNET_PRIVINTERFACE), and run:

```
[root@clc]# vconfig add <interface> 10
[root@clc]# ifconfig <interface>.10 192.168.1.1 up
```

Next, on the node, choose the interface on the local network (and is set in eucalyptus.conf as VNET_PRIVINTERFACE), and run:

```
[root@node1]# vconfig add <interface> 10
[root@node1]# ifconfig <interface>.10 192.168.1.2 up
```

Then, perform a ping between the two hosts to validate the interface settings.

On the front end:

```
[root@clc]# ping 192.168.1.2
```

On the node:

```
[root@node1]# ping 192.168.1.1
```

If this VLAN clean test fails, 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).

Available Public IPs: Instances typically have two IPs associated with them: a private one (belonging to the virtual subnet allocated to their group) and a public one (an IP belonging to the subnet that allows the CC to communicate externally). Most Eucalyptus networking modes are responsible for the allocation of both sets of IPs: In this case the administrator must specify in advance a range of both private IP addresses (used for the virtual subnet) and public IP addresses (used for accessing VM instances).

DHCP server: In all Eucalyptus networking modes (except SYSTEM), a DHCP server binary must be installed on the CC machine. Usually this DHCP server is not configured and thus *not* active. In these modes an active DHCP server on the same subnet may prevent VM instances from receiving the correct IP address. In SYSTEM mode, however, a pre-configured DHCP server already active on the physical subnet is required.

For the modes requiring an installed DHCP server, that binary must be command-line compatible with ISC DHCP Daemon version 3.0 (this daemon is provided with most Linux distributions, for example with RHEL/CentOS it is provided in a package named dhcp). To configure Eucalyptus to use the DHCP server, you must edit eucalyptus.conf to instruct Eucalyptus as to the location of the DHCP binary, as shown:

```
VNET_DHCPDAEMON="/usr/sbin/dhcpd"
```

If your DHCP daemon binary is configured to run as user "root," (as is the case with RHEL/CentOS and openSUSE) then you do not need to specify a VNET_DHCPUSER.

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 or later), then you must configure Eucalyptus to be aware of that user:

```
VNET_DHCPUSER="<dhcpUsername>"
```

2. Choosing a Networking Mode

Before choosing a networking mode for your Eucalyptus network configuration, it is important to understand the interrelationship between feature availability and the requirements of the underlying physical network for each Eucalyptus networking mode—some features can be implemented only if certain requirements are met by the underlying physical network.

The following chart shows the relationship between the networking features available in each Eucalyptus networking mode and the corresponding requirements for the underlying physical network.

NETWORKING MODE	PHYSICAL NETWORK REQUIREMENTS EUCALYPTUS NETWORKING FEATURES								
	VLAN clean	Public IPs	Non-Eucalyptus DHCPD	Connectivity	IP control	Security Groups	Elastic IPs	Metadata service	VM Isolation
MANAGED	!	!		!	!	!	!	!	!
MANAGED- NOVLAN		!	"	!	!	!	!	!	!#
STATIC		!	"	!	!			!	
SYSTEM			!	!				!	

" - A non-Eucalyptus DHCP server present on the network will interfere with Eucalyptus IP address allocation.
- Provides Layer-3 only VM isolation.

Eucalyptus offers four different networking modes: (MANAGED, MANAGED-NOVLAN, SYSTEM, and STATIC). MANAGED mode is the most full-featured networking mode, providing security groups, VM isolation, elastic IPs and Metadata service. Note that in MANAGED mode the underlying physical network must be VLAN clean, as Eucalyptus provides and manages its own VLAN tagging. If your network is not VLAN clean, you can use MANAGED-NOVLAN mode, which provides the full set of networking features, with the exception of VM isolation between instances.

In the remaining networking modes, SYSTEM and STATIC, there are no virtual subnets—VM instances appear on the physical network as if they were physical machines; and VM instances are directly bridged with the NC machine's physical ethernet device. SYSTEM mode is designed to be the least demanding on the physical network. In this mode, VM instances obtain their configuration from the DHCP server serving the whole physical network. STATIC 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, an unused MAC/IP pair is chosen for the instance. A Eucalyptus-controlled DHCP server then serves the VM instances.

3. Configuring Eucalyptus Networking Modes

In this section, we provide detailed configuration instructions—including configuration examples and additional information—for each of the four Eucalyptus networking modes.

3.1 About Network Configuration Examples

For the configuration examples in the following sections, we use two different network configurations loosely based on our Eucalyptus Community Cloud (shown in **Figure 1** and **Figure 2** below). In both configurations, the public network used is 173.205.188.0/24 with the router at 173.205.188.1 and the local DNS server at 173.205.188.129.

Figure 1:

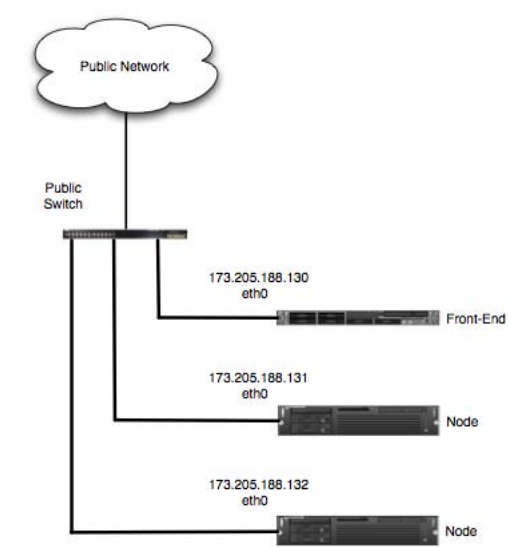


Figure 1 has a very simple configuration: all the machines have one Ethernet device (eth0) and they are all connected directly to the public network.

Figure 2:

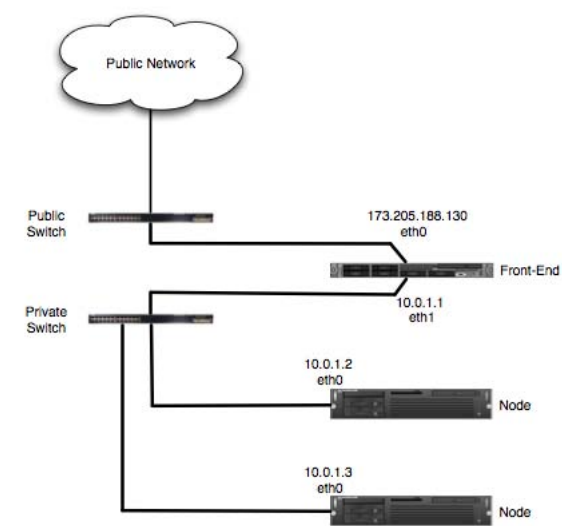


Figure 2 shows node machines on a private subnet using the front-end machine as a gateway to the public network. Note that the front-end machine has two Ethernet devices (eth0 is on the public network; eth1 is on the private network) and uses NAT to allow the nodes access to the public network. The network between the front end and the node machines is a private network of the range 10.0.1.0/24.

3.2 About Bridge Names

Most Eucalyptus networking modes require a bridge. Bridge names are both hypervisor and Linux distribution dependent. To properly configure a network mode you must know the bridge name for your system. Typically bridge names are as follows:

- For Xen 3.0 or earlier: **xenbr0**
- For Xen 3.2 and above: **eth0**
- Most distributions using KVM: **br0**

To ensure that you are using the correct bridge within your Eucalyptus configuration, enter the `brctl show` command as shown:

```
[root@clc]# brctl show
bridge name      bridge id        STP enabled      interfaces
virbr0           8000.000000000000 yes              peth0
xenbr0           8000.fefffffffff no
```

Note that the bridge name `virbr0` is created by `libvirt`. This name should not be used. Ensure the bridge is associated with the correct Ethernet device. In the above example, `peth0` is attached to the bridge.

For the remainder of this document, we assume that you have correctly identified the bridge and that such bridge is named **xenbr0**, as shown above.

3.3 About VNET_ Options

All network-related options specified in `eucalyptus.conf` use the prefix `VNET_`. The following options are the most commonly used:

VNET_DNS

This option is used to specify a nameserver available on your network. DNS must be specified as an IP address.

VNET_SUBNET, VNET_BROADCAST, VNET_NETMASK

These three options—network address, the broadcast address on the network, and the subnet mask, respectively—work together to define the configuration of a specific network. It is necessary to specify all three options when Eucalyptus requires a virtual subnet.

VNET_ADDRESSPERNET

This option 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). This option is used only when security groups are available. Typically these numbers are 16, 24, 32, 64, etc, but should never be less than 8. The value specifying this option, alongside `VNET_NETMASK`, will determine the number of available security groups in the system (`VNET_NETMASK` determines the size of the address space, while `VNET_ADDRESSPERNET` determines how the address space is partitioned, so with a bigger limit on security group size, fewer security groups can be created). **WARNING:** If `VNET_ADDRESSPERNET` is too large relative to `VNET_NETMASK` you may have very few security groups or the CLC may refuse to start altogether.

VNET_PUBLICIPS

This is the list or range of public IP addresses available for VMs. You can specify the IP addresses as a list, for example: "10.0.0.1 10.0.0.2 10.0.0.3" or as a range, for example: "10.0.0.1-10.0.0.3."

3.4 Networking Modes

Here we show you the proper configuration for each Eucalyptus networking mode. Included are requirements, limitations, VNET_ options (that must be set in `eucalyptus.conf`), and a configuration example for each mode. We also discuss important caveats where applicable.

3.4.1 MANAGED mode

In this mode, Eucalyptus manages the local network of VM instances and provides all networking features Eucalyptus currently supports, including VM network isolation, security groups, elastic IPs, and metadata service. Note that in this mode each security group requires a separate VLAN, which Eucalyptus controls and maintains, thus the underlying physical network must be VLAN clean.

Requirements

- There is an available range of IP addresses to be used for the virtual subnets that do not interfere with the physical network. Typically these IP addresses are selected from the private IP ranges: 192.168.x.x, 10.x.x.x, etc.
- Network must be VLAN clean, meaning that all switch ports that Eucalyptus components are connected to will allow and forward VLAN tagged packets.
- You are not running a firewall on the front end (CC) or your firewall is compatible with the dynamic changes performed by Eucalyptus when working with security groups. (Note that Eucalyptus will flush the 'filter' and 'nat' tables upon boot).
- A range of IP addresses must be available for use by Eucalyptus.
- Front end must have installed DHCP server daemon compatible with ISC DHCP Daemon version 3.0.X.

Limitations

None.

Configuration

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 not be required depending on your installation, as follows):

```
VNET_MODE="MANAGED"
VNET_PUBINTERFACE
VNET_PRIVINTERFACE
VNET_DHCPDAEMON
#VNET_DHCPUSER
VNET_SUBNET
VNET_NETMASK
VNET_DNS
VNET_ADDRSPERNET
VNET_PUBLICIPS
#VNET_CLOUDIP
#VNET_LOCALIP
```

On each node:

```
VNET_MODE="MANAGED"
VNET_PUBINTERFACE
VNET_PRIVINTERFACE
```

The Eucalyptus administrator must configure the front-end machine's `eucalyptus.conf` file, for the appropriate Ethernet devices, as follows:

First, with a valid Ethernet device attached to the public network (let's say `eth0`), configure as follows:

```
VNET_PUBINTERFACE="eth0"
```

Then, you must specify the Ethernet device on the physical network shared with the nodes. This could be the Ethernet device attached to the public network (`eth0`), or, if this is a second device, (let's say `eth1`), you would configure as follows:

```
VNET_PRIVINTERFACE="eth1"
```

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

```
VNET_PUBINTERFACE="eth0"
```

To configure Eucalyptus for your DHCP, see the above section: *1. Networking Features and Requirements, DHCP server*.

It is also necessary to instruct Eucalyptus about the unused subnet. To do this, you must configure `VNET_SUBNET`, `VNET_ADDRESSPERNET`, etc., as described in the above section: *About VNET_ options*.

Caveats

In MANAGED mode, Eucalyptus will flush the front-end machine'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 on 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 if you are completely sure that it will not interfere with the operation of Eucalyptus.

```
[root@clc]# iptables-save > $EUCALYPTUS/var/run/eucalyptus/net/iptables-preload
```

Configuration Example

For our MANAGED mode example, we use the network configuration shown in **Figure 2**. To configure this mode properly, we must choose an unused private network for our instances. Since the 10.0.1.0/24 subnet is being used, we specify an alternate subnet 192.168.0.0/16.

The following IP addresses are available for use by instances in Eucalyptus: 173.205.188.131-173.205.188.150

On the front-end machine:

```
VNET_DHCPDAEMON="/usr/sbin/dhcpd"
VNET_PUBINTERFACE="eth0"
VNET_PRIVINTERFACE="eth1"
VNET_MODE="MANAGED"
VNET_SUBNET="192.168.0.0"
VNET_NETMASK="255.255.0.0"
VNET_DNS="173.205.188.129"
VNET_ADDRSPERNET="32"
VNET_PUBLICIPS="173.205.188.131-173.205.188.150"
```

On the nodes:

```
VNET_MODE="MANAGED"
VNET_PUBINTERFACE="eth0"
```

At this point, we show you how to determine the actual number of active security groups. In our example there are 65536 addresses available (192.168.0.0/16). If we divide by the number of addresses per network (set to 32 below), we find the maximum number of subnets is (65536 / 32 = 2048). We also need to know the number of VLANs available to Eucalyptus. You can find this in the Web UI under the configuration tab > Clusters > Use VLAN tags. Let's assume you have the default range of VLANS from 10 to 4095. You can now calculate the number of security groups:

```
#security groups = min(VLAN-end, #subnet) - VLAN-start
```

In our example,

```
min(4095, 2048) - 10 = 2038
```

Thus we can have 2038 active security groups at any given time.

3.4.2 MANAGED-NOVLAN Mode

In this mode, Eucalyptus will fully manage the local VM instance network and provides all of the networking features Eucalyptus currently supports, including security groups, elastic IPs, etc., but it does not provide VM network isolation.

Requirements

- There is an available range of IP addresses to be used for the virtual subnets that do not interfere with the physical network. Typically these IP addresses are selected from the private IP ranges: 192.168.x.x, 10.x.x.x, etc.
- You are not running a firewall on the front end (CC) or your firewall is compatible with the dynamic changes performed by Eucalyptus when working with security groups. (Note that Eucalyptus will flush the 'filter' and 'nat' tables upon boot).
- A range of public IP addresses must be available for use by Eucalyptus.
- Front end must have installed DHCP server daemon compatible with ISC DHCP Daemon version 3.0.X

Limitations

- Layer-3 only VM isolation

Configuration

The options in `eucalyptus.conf` that must be configured correctly in MANAGED-NOVLAN mode are as follows:

For configuration of the front-end machine, see the front-end configuration for MANAGED mode, which is identical to MANAGED-NOVLAN.

On each node:

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

Nodes must have VNET_BRIDGE set properly:

```
VNET_BRIDGE="xenbr0"
```

Caveats

See the caveats for MANAGED mode in the preceding section.

Configuration Example

For this example, we refer to the network configuration shown in **Figure 2**. In this case the bridge on the node is “xenbr0.”

On the front-end machine:

```
VNET_DHCPDAEMON="/usr/sbin/dhcpd"
VNET_PUBINTERFACE="eth0"
VNET_PRIVINTERFACE="eth1"
VNET_MODE="MANAGED-NOVLAN"
VNET_SUBNET="192.168.0.0"
VNET_NETMASK="255.255.0.0"
VNET_DNS="173.205.188.129"
VNET_ADDRSPERNET="32"
VNET_PUBLICIPS="173.205.188.131-173.205.188.150"
```

On the node(s):

```
VNET_MODE="MANAGED-NOVLAN"
VNET_BRIDGE="xenbr0"
```

3.4.3 System Mode

There is very little Eucalyptus configuration required to use SYSTEM mode—Eucalyptus mostly stays 'out of the way' in terms of VM networking.

Requirements

- The Ethernet device on the nodes that communicates with the CC must be bridged.
- A pre-existing DHCP server must be running and configured.

Limitations

- No Elastic IPs
- No Security Groups
- No VM isolation

Configuration

The options in 'eucalyptus.conf' that must be configured correctly in SYSTEM mode are as follows:

On the front end and node(s)

```
VNET_MODE="SYSTEM"
```

In each Eucalyptus node controller's (NC) 'eucalyptus.conf' file, make sure that the parameter 'VNET_BRIDGE' is set to the name of the bridge device that is connected to your local Ethernet:

```
VNET_BRIDGE="xenbr0"
```

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. Note that your front-end machine does not need to have any bridges (this is fine, as VNET_BRIDGE is only relevant for node controllers, and will be safely ignored by the front-end components).

Configuration Example

For our SYSTEM mode example we use the configuration illustrated in **Figure 1**. In this example, the node has a bridge called xenbr0. The following is the proper configuration parameters for eucalyptus.conf:

On front-end machine:

```
VNET_MODE="SYSTEM"
```

On node machine(s):

```
VNET_BRIDGE="xenbr0"
VNET_MODE="SYSTEM"
```

3.4.4 Static Mode

In this mode, Eucalyptus manages VM IP address assignment by maintaining its own DHCP server with one static entry per VM.

Requirements

- The Ethernet device on the nodes that communicates with the CC must be bridged.
- A range of IP addresses must be available for use by Eucalyptus.
- NO pre-existing DHCP server on subnet. (Or, existing DHCP server must be configured to *not* serve instances.)
- Front end must have installed (but not configured or running) DHCP server daemon compatible with ISC DHCP Daemon version 3.0.X.

Limitations

- No Elastic IPs
- No Security Groups
- No VM isolation

Configuration

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 not be required depending on your installation, see below for details):

```
VNET_MODE="STATIC"
VNET_PRIVINTERFACE
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 machine's `eucalyptus.conf` file first with a valid, configured Ethernet device that is attached to the same physical Ethernet as the Eucalyptus nodes:

```
VNET_PRIVINTERFACE="eth0"
```

It is also necessary to instruct Eucalyptus about the subnet being used by the CC and NC. To do this, you must configure `VNET_SUBNET`, etc., as described earlier in Section 3.3: About `VNET_` options. (Note that `VNET_ADDRESSPERNET` is not used in this mode). In addition, you must specify the router on your subnet (`VNET_ROUTER`), and you must provide a list of static MAC addresses/IP addresses (`VNET_MACMAP`), as shown in the following configuration example:

Configuration Example:

For our STATIC mode example we refer to **Figure 1**. As with SYSTEM mode, the nodes have a bridge named `xenbr0`. The following IP addresses are available for use by instances in Eucalyptus: 173.205.188.133-173.205.188.135.

On the front-end machine:

```
VNET_DHCPDAEMON="/usr/sbin/dhcpd"
VNET_PRIVINTERFACE="eth0"
VNET_MODE="STATIC"
VNET_SUBNET="173.205.188.0"
VNET_NETMASK="255.255.255.0"
VNET_BROADCAST="173.205.188.255"
VNET_ROUTER="173.205.188.1"
VNET_DNS="173.205.188.129"
VNET_MACMAP="AA:DD:11:CE:FF:ED=173.205.188.133
AA:DD:11:CE:FF:EE=173.205.188.134 AA:DD:11:CE:FF:EF=173.205.188.135"
```

On the nodes:

```
VNET_BRIDGE="xenbr0"
VNET_MODE="STATIC"
```

4. Multi-cluster Networking

Eucalyptus EE supports multiple clusters within a single Eucalyptus cloud. This section briefly describes how Eucalyptus manages the networking aspect of a multi-cluster setup. First, in SYSTEM or STATIC networking modes, Eucalyptus does not perform any special configuration for a multi-cluster setup. In MANAGED and MANAGED-NOVLAN modes, Eucalyptus will set up layer-two tunnels between your clusters, so that virtual machines that are in the same security group, but distributed across clusters (potentially each in their own broadcast domain), can communicate with one another. We use the 'vtun' package to handle all layer-two tunneling between clusters.

In most cases, if 'vtun' is installed on each of your CCs, multi-cluster tunneling is automatically handled by each CC.

Caveats

Depending on your networking mode and network topology, you will want to keep the following caveats in mind with respect to your Eucalyptus network configuration:

MANAGED mode - During normal operation, you will see many tunnel interfaces being created and destroyed as virtual networks are constructed and torn down.

MANAGED-NOVLAN - Your CC will need to be configured with a bridge as its private interface (VNET_PRIVINTERFACE) in order for vtun tunneling to work in this mode.

MANAGED and **MANAGED-NOVLAN** - The CC attempts to auto-discover its list of local IP addresses upon startup, but if the IP that was used to register the CC is not locally available, you can override the CC's notion of 'self' by setting the 'VNET_LOCALIP' variable in `eucalyptus.conf`.

MANAGED and **MANAGED-NOVLAN modes** - Do not run two CCs in the same broadcast domain with tunneling enabled, this will potentially lead to a broadcast storm as tunnels start forwarding packets in a loop on your local network.

If you wish to disable tunneling altogether, set 'VNET_LOCALIP=0.0.0.0' in `eucalyptus.conf`.

5. Network Configuration for Components on Separate Machines

If your cluster controller (CC) and cloud controller (CLC) are running on separate hosts, the following value needs to be set within the CC's configuration file:

VNET_CLOUDIP=<ip-of-cloud-controller>

If you are running multiple clusters, you may wish to explicitly specify the IP address that the CC used to register with the CLC. You may set the variable within the configuration file for each of the CCs.

VNET_LOCALIP=<ip-of-cluster-controller>

If the `VNET_LOCALIP` value is not set, the CC will attempt to determine this value, automatically.

EBS Configuration

Block-based storage, or EBS (elastic block store), functionality of Eucalyptus allows raw block devices to be attached to instances as volumes. Users typically create a partition and a filesystem on the EBS attached volume. Data stored on EBS volumes persists after an instance is terminated.

EBS in Eucalyptus is exposed to VM instances using two technologies,

- **iSCSI** is a Layer-3 block-level storage technology. To use it, set `DISABLE_ISCSI="N"` in the `eucalyptus.conf` file on your Storage Controller host. Note: If you are using Rhel/Centos, Fedora, or OpenSUSE, in `etc/sudoers`, `Defaults requiretty` must be commented out (`#Defaults requiretty`).
or
- **AoE** (ATA-over-Ethernet) is a Layer-2 block-level storage technology. To use it, set `DISABLE_ISCSI="Y"` in the `eucalyptus.conf` file on your Storage Controller host.

AoE is attractive to users whose guest VMs/instances run in the same broadcast domain as the Eucalyptus Storage Controller server, whereas, iSCSI is a layer 3 technology and can be used across broadcast domains.

In addition to the choice of the underlying technology used to expose EBS volumes, the cloud administrator may specify a number of limits and configuration options, via the Eucalyptus web interface, which include,

- Maximum volume size per volume.
- Total disk space reserved for volumes.
- Total space reserved for snapshots.
- Interface to export volumes on (AoE only).
- Whether or not volumes should be zero filled.
- Path to volumes on disk.

Management

This section of the Administrator's Guide describes tasks that can be performed on a completed Eucalyptus installation, whether it was installed from source or from packages.

Managing Eucalyptus Images (2.0)

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.

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

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!):

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

Next, add the root filesystem image to Walrus:

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

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:

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

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 'euca-bundle-image' step

```
euca-bundle-image -i <vm image file> --kernel <eki-XXXXXXX> --ramdisk <eri-XXXXXXX>
```
2. A user may choose a specific kernel/ramdisk at instance run time as an option to 'euca-run-instances'

```
euca-run-instances --kernel <eki-XXXXXXX> --ramdisk <eri-XXXXXXX> <emi-XXXXXXX>
```
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-xxxxxxx> and optionally an <eri-xxxxxxx> as the defaults kernel/ramdisk to be used.

Deleting Images

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

```
euca-deregister <emi-XXXXXXX>
```

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

```
euca-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:

```
euca-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 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
```

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

```
euca-bundle-image -i euca-ubuntu-9.04-x86_64/kvm-kernel/initrd.img-2.6.28-11-generic --ramdisk true
euca-upload-bundle -b ubuntu-ramdisk-bucket -m /tmp/initrd.img-2.6.28-11-generic.manifest.xml
```

```
euca-register ubuntu-ramdisk-bucket/initrd.img-2.6.28-11-generic.manifest.xml
(set the printed eri to $ERI)

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

Web-Based Management

Several management tasks can be performed directly through the Eucalyptus Web interface, after logging in with appropriate administrative privileges.

User Management

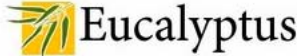
User sign-up

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.



Please, fill out the form:

Mandatory fields:

Username:	<input type="text" value="john"/>
Password:	<input type="password" value="*****"/>
Password, again:	<input type="password" value="*****"/>
Full Name:	<input type="text" value="John Doe"/>
Email address:	<input type="text" value="johndoe@example.com"/>

Optional fields:

Telephone Number:	<input type="text"/>
Project Leader:	<input type="text"/>
Affiliation:	<input type="text"/>
Project Description:	<input type="text"/>

Or

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.



Thank you!

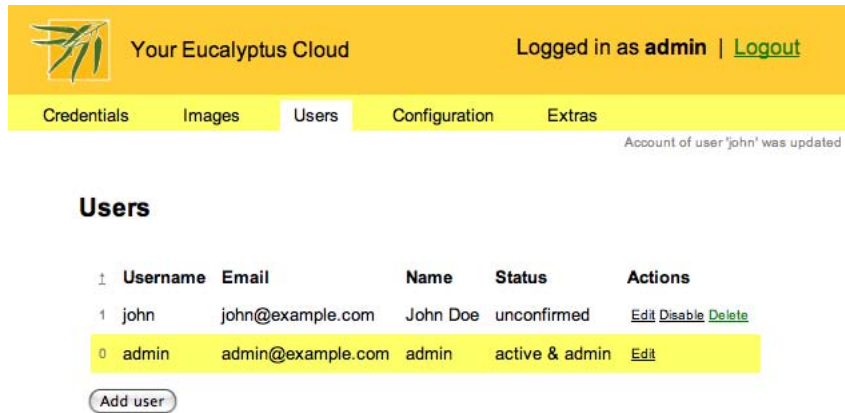
Thank you for signing up! Your request has been forwarded to the cloud administrator. If your application is approved, you will receive an email message (at the address you specified) with instructions for activating your account.

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.

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.



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.

Command-Line-Based Management

Command-line tools offer an alternative to Web-based management. A few tasks can only be done from the command line.

Managing Nodes

Once you have a running Eucalyptus system you can add and remove nodes (hosts running Node Controllers) by executing on a Cluster Controller machine:

```
$EUCALYPTUS/usr/sbin/euca_conf --register-nodes "<nodename1> ... <nodenameN>"
```

You will be asked for password to login to <nodenameX>: this is needed to propagate the cryptographic keys. Similarly, to remove a node, execute:

```
$EUCALYPTUS/usr/sbin/euca_conf --deregister-nodes "<nodename1> ... <nodenameN>"
```

Adding Users

Warning: Please note that boto must be installed for the following commands to function properly. One method of ensuring this is to install Euca2ools.

The following commands assume that the environment variables exported by a 'eucarc' file for an administrative Eucalyptus user have been set.

Users can be added from the command-line with:

```
$EUCALYPTUS/usr/sbin/euca-add-user USERNAME
```

possibly with options `--email=EMAIL` and `--admin` (all other properties for a user must be set through the Web interface currently).

To delete a user:

```
$EUCALYPTUS/usr/sbin/euca-delete-user USERNAME
```

Backup of Eucalyptus (2.0)

Backing up and restoring a Eucalyptus installation involves saving and restoring the contents of *five file-system locations*. Three are on the CLC machine:

- The **configuration file** (`$EUCALYPTUS/etc/eucalyptus.conf`)
- The **database files** (`$EUCALYPTUS/var/lib/eucalyptus/db`)
- The **cryptographic keys** (`$EUCALYPTUS/var/lib/eucalyptus/keys`)

One on the Walrus machine (which is the same as CLC machine in a single-cluster installation):

- The **Walrus buckets** ("Buckets path" in Web configuration, by default `$EUCALYPTUS/var/lib/eucalyptus/bukkits`)

And one on each of the cluster head nodes (again, same as the CLC machine in a single-cluster installation):

- The **SC volumes** ("Volumes path" in Web configuration, by default \$EUCALYPTUS/var/lib/eucalyptus/volumes)

If the files at these locations are backed up, a Eucalyptus installation can be fully restored after a crash or a failed upgrade. What follows is a step-by-step guide to backup and restoration.

Part I: Backup

1. Clean up Eucalyptus running state

- Note the value of the "Buckets path" and "Volumes path" for each cluster, listed under the "Configuration" tab of the Web interface. (That is where all uploaded images, user buckets, user volumes, and snapshots are located.)

- Terminate **all** Eucalyptus instances on **all** nodes

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

- Shut down all Eucalyptus components on **all** nodes, issuing the commands relevant for a node:

```
$EUCALYPTUS/etc/init.d/eucalyptus-nc stop
$EUCALYPTUS/etc/init.d/eucalyptus-cc cleanstop
$EUCALYPTUS/etc/init.d/eucalyptus-cloud stop
```

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

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

2. Back up the current installation

- Calculate the disk space required to store the files about to be backed up (this is most relevant for buckets and volumes, which can be large). E.g., on a single-cluster installation with default Buckets and Volumes paths:

```
du -sh $EUCALYPTUS/var/lib/eucalyptus/
```

- Create a directory for storing these (\$BACKUP) on a volume with enough disk space

```
export BACKUP=/path/to/backup/directory
mkdir -p $BACKUP
```

- Mirror the five locations, taking care to preserve the permissions on all files. E.g., on a single-cluster installation with default Buckets and Volumes paths:

```
cp -a $EUCALYPTUS/etc/eucalyptus/eucalyptus.conf \
$EUCALYPTUS/var/lib/eucalyptus/keys \
$EUCALYPTUS/var/lib/eucalyptus/db \
$EUCALYPTUS/var/lib/eucalyptus/bukkits \
$EUCALYPTUS/var/lib/eucalyptus/volumes \
$BACKUP
```

or, alternatively, with `tar`:

```
tar cvf $BACKUP/eucalyptus-backup.tar \
$EUCALYPTUS/etc/eucalyptus/eucalyptus.conf \
$EUCALYPTUS/var/lib/eucalyptus/keys \
$EUCALYPTUS/var/lib/eucalyptus/db \
$EUCALYPTUS/var/lib/eucalyptus/bukkits \
$EUCALYPTUS/var/lib/eucalyptus/volumes
```

In either case, be careful if any of the above are symbolic links as they may be copied instead of the directories they point to. Check that the backup indeed contains files from the original locations.

Part II: Roll back to a previously installed version

Note: Please ONLY execute these steps if you wish to revert to your previously installed Eucalyptus version.

1. Clean up Eucalyptus running state

Same as in the Backup step, make sure no Eucalyptus components are running on any of the nodes

2. Optionally update/downgrade Eucalyptus-related binary packages

If you are trying to recover from a broken upgrade by rolling back or by trying the upgrade again, this would be the right time to

- remove all software components related to Eucalyptus (e.g., `rpm -e` or `apt-get remove`) and
- install the appropriate version by following the instructions in the Installation section.

Warning: DEBs will restart the services: be sure you stop them again before copying back the backed-up files.

3. Replace the saved state

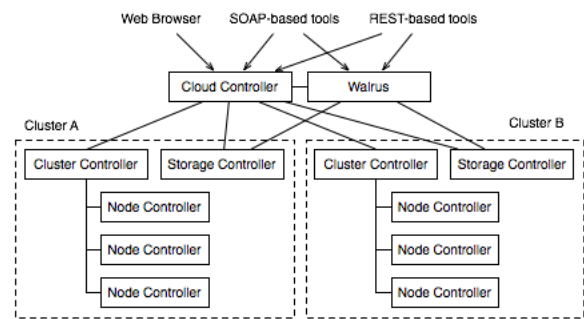
- Depending on how you backed up, copy the files back either with `cp`:

```
cp -a $BACKUP/eucalyptus.conf $EUCALYPTUS/etc/eucalyptus
cp -a $BACKUP/keys $BACKUP/db $BACKUP/bukkits $BACKUP/volumes $EUCALYPTUS/var/lib/eucalyptus
```

or with `tar`:

```
cd $EUCALYPTUS
tar xvf $BACKUP/eucalyptus-backup.tar
```

Advanced Setups



A Eucalyptus setup consists of a number of web services components -- The Cloud Controller (**CLC**), Walrus, Storage Controller (**SC**), Cluster Controller (**CC**) and the Node Controller (**NC**). In a single cluster setup, all front end web services (all services except the NCs) run on a single physical host. In a more advanced configuration, you can choose to run the CLC, Walrus, SC and CC on a separate physical machines, or you can combine them as you see fit. For example, one reason to separate Eucalyptus components is to improve the overall performance of the system by distributing different types of work (i.e. place Walrus on a machine that has a fast disk subsystem, while the CLC can be placed on another machine with a fast CPU).

In addition, in a multi-cluster setup, you may configure more than one cluster with Eucalyptus, either for performance or management purposes. This guide briefly describe multi-component and multi-cluster setups. In a multi-cluster

installation, there must be a single Cloud Controller, a single Walrus, one Cluster Controller and Storage Controller pair per cluster, and one or more Node Controllers grouped within each cluster, as illustrated in the adjacent figure.

We assume that you familiar and comfortable with single cluster Eucalyptus installation, configuration, management and usage before proceeding to more advanced configurations. We also assume that you are able to install individual Eucalyptus components on the distribution and architecture of your choice. Please refer to the previous installation sections of the Administrator's Guide for information regarding distribution specific installation methods, package names, dependencies, etc.

Multi-component and Multi-cluster Setup

A Eucalyptus installation consists of five components: A Cloud Controller (CLC), Walrus, one or more Cluster Controller (CC) and Storage Controller (SC) pairs, and one or more Node Controllers (NC). Once you have the components installed on your physical hosts, using the topology of your choice, you're ready to enable the services and stitch together your Eucalyptus cloud by 'registering' the components with one another.

Enabling services

If you installed Eucalyptus from source and rsynced the build to your physical hosts, you will need to explicitly enable or disable the services that you wish to run (all are disabled by default). This can be done by running

```
$EUCALYPTUS/usr/sbin/euca_conf {--enable,--disable} cloud
$EUCALYPTUS/usr/sbin/euca_conf {--enable,--disable} walrus
$EUCALYPTUS/usr/sbin/euca_conf {--enable,--disable} sc
```

to enable and disable the CLC, Walrus and SC respectively. CLC, Walrus and SC execution and termination is controlled using the same script, "\$EUCALYPTUS/etc/init.d/eucalyptus-cloud {start,stop,restart}"

The CC is controlled using "\$EUCALYPTUS/etc/init.d/eucalyptus-cc {start,stop,restart,cleanstart,cleanstop,cleanrestart}"

If you have installed from packages, service enabling is done during package install, and service startup may have been done depending on your packaging method. If you install multiple services on the same host, but do not wish to enable them all, you can "--disable" the service, followed by a 'restart' of 'eucalyptus-cloud'.

After the services are up and running, you'll need to stitch together your Eucalyptus cloud by registering them, so that they can start to communicate with one another.

Registering Services

First, you will need to inform the Cloud Controller (CLC) the location of Walrus. To do this, run the following command on the **CLC host**.

```
$EUCALYPTUS/usr/sbin/euca_conf --register-walrus <Walrus IP address>
```

where "Walrus IP address" is the IP address of the host on which you have installed Walrus. In case your host has multiple IP addresses, pick the one that is visible from the CLC and clients, and note that 'localhost' or '127.*.*.*' are not valid Walrus endpoint addresses, even

if your Walrus service is running on the same host as your CLC.

Next, you will need to register a Cluster Controller (CC) and a Storage Controller (SC) with the CLC. There is one CC/SC pair per cluster.

To do so, run the following commands on the **CLC host**,

```
$EUCALYPTUS/usr/sbin/euca_conf --register-cluster <clustername> <CC IP address>
$EUCALYPTUS/usr/sbin/euca_conf --register-sc <clustername> <SC IP address>
```

where "CC IP address" and "SC IP address" are IP addresses of the CC host and the SC host respectively and "clustername" is the name you want to use for your cluster. Again, note that 'localhost' and '127.*.*.*' addresses are invalid.

If you are installing the Cloud Controller (CLC), Cluster Controller (CC), Walrus and the Storage Controller (SC) on same machine, the IP address above will be the same in all steps.

If you setting up multiple clusters, you will need to install a CC and SC per cluster and you will need to run the above "--register-cluster" and "--register-sc" commands for each cluster.

If you are setting up multiple clusters, the networking options on each CC must be identical with the exception of VNET_PUBLICIPS, which may be set to CC specific values. In addition, if you are running in MANAGED-NOVLAN networking mode, then each CC's VNET_PRIVINTERFACE must be set to a valid bridge that is configured and running properly before the Eucalyptus CCs are started.

Finally, you need to register nodes with the CC. To do so, run the following command on the **CC host**

```
$EUCALYPTUS/usr/sbin/euca_conf --register-nodes "<Node0 IP address> <Node1 IP address> ... <NodeN IP address>"
```

where "<NodeX IP address>" is the IP address of a host that is running a Node Controller (NC).

In a multi-cluster setup, you will need to run this command on each CC in your setup.

At this point, you have successfully registered Eucalyptus components and you are ready to proceed to configuration.

Monitoring

Eucalyptus 2.0 provides some facilities to help in monitoring the status of running components, of running VMs and statistics on storage uses.

We provide two example script to integrate with nagios and ganglia. Nagios will be able to monitor the status of the machine running Eucalyptus components as well as the components itself, while ganglia can provide more information on the resource usage.

The examples we provides, requires a running installation of nagios and ganglia. For example on debian/ubuntu, one can start followign nagios installation until the monitoring page is accessible from the web browser. After that, one can run the example script (followig the instruction in README.Monitoring) to add the rules to monitor Eucalyptus, and restarting nagios will provide basic monitoring facilities.

Similarly for ganglia, one has to have a running ganglia installation, after which the appropriate script (provided in the directory extras) will injects some extra informations which will eventually be visible. It can take sometime to have the information to show up on ganglia's graphs.

The scripts can be used as a template for other monitoring infrastructures.

Setting up Dynamic DNS

The Eucalyptus cloud front end has a Domain Name System (DNS) service built into it that will respond to DNS requests in order to support virtual hosting of buckets (mapping bucket names to IP addresses) and instance DNS (mapping of hostnames to instance public and private IP addresses).

Enabling DNS in Eucalyptus

Since most Linux distributions have "dnsmasq" or "bind" or another DNS server running on the same physical host as the CLC, DNS in Eucalyptus is disabled by default. The Eucalyptus administrator can enable DNS after disabling or re-configuring any other services that listen on port 53.

To do so, edit \$EUCALYPTUS/etc/eucalyptus/eucalyptus.conf on the CLC machine and change the value of "DISABLE_DNS" to,

```
DISABLE_DNS=N
```

then, restart "eucalyptus-cloud".

Note: To verify that the DNS service is running, try running "netstat -al" and look for 53. Port 53 should be bound by the process "eucalyptus-cloud"

Configuring DNS in Eucalyptus

First, login to the front end admin web interface (<https://<front end IP>:8443>) and click on the "Configuration" tab.

Next, pick a subdomain within your domain that the Eucalyptus DNS system will service. Set the nameserver IP to the IP of the cloud front end that is accessible by your master DNS system. Then, set the domain that the Eucalyptus front end will service to <subdomain>. <domain>

Finally, you will need to change your master DNS configuration to point to the CLC public IP (entered in the above step) as the nameserver for your chosen subdomain within your organization. Optionally, you can change your client's config (/etc/resolv.conf on Linux) to point to the cloud front end IP directly, but note that you will have to do this on each system that needs to access the Eucalyptus DNS service.

Using DNS

After DNS is setup correctly, creation and deletion of buckets will automatically create DNS entries. The DNS service embedded in the CLC will respond with the Walrus IP when a DNS request is made for <bucketname>.walrus.<subdomain>.

In addition, instances IPs will be mapped as euca-A.B.C.D.eucalyptus.<subdomain>, where A.B.C.D is the IP address (or addresses) assigned to your instance.

Troubleshooting Eucalyptus

1.0 How to Troubleshoot Eucalyptus

To troubleshoot Eucalyptus, the administrator must know the location of the Eucalyptus components, that is, on which machine each component is installed. The administrator must have root access to each machine hosting the components and must know the network configuration connecting the components.

Usually when an issue arises in Eucalyptus, you can find a clue or trace or record that suggests the nature of the problem either in the eucalyptus log files or in the system log files. Assuming Eucalyptus is installed in root (/), the eucalyptus logs are located on each machine hosting a component in the following directory: `/var/log/eucalyptus/`.

Here are the relevant logs for each component:

Cloud Controller (CLC), Walrus, and Storage Controller (SC):

- `cloud-debug.log`
- `cloud-error.log`
- `cloud-output.log`

Cluster Controller (CC):

- `cc.log`
- `axis2c.log`
- `httpd-cc_error_log`

Node Controller (NC):

- `nc.log`
- `axis2c.log`
- `httpd-nc_error_log`
- `euca_test_nc.log`

You can control the amount of information displayed in the logs by modifying variables in `eucalyptus.conf`. For the CLC, SC and walrus, you must modify the `CLOUD_OPTS` variable by adding the parameter `--log-level=LEVEL`. For the CC and NC, you must modify the variable `LOGLEVEL=LEVEL`. The possible values for *LEVEL* are: `DEBUG`, `INFO`, `WARN`, `ERROR` and `FATAL`. After changing these values, you must restart the components for the changes to take effect.

In addition, information regarding the nature of an issue may appear in the system's logs. In particular, you might want to search for clues in `/var/log/xen/`.

It is also important to understand the elements of the network on your system. For example, you may wish to list bridges to see which devices are enslaved by the bridge. To do so, use the `brctl` command. You may also want to list network devices and evaluate existing configurations. To do so, you can use these commands: `ip`, `ifconfig`, and `route`. You can also use `vconfig`, if, for example, you wish to evaluate VLAN configuration (MANAGED mode only).

Administrator credentials allow access to more information than user credentials. For example, with administrator credentials `euca-describe-instances` gives you additional information, including all instances running by all users on the system. Thus, make sure you have

Euca2ools installed with proper administrator credentials.

2.0 Common Eucalyptus Problems and Solutions

Here we provide troubleshooting strategies and solutions for these commonly occurring issues in Eucalyptus:

- Are all Eucalyptus components registered?
- Is Eucalyptus running?
- Is the CC running correctly?
- Are the NCs running correctly?
- Is libvirt configured correctly?
- What if `euca-describe-availability-zones verbose` returns 000/000?
- What if I cannot allocate elastic IPs?
- What if `euca-run-instances` returns not enough resources?
- How do I check that my linux-based instance is fully booted?
- What if my instance stays in "pending" state?
- What if I cannot ssh into the instance?
- What if I receive a WARNING when I try to ssh into the instance?
- What if components are not communicating with each other?
- Is there enough disk space on walrus?
- Can CPUs (cores) be overcommitted?
- Can memory be overcommitted?
- How do I debug an image?
- What if my very large-size image won't start?
- What if `euca-upload-bundle` fails?
- What if I cannot create EBS volumes or snapshots?
- What if my EBS volume will not attach (AoE)?
- What if Eucalyptus fails to start the DHCP server?
- What if my images are not reachable?
- What if my instance reports public and private IP as 0.0.0.0 (SYSTEM mode only)?
- My interface lost its address and now shows 169.254.169.254. What happened?

Are all Eucalyptus components registered?

You can use the `euca_conf` to check that all components are registered correctly. To do so, on the CLC machine (as root user) run these commands:

- `euca_conf --list-clusters`
- `euca_conf --list-scs`
- `euca_conf --list-walruses`
- `euca_conf --list-nodes`

Check that the IP addresses returned are consistent with your network configuration. For example, Walrus should be registered with a public IP, not localhost (127.0.0.1).

 [TOP](#)

Is Eucalyptus running?

You can quickly check to confirm that the CLC is running, by accessing the Web UI (**`https://<IPAddress>: 8443`**). Once you've confirmed the CLC is running, check to see that the components are correctly registered (see above). A very useful high-level check can be performed with `euca-describe-availability verbose` (with admin credentials), which will indicate if your cloud resources are available. The output of the command will indicate the maximum capacity of your cloud installation for each VM Type (e.g., `m1.small`, `c1.medium`, `m1.large`, etc.) and the current availability of each VM type. The following example shows the cloud is unloaded and all resources are available.

```
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
```

 [TOP](#)

Is the CC running correctly?

First, check that the CC has been started and registered (as describe above). Next, check on the CC machine to confirm that the `cc.log` is growing (i.e., the CLC is polling the CC). If not, the registration was not successful for several possible reasons, including an incorrect key, wrong IP address, firewall impediment, etc. You may also want to inspect the other eucalyptus log files on the CC.


[TOP](#)

Are the NCs running correctly?

First, check that the CC is running correctly (see above). Next, check that the NC has been started and registered with the correct CC (in the event you have more than one CC). Now, check the `cc.log` on the CC to confirm the CC is polling the NC. (If not, the node may not be registered correctly). Now check the `nc.log` to confirm the NC is being polled by the CC. (If not, check the eucalyptus log files on the NC machines for errors (e.g., incorrect keys, cannot talk to hypervisor, libvirt misconfigured etc.).


[TOP](#)

Is libvirt configured correctly?

For information on proper configuration of libvirt, see Hypervisor Configuration


[TOP](#)
What if `euca-describe-availability-zones verbose` returns 000/000?

Follow the steps in the previous troubleshooting solutions above: Check that the CC, NC, and CLC are running correctly. Next, check that there are enough resources available (for example disk space) on the NC machines and that they are accessible to the user “eucalyptus” (for example the disk space is accessible).


[TOP](#)

What if I cannot allocate elastic IPs?

First, check use the `euca-describe-addresses` command to see if there is available IPs. If not examine your configuration, in particular the value of VNET_PUBLICIPS (see *Section 8: Eucalyptus EE Networking Configuration*).

If all IPs are taken, you may need to allocate more IPs to Eucalyptus. If IPs are available, but you still get errors, you may need to perform a clean restart of the CC.


[TOP](#)
What if `euca-run-instances` returns not enough resources?

Use the `euca-describe-availability-zones verbose` command to confirm that you have available resources. If you do have resources available, check that you also have available public IP addresses. (Try allocating and de-allocating an IP Address). Next, check that the root file system of the image you want to run fits with the size of the instance type you are using.


[TOP](#)

How do I check that my linux-based instance is fully booted?

If you use KVM, use `euca-get-console-output` to get the console output of the instance. If you use XEN and you get an error, log into the NC machine as root and use the `xm console` command to get the console output.

Now, check in the instance console output to confirm that the instance is booted (that is the instance shows the kernel messages and that there are no errors mounting the root file system).



TOP

What if my instance stays in “pending” state?

If your image is very big it may take a very long time to boot. To check for errors in the preparation of the instance, log into the NC as root, and check the `nc.log` for information about your instance. Reasons for the failure might include: Difficulty communicating with walrus (check in `$INSTANCE_PATH/<user>/<instance id>` to determine if the kernel/initrd and root are correct); errors in preparing the image (check in the `nc.log`); errors talking to libvirt/hypervisor (again check `nc.log`, libvirt logs, etc.).



TOP

What if I cannot ssh into the instance?

Make sure that the security group the instance is using allows ssh (port 22) connections from the client you are using. Check that the instance is fully booted (as explained above). Check that the network configuration for your mode is correct (in particular the `VNET_*INTERFACE` values).



TOP

What if I receive a WARNING when I try to ssh into the instance?

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

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

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

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

```
$ sed '11' d /home/bob/.ssh/known_hosts
```



TOP

What if components are not communicating with each other?

Check that there is no firewall between them. Check that the IP address used during configuration is correct. Check that there is connectivity between each of the machines hosting the components using the IP specified during configuration. Check that the components are running (as described above). Check also that each machine hosting components is running NTP and that the machines' internal clocks are synchronized.



TOP

Is there enough disk space on walrus?

Walrus deals with, possibly, very big images. The size of available disk space should be at least three times the size of the image you wish to upload. The reason is that the image needs to be uploaded, then decrypted before sending it to the NC, which requires in itself approximately twice the size of the

image. In addition, temporary files are created, so three-times the image size times is a safe amount to reserve.



TOP

Can CPUs (cores) be overcommitted?

By default, NCs allocate 1 real core/CPU per virtual core/CPU. That is, if an instance requires 2 cores/CPU, and the NC has only 2 cores/CPU then no more instances will be allowed on that NC. The NC's CPUs can be overcommitted using the `MAX_CORES` options in `eucalyptus.conf`. Note that you must restart the NC after modifying the value. (Note that performance may suffer when cores are overcommitted).



TOP

Can memory be overcommitted?

NO. Unlike the CPUs/cores, memory cannot be overcommitted. The total amount of memory that the hypervisors allocates to VMs cannot exceed the total amount of physical memory on the node.



TOP

How do I debug an image?

To debug an image as used by Eucalyptus: Set `MANUAL_INSTANCES_CLEANUP` to 1. In this case, when an instance fails, the temporary files (i.e., root file system, kernel, etc.) are not deleted. You can find these files at `$INSTANCE_PATH/<user>/<instanceId>` along with the `libvirt.xml` configuration file used to start the instance. You can then modify the `libvirt.xml` (the network part will need to be modified) and start the instance manually using `virsh create`.



TOP

What if my very large-size image won't start?

On the "Configuration" page of the Eucalyptus Web UI, under "Walrus configuration." confirm the "space reserved for unbundling images" is enough to contain your image. If not, increase the size of space reserved in the field provided. (Note that very large images can take a long time to boot).



TOP

What if `euca-upload-bundle` fails?

If you are trying to upload to an already existing bucket, Eucalyptus will return a "409" error. 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. Note: If you are using `Euca2ools`, this is not necessary. In addition, when using `ec2-upload-bundle`, make sure that there is no "/" at the end of the bucket name.



TOP

What if I cannot create EBS volumes or snapshots?

Make sure you have enough loopback devices. (Note that you should have received a warning when starting Eucalyptus components). On most distributions, the loopback driver is installed as a module. The following will increase the number of loopback devices available:

```
[root@clc]# rmmod loop ; modprobe loop max_loop=256
```



TOP

What if my EBS volume will not attach (AoE)?

AoE requires the SC and NCs to be on the same physical subnet. You can check and change the Ethernet device used by the SC to export the AoE volumes by modifying the "Storage Interface" field found in the "Storage Controller" section (on the Configuration page of the Eucalyptus Web UI). (Note that this problem will arise only when the machine hosting the SC has multiple Ethernet devices). AoE will not export to the same machine that the server is running on, which means that the SC and NC must be hosted on separate physical host machines.



What if Eucalyptus fails to start the DHCP server?

All networking modes, except SYSTEM, will start a DHCP server when instances are running. The CC log may report a failure to start the DHCP server. Or, you may notice upon starting an instance that the DHCP server is missing on the CC machine (You use the `ps` command to check for the presence of DHCP server). Also, make sure that your DHCP binary is compatible with ISC DHCP daemon 3.x and that the binary specified in VNET_DHCPDAEMON is correct. You may see errors in the `httpd-cc_error_log`.



What if my images are not reachable?

To check that your Eucalyptus installation is properly configured, we recommend first running a Eucalyptus-prepared image (downloadable via the "image" tab on the Eucalyptus Web interface). Check to see that your instance is fully booted (as described above). Check that the security group used by the instance allows for connectivity from the client. For example, if using ssh, port 22 should be open. You will also need to check in the `eucalyptus.conf` file for the values of the VNET_PRIVINTERFACE and VNET_BRIDGE (when applicable) on both the CC and NC machine(s) and that the Ethernet devices specified are on the same physical subnet. Check if DHCP server has started (as described above).

If you have a DHCP server on your LAN, it may be possible that the cloud controller's DHCP server not to provide an IP address to your instances. Since all the cloud instances have MAC addresses beginning with d0:0d you may want to tell your main DHCP server to ignore requests sent from these MAC addresses.



What if my instance reports public and private IP as 0.0.0.0 (SYSTEM mode only)?

The solution to this problem is to have your VM ping the CC. This will exercise the networking layer in your VM, and it will then acquire a valid IP address.



My interface lost its address and now shows 169.254.169.254. What happened?

You are probably using the `ifconfig` command to see the Ethernet device configuration, which only shows one address per interface. Please use the `ip addr show` command to see all addresses associated with the interface.

