Eucalyptus Administrator's Guide (2.0)

Export: /etc/init.d/eucalyptus-cc start

And on each of the compute nodes run:

Export: /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 SEUCALYPUS/etc/eucalyptus/eucalyptus.conf (controller, make sure to use the 'cleanstart', 'cleanstop', and/or 'cleanrestart' directives to the init script file (appears in /etc/rc.d/init.d/eucalyptus 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 `SV_VERSION` must be set to the version of Eucalyptus you wish to install. For example, you can set the value to 2.0.2 using bash:

Export: VERSION=2.0.2

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:

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

   Export: yum install -y ntp
   ntpdate.pool.ntp.org

2. 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):

   Export: yum install -y java-1.6.0-openjdk ant ant-nodeps dhcp bridge-utils perl-Convert-ASN1.noarch scsi-target

3. Node has a fully installed and configured installation of Xen that allows controlling the hypervisor via HTTP from localhost.

   Export: yum install -y xen
   sed --in-place 's#(xend-http-server no)/xend-http-server yes/# /etc/xen/xend-config.xmp
   sed --in-place 's#(xend-address localhost)/xend-address localhost)="/etc/xen/xend-config.xmp
   /etc/init.d/xend restart

4. 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:

1. 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:

   [EUCALYPUS]
   name=Eucalyptus
   baseurl=http://.../CentOS5/eucalyptus
   enabled=1
   gpgcheck=1
   gpgkey=http://.../CentOS5/eucalyptus/gpgkey
   
2. 

   Download and install the RPMs from the repository.
Eucalyptus Administrator's Guide (2.0)

`[eucast]`
`name=Eucalyptus`
`baseurl=http://www.eucalyptussoftware.com/downloads/repo/eucalyptus/${VERSION}/yum/centos/
enabled=1`

Now install Eucalyptus on the front-end:

`yum install eucalyptus-${VERSION}-core.${ARCH} eucalyptus-walrus.${ARCH} eucalyptus-sc.${ARCH} --no-
and install Eucalyptus on the node:

`yum install eucalyptus-${ARCH} --nogpgcheck`

where ${ARCH} is the architecture of your host (either 'i386' or 'x86_64').

1. 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://open.eucalyptus.com/downloads](http://open.eucalyptus.com/downloads) (look for a CentOS tarball of the right Eucalyptus version and architecture).

Untar the bundle in a temporary location:

```
tar xzvf 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-${VERSION}-rpm-deps-x86_64
rpm -Uvh eucalyptus-${VERSION}-rpm-deps-x86_64
rpm -Uvh acetoos-211.e14.x86_64.rpm \  
  eucal-axis2c-1.6.0-1.x86_64.rpm \  
  eucal-rampartc-1.3.0-1.x86_64.rpm \  
  vbld-14-1mdw2008-1.x86_64.rpm \  
  vtun-3.0.2-1.e15.rf.x86_64.rpm \  
  lzo2-2.02-3.e15.rf.x86_64.rpm \  
  perl-Crypt-OpenSSL-Random-0.94-1.e15.rf.x86_64.rpm\  
  perl-Crypt-OpenSSL-RSA-0.25-1.e15.rf.x86_64.rpm\  
  perl-Crypt-X509-0.32-1.e15.rf.noarch.rpm\  
  python25-2.5.1-basho1n1.x86_64.rpm\  
  python25-devel-2.5.1-basho1n1.x86_64.rpm\  
  python25-lib6-2.5.1-basho1n1.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-g1-${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 acetoos-211.e14.x86_64.rpm \  
  eucal-axis2c-1.6.0-1.x86_64.rpm \  
  eucal-rampartc-1.3.0-1.x86_64.rpm \  
  perl-Crypt-OpenSSL-Random-0.94-1.e15.rf.x86_64.rpm\  
  perl-Crypt-OpenSSL-RSA-0.25-1.e15.rf.x86_64.rpm\  
  perl-Crypt-X509-0.32-1.e15.rf.noarch.rpm\  
  python25-2.5.1-basho1n1.x86_64.rpm\  
  python25-devel-2.5.1-basho1n1.x86_64.rpm\  
  python25-lib6-2.5.1-basho1n1.x86_64.rpm
```

```
cd ..
```

then install the node controller RPM with dependencies:

```
rpm -Uvh eucalyptus-${VERSION}-x86_64.rpm \  
  eucalyptus-g1-${VERSION}-x86_64.rpm \  
  eucalyptus-sc-${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 file at:

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

Confirm that the following lines are uncommented, as shown:

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

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

```
su eucalyptus -c "virsh 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-no 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 requirements 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 2.0.2 using bash:

```
export VERSION=2.0.2
```

### Prerequisites

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

1. Front-end, node and client machine system clocks are synchronized (i.e. using NTP).
   
   ```
   ntp -p no -r pool.ntp.org
   yast2 -i ntp
   /etc/init.d/ntpd restart
   ```

2. 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
   ```

   and on the node
   
   ```
   zypper install vlan apache2 perl-Crypt-OpenSSL-RSA perl-Crypt-OpenSSL-Random tgt
   ```

3. 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.

4. 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
   ```

4. We recommend that you verify your Xen installation by manually bringing up a VM and testing that it has network connectivity using bridged networking.

4. 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).
there.

- **Important:** Verify that the Buckets Path and Volumes Path settings under the Configuration tab of the Web interface match 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).

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 backed up eucalyptus upgrade during a package-based upgrade, then after re-installing the old packages, copy back the saved backed up copies of db/*, keys/*, etc/eucalyptus/eucalyptus.conf to your restored installation. Then, start the system before.

Configuration

This section of the Administrator's Guide describes how to configure Eucalyptus, both during installation and after reconfiguring 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/bin/eucalyptus --register-walrus <front end IP address>
$EUCALYPTUS/usr/bin/eucalyptus --register-cluster <clustername> <front end IP address>
$EUCALYPTUS/usr/bin/eucalyptus --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/bin/eucalyptus --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: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 for 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 eucalyptus_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. 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 Eucalyptus tools, 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](http://www.linux-kvm.org/page/Virtio) or click [here](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 `vmm` 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.xsp
```

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)
(vmcpasswd ' ')
```

**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

```
```

17 von 40

13.01.2011 08:47
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/dhcpcd"

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 'dhcpcd' as is the case in Ubuntu 8.10 or later), then you must configure Eucalyptus to be aware of that user:

VNET_DHCPUSER="<dhcpcdUsername>"

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.

<table>
<thead>
<tr>
<th>NETWORKING MODE</th>
<th>PHYSICAL NETWORK REQUIREMENTS</th>
<th>EUCALYPTUS NETWORKING FEATURES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VLAN clean</td>
<td>Connectivity IP control Security Groups Elastic IPs Metadata service VM isolation</td>
</tr>
<tr>
<td>MANAGED</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MANAGED-NOVLAN</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>STATIC</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>SYSTEM</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

- 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 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@ec2]# brctl show
```
bridge name | bridge id | STP enabled | interfaces   
--- | --- | --- | --- 
virbr0 | 8000.000000000000 | yes |  
xebr0 | 8000.000000000000 | no | peth0  

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 each bridge is named xebr0, 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_ADDRESSSPERNET**

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_ADDRESSSPERNET 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_ADDRESSSPERNET 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):
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
neuca-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
neuca-upload-bundle -b <initrd bucket> -m /tmp/<initrd file>.manifest.xml
neuca-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-xxxxx> --ramdisk <eri-xxxxx>
   ```

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-xxxxxx> --ramdisk <eri-xxxxxx> <emi-xxxxxx>
   ```

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 `<cki-xxxxxx>` and optionally an `<ери-xxxxxx>` 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-xxxxxx>
```

Then, you can remove the files stored in your bucket. Assuming you have sourced your 'ecarc' 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 zxfv 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
neuca-upload-bundle -b ubuntu-kernel-bucket -m /tmp/vmlinuz-2.6.28-11-generic.manifest.xml
neuca-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
neuca-upload-bundle -b ubuntu-ramdisk-bucket -m /tmp/initrd.img-2.6.28-11-generic.manifest.xml
neuca-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
neuca-upload-bundle -b ubuntu-image-bucket -m /tmp/ubuntu-9.04.x86_64.manifest.xml
neuca-register ubuntu-image-bucket/ubuntu-9.04.x86_64.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

A lot of Eucalyptus management can be performed through the Web interface, after logging in with administrative privileges.