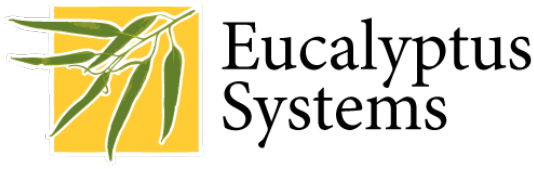
Eucalyptus®

Cloud Computing Platform

Administrator's Guide

Enterprise Edition 2.0



Eucalyptus Systems, Inc.

Eucalyptus Cloud Computing Platform
Administrator's Guide
Enterprise Edition 2.0

Table of Contents

Introduction

Eucalyptus Enterprise Edition (EE) 2.0 is a Linux-based software architecture that implements scalable, efficiency-enhancing private and hybrid clouds within an enterprise's existing IT infrastructure. Eucalyptus provides Infrastructure as a Service (IaaS). This means that users can provision their own collections of resources (hardware, storage, and network) via Eucalyptus' self-service interface on an as-needed basis. A Eucalyptus cloud is deployed across an enterprise's "on-premise" data center and is accessed by users over enterprise intranet. Thus sensitive data remains secure from external intrusion behind the enterprise firewall.

Eucalyptus was designed from the ground up to be easy to install and as non-intrusive as possible. The software framework is highly modular, with industry-standard, language-agnostic communication. Eucalyptus is also unique by providing a virtual network overlay that both isolates network traffic of different users and allows two or more clusters to appear to belong to the same Local Area Network (LAN). Eucalyptus also interoperates seamlessly with Amazon's EC2 and S3 public cloud services and thus offers the enterprise a hybrid cloud capability.

Initially developed to support the high performance computing (HPC) research of Professor Rich Wolski's research group at the University of California, Santa Barbara, Eucalyptus is engineered based upon design principles that ensure compatibility with existing Linux-based data center installations. Eucalyptus can be deployed without modification on all major Linux OS distributions, including Ubuntu, CentOS, openSUSE, and Debian. Ubuntu distributions now include the Eucalyptus software core as the key component of the *Ubuntu Enterprise Cloud.*

Who Should Read this Guide?

This guide is written for system, network, and cloud administrators tasked with deploying, managing, and maintaining the Eucalyptus EE 2.0 cloud platform.

What's in this Guide?

This guide contains comprehensive instructions and reference materials to assist you in setting up, installing, configuring, managing and maintaining the Eucalyptus EE 2.0 cloud platform. The following is an overview of its contents:

Part I: Welcome to Eucalyptus EE 2.0

Part I provides an introduction to the components of a Eucalyptus cloud, including Cloud Controller (CLC), Walrus (W), Cluster Controller (CC), Storage Controller (SC), Node Controller (NC) and VMware Broker (Broker). We also introduce you to the new features available in Eucalyptus EE 2.0, including Windows VM support, VMware support, SAN support, and User and Groups management and Accounting reports.

Part II: Deploying Your Eucalyptus Cloud

Part II shows you how to deploy the Eucalyptus cloud in three phases: (1) Setting up your environment (installation prerequisites); (2) Installing Eucalyptus dependency and component software; and (3) Registering and configuring your Eucalyptus cloud. Setup requirements are provided for each of the new Eucalyptus EE 2.0 optional features (Windows VM, VMware, and SAN support); and detailed instructions are provided for installing Eucalyptus EE 2.0 on CentOS 5.5 and openSUSE 11.2.

Part III: Managing Your Eucalyptus Cloud

Part III introduces you to the management features available in Eucalyptus EE 2.0. We begin with a tour of the Eucalyptus Web UI and proceed with a discussion of the new *Users and Groups* management interface —an easy graphical way for administrators to create groups, add users, and associate clusters with groups. Lastly, we present the new *Accounting Reports* interface that lets administrators view information on cloud activities and resource usage, which can be used as a basis for cloud auditing and resource management.

Part IV: Advanced Networking and Troubleshooting

Part IV discusses the four Eucalyptus networking modes: SYSTEM, STATIC, MANAGED, and MANAGED NOVLAN. We begin with an overview of each mode and show you how to configure each mode depending on the networking and security needs of your organization. Finally, we provide a troubleshooting section that includes recommended fixes to common configuration issues.

Appendices

Appendix A: Creating Windows images for use in Eucalyptus shows you how to create a Windows VM image (from scratch) from which you can run instances in Eucalyptus EE with Windows VM support.

Appendix B: Installing and Configuring Hypervisors provides detailed instructions on installing and configuring Xen and KVM hypervisors on your system for use with Eucalyptus; and it shows you how to test and confirm that Eucalyptus is communicating properly with your hypervisor via the *libvirt virtualization API*.

Appendix C: Admin tools presents in tabular form several Eucalyptus commands, including command-line alternates for several Users and Groups management functions.

Appendix D: An Overview of Cloud Computing presents a broad overview of the general cloud computing landscape and is useful for those who wish to gain a better understanding of how cloud computing can benefit the IT organization.

For Euca2ools (Eucalyptus client tools) instructions, see *the Eucalyptus EE User Guide.*

What's New in Eucalyptus EE 2.0?

Eucalyptus EE 2.0 adds these new extended features to the Eucalyptus core:

- Windows VM support
- VMware Support
- SAN support
- Users and Groups management
- Accounting Reports

For a comprehensive introduction to these features, see *Section 2.2: New Features in Eucalyptus EE 2.0.*

Conventions Used in this Guide

Instructional examples in this guide generally refer to a *single cluster* Eucalyptus cloud deployment, where all Eucalyptus components (except Node Controller) are installed on a single machine, generally referred to in the text as the *front-end machine* or, simply, the *front end*. The Node Controller is installed on one or more node machines referred to as either *node machine(s)* or *node(s)*. Note that in more advanced configurations, components may be installed on separate machines. In this case "front end" refers only to the machine hosting the Cloud Controller, while we may introduce additional machines that host Walrus, Storage Controller, and Cluster Controller. Command line examples in this guide generally use the following format to designate the user and the specific machine on which the administrator performs a given function:

HOST MACHINE	USER
Front-end (Cloud Controller)	[root@clc]#
Walrus	[root@walrus]#
Cluster Controller	[root@cc]#
Storage Controller	[root@sc]#
Node Controller	[root@node1]#
Client	[admin@client1]$

For example, in the above definitions, we see the user *root* on the machine called *clc.* Typically, in our examples we assume a single cluster deployment, with the Cloud Controller, Walrus, Storage Controller, and Cluster Controller on the front-end machine.

Left/right arrows **< >** are used to indicate variables. For example, within a code sample, a specific IP Address might be represented as **<IPAddress>** (where the actual input might be 192.168.7.82). Similarly, a command argument requiring the name of a node or cluster may appear as **<nodeName>** or **<clusterName>.** Arguments accepting more than one variable inputs use the variable "N" to indicate multiple possible variable inputs. For example a command argument that accepts multiple node names may appear as **<nodeName1 … nodeNameN>.**

Fonts: General instructional text is presented in Cambria 12 pt font. Command line input/output, as well as directory locations are printed in Courier 10 pt. font. For example: `/etc/eucalyptus`.

Bolded text is used within text discussions and command line samples to draw attention to the specific command or portion of command line input/output under discussion.

Italics are used throughout the text to indicate names (for example a host machine called *clc*) and to emphasize important Eucalyptus and cloud computing terms and phrases.

Contacting Eucalyptus

Please send any questions, corrections, comments, or suggestions for this Administrator's guide to documentation@eucalyptus.com.

Part I – Welcome to Eucalyptus EE 2.0

Welcome to Eucalyptus! Part I introduces the Eucalyptus EE 2.0 private cloud platform. First, we provide an overview of the *Eucalyptus Components*—the basic software modules that comprise a Eucalyptus cloud. We show you what these components are, what they do, and how they interact to form an efficiency-enhancing Eucalyptus cloud. Then we introduce the version 2.0 features that are new to Eucalyptus EE, including support for Windows VMs, VMware infrastructure, and enterprise-grade SAN devices.

Who should read Part I?

The material in Part I is directed towards system administrators and other IT professionals who wish to gain an understanding of Eucalyptus software components (and the extended features of Eucalyptus EE 2.0) as a basis for administering the Eucalyptus cloud in an enterprise data center environment.

Part I discusses these topics:

- Eucalyptus components.
- New features in Eucalyptus EE 2.0.

Section 1: Introducing Eucalyptus Components

The Eucalyptus cloud system has five high-level components—*Cloud Controller* (*CLC*), *Walrus, Cluster Controller* (*CC*), *Storage Controller* (*SC*) and *Node Controller* (*NC*), which are depicted in **Figure 1.1**. Each high-level system component has its own Web interface and is implemented as a stand-alone Web service. This has two major advantages: First, each Web service exposes a well defined language-agnostic API in the form of a WSDL document containing both the operations that the service can perform and the input/output data structures. Second, Eucalyptus leverages existing Web-service features such as security policies (WSS) for secure communication between components and relies on industry-standard web-services software packages.

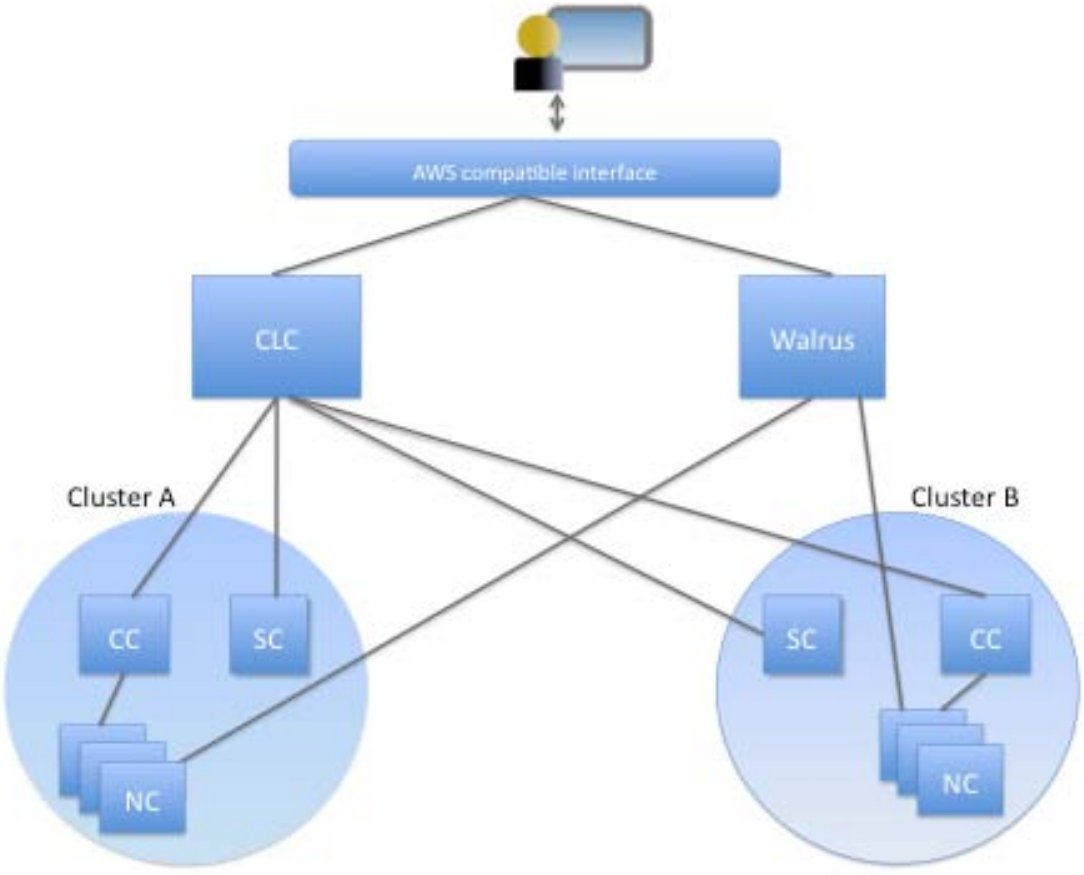


Figure 1.1 Eucalyptus Cloud components: Cloud Controller (CLC), Walrus, Cluster Controller (CC), Storage Controller (SC), and Node Controller (NC). Diagram shows interconnectivity of components in a two-cluster Eucalyptus cloud system.

Eucalyptus EE with VMware support includes an additional component: VMware Broker. A detailed description of each Eucalyptus Component follows.

1.1 Cloud Controller (CLC)

The *Cloud Controller (CLC)* is the entry-point into the cloud for administrators, developers, project managers, and end-users. The CLC is responsible for querying the node managers for information about resources, making high-level scheduling decisions, and implementing them by making requests to the Cluster Controllers (CC). The CLC, as shown in Figure 1, is also the interface to the management platform. In essence, the CLC is responsible for exposing and managing the underlying virtualized resources (servers, network, and storage) via a well-defined industry standard API (Amazon EC2) and via a Web-based user interface.

1.2 Walrus

Walrus allows users to store persistent data, organized as buckets and objects. It allows one to create, delete, list buckets, put, get, delete objects, and set access control policies. Walrus is interface compatible with Amazon's S3, providing a mechanism for storing and accessing virtual machine images and user data.

1.3 Cluster Controller (CC)

The *Cluster Controller (CC)* generally executes on a cluster front-end machine, or any machine that has network connectivity to both the machines running the Node Controller (NC) and to the machine running the CLC. CCs gather information about a set of virtual machines (VM) and schedules VM execution on specific nodes. The CC also manages the virtual instance network and participates in the enforcement of SLAs as directed by the CLC. All nodes associated with a single CC must be in the same broadcast domain (Ethernet).

1.4 Storage Controller (SC)

The *Storage Controller (SC)* provides functionality similar to the Amazon Elastic Block Storage (EBS) and is capable of interfacing with various storage systems (NFS, iSCSI, etc.). Elastic block storage is a block device that can be attached to an instance file system. An EBS cannot be shared across instances but does allow a snapshot to be created and stored in a central storage system such as Walrus, the Eucalyptus storage service. Eucalyptus EE with SAN support lets you use your enterprise-grade SAN devices to host EBS storage within a Eucalyptus cloud.

1.5 Node Controller (NC)

The *Node Controller (NC)* is executed on every machine that is designated for hosting VM instances. The NC controls VM activities, including the execution, inspection, and termination of VM instances. It also fetches and cleans up local copies of instance images (the kernel, the root file system, and the ramdisk image), and it queries and controls the system software (host OS and the hypervisor) in response to queries and control requests from the CC. The NC is also responsible for the management of the virtual network endpoint.

1.6 VMware Broker

VMware Broker (Broker) is an additional Eucalyptus component activated only in versions of Eucalyptus EE with VMware support. (Note that the Broker is not depicted in Figure 1.1). VMware Broker enables Eucalyptus to deploy VMs on VMware infrastructure elements and mediates all interactions between the Cluster Controller (CC) and VMware hypervisors (ESX/ESXi) either directly or through VMware vCenter. For additional information on VMware Broker, see *Section 2.2: Introducing VMware support*.

Section 2: New Features in Eucalyptus EE 2.0

Eucalyptus EE 2.0 extends the capabilities of the Eucalyptus cloud platform with support for Windows VMs, support for VMware infrastructure elements (e.g., ESX/ESXi, and vCenter), and support for enterprise-grade SAN devices. In addition, Eucalyptus 2.0 EE introduces an enhanced Web UI that includes new *Users and Groups* management and *Accounting Reports* interfaces The following section takes a closer look at some of these new features in Eucalyptus EE.

2.1 Introducing Windows VM Support

Eucalyptus EE with Windows VM support lets you run and manage Windows-based virtual machines inside a Eucalyptus cloud computing system.

What features does Windows VM support provide?

Eucalyptus EE with Windows VM support lets you:

- Run Windows-based virtual machines (VMs). Eucalyptus currently supports Windows 2003 Server, Windows 2008 Server, and Windows 7.
- Manage and control Windows VMs using Euca2ools.
- Create new Windows VM instances from running Window VMs using EC2 compatible `bundle-instance, describe-bundle-task,` and `cancel-bundle-task` commands.
- Access Windows VM instances inside Eucalyptus using AWS "get-password" protocol via standard RDP client tools.
- Deploy Windows VM instances on multiple hypervisors, including Xen, KVM, and VMware (ESX/ESXi).
- Create new Windows VMs from base Windows OS installation files (.iso images or CD/DVD).

2.2 Introducing VMware Support

Eucalyptus EE with VMware support enables Eucalyptus to deploy and manage virtual machines on VMware hypervisors (ESX/ESXi, either directly or through VMware vCenter), while offering an EC2/S3-compatible interface to end users. Eucalyptus EE with VMware support includes an additional component —*VMware Broker*. VMware Broker arbitrates all interaction between Eucalyptus and VMware

infrastructure components (i.e., ESX/ESXi, and vCenter). The following diagram (**Figure 2.2**) depicts the architecture of a Eucalyptus cloud with VMware Broker (and VMware infrastructure components) deployed alongside a typical open-source (KVM or XEN hypervisor-based) deployment.

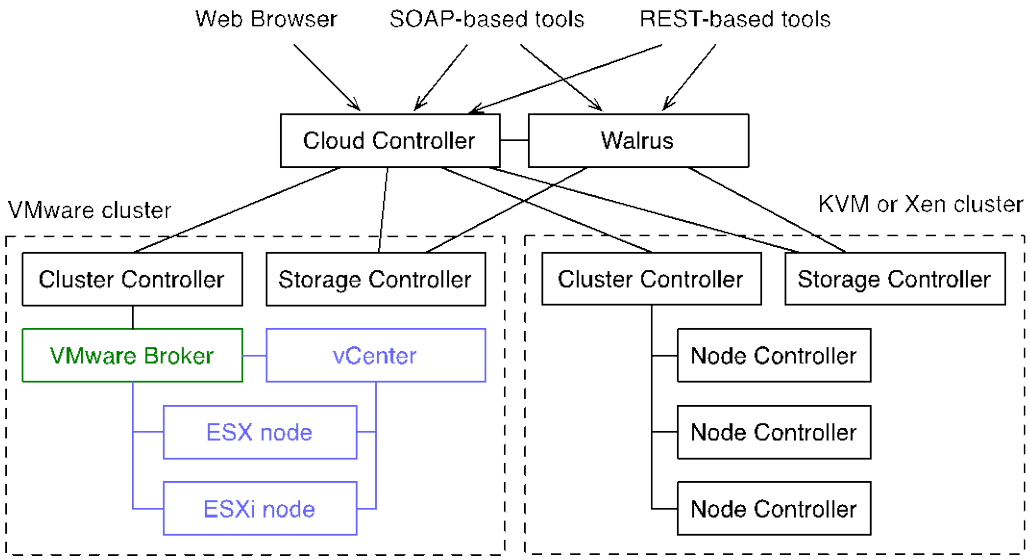


Figure 2.2 High-level architecture of Eucalyptus EE with VMware support. The VMware Broker handles all interaction between the Eucalyptus cloud and VMware infrastructure components and can run concurrently with a standard open-source (KVM or Xen hypervisor-based) cluster.

What features does VMware support provide?

Eucalyptus EE with VMware support provides:

- Support for VMware vCenter 4.0, ESX/ESXi 4.0
- Compatibility with VMware vSphere Client, which can be used alongside Eucalyptus
- Ability to incorporate both VMware (ESX/ ESXi) and open-source (XEN and KVM) hypervisors into a single cloud infrastructure.
- Extend cloud-based features (e.g., elastic IPs, security groups, S3, etc.) to a VMware infrastructure via the Eucalyptus software.

2.3 Introducing SAN Support

Eucalyptus EE with SAN support lets you integrate enterprise-grade SAN (Storage Area Network) hardware devices into a Eucalyptus cloud. SAN support extends the functionality of the Eucalyptus Storage Controller (SC) to provide a high-

performance data conduit between VMs running in Eucalyptus and attached SAN devices. Thus, Eucalyptus EE with SAN support provides a production-level EBS (block storage) solution for your Eucalyptus cloud.

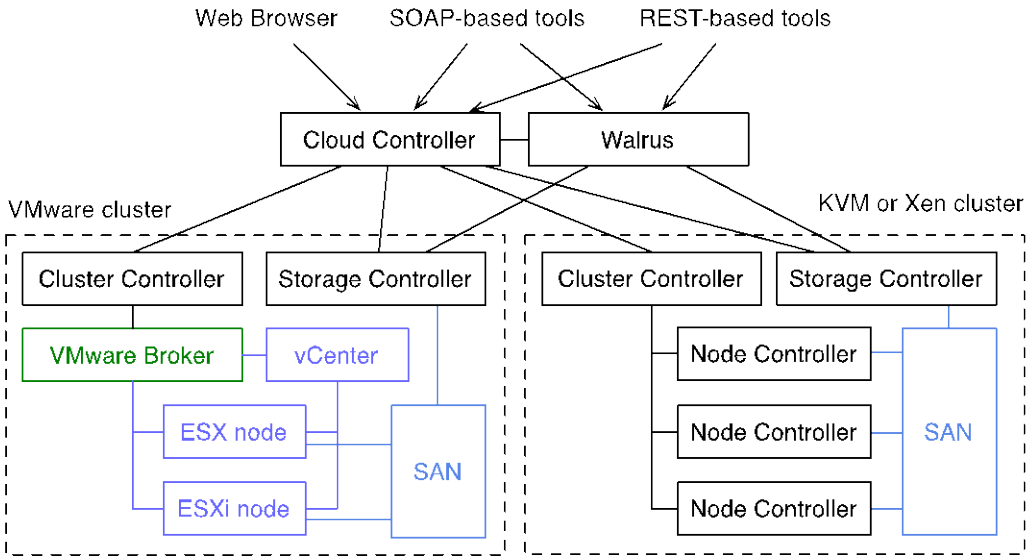


Figure 2.2 High-level architecture of Eucalyptus EE with SAN Support. Extended functionality of the Eucalyptus Storage Controller (SC) allows for integration of enterprise-grade SAN devices into a Eucalyptus cloud. VMs running in Eucalyptus link directly to attached SANs, thus avoiding I/O communication bottlenecks of the physical host.

What features does SAN support provide?

Eucalyptus EE with SAN support lets you:

- Integrate Eucalyptus block storage functionality (dynamic block volumes, snapshots, creating volumes from snapshots, etc.) with existing SAN devices.
- Link VMs in the Eucalyptus cloud directly to SAN devices, thereby removing I/O communication bottlenecks of the physical hardware host.
- Incorporate enterprise-level SAN features (high-speed, large-capacity, reliability) to deliver a production-ready EBS (block storage) solution for the enterprise.
- Attach SAN devices to Eucalyptus deployments on XEN, KVM, and VMware (ESX/ESXi 4.0) hypervisors.

2.4 Introducing Users and Groups Management Interface

Eucalyptus EE adds a new *Users and Groups* Web interface that provides administrators with a convenient graphical tool for managing users and groups interacting with the Eucalyptus cloud. The interface lets administrators create

groups based on business entities (e.g., engineering, accounting, publishing), add users to these groups, and apportion cloud resources by assigning permissions to availability zones (clusters) on a per group basis.

What features does the Users and Groups management interface provide?

- Add, modify, and delete users
- Create groups based on one or more availability zones.
- Assign and remove users to and from groups

2.5 Introducing Accounting Reports

Eucalyptus 2.0 EE introduces a new *Accounting Reports* interface, which provides administrators with tools for viewing statistics of Eucalyptus cloud activities and resource usage. These include system events (i.e., functions performed in the cloud), user and group resource usage, and Eucalyptus component deployment. Administrators can use this information as a basis for auditing and resource allocation.

What features does Accounting Reports provide?

- System Events logs
- User and Group event logs
- Component log
- Resource Usage (IP Addresses, Instances, Block Storage)
- Customizable reports on per time basis (i.e., date range)

Part II – Deploying Your Eucalyptus EE Cloud

Part II shows you how to deploy your Eucalyptus EE cloud. First, we provide the prerequisite steps you must perform to setup your environment for a Eucalyptus installation. Next, you'll learn how to install Eucalyptus from binary packages (on CentOS 5.5, openSUSE 11.2, and Ubuntu Server 10.04 as examples), and how to license and start your cloud. Then we show you how to register and perform an initial Eucalyptus configuration. Finally, we give you the necessary steps for configuring Eucalyptus EE with Windows VM support, VMware support, and SAN support.

Deployment Overview:

Step1: Setup the environment (installation prerequisites).
Step 2: Install Eucalyptus.
Step 3: Configure the cloud.

Who should read Part II?

The material in Part II is designed to help system administrators and other IT professionals install and configure a Eucalyptus cloud in a professional data center environment.

Part II discusses these topics:

- Baseline prerequisites
- Additional prerequisites for Windows VM, VMware, and SAN support
- Downloading packages
- Installing Eucalyptus EE dependencies, components, and license
- Starting Eucalyptus
- Registering Eucalyptus
- Adding and removing nodes
- Configuring Eucalyptus EE for Windows VM support, VMware support, and SAN support

Section 3: Installing Eucalyptus EE 2.0

A Eucalyptus cloud installation consists of five basic components, plus an additional component with installations of Eucalyptus EE with VMware support. The ***Cloud Controller*** (CLC) and ***Walrus*** are top-level components, with one of each in a cloud installation. The Cloud Controller 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 implements bucket-based storage, which is available outside and inside a cloud through S3-compatible SOAP and REST interfaces.

Top-level components (CLC and Walrus) can aggregate resources from multiple clusters (i.e., collections of nodes sharing a LAN segment, possibly residing behind a firewall). Each cluster has 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 has a ***Node Controller*** (NC) for controlling the hypervisor. Communication among these components takes place over SOAP with WS-security.

Eucalyptus EE with VMware support includes an additional component—***VMware Broker***—that enables Eucalyptus to deploy VMs on VMware hypervisors (ESX/ESXi) directly or through VMware vCenter.

About These Instructions

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

3.1 Setting up the Eucalyptus EE Environment (Prerequisites)

This section shows you how to set up your computing environment for a Eucalyptus EE installation on Centos 5.5. Baseline prerequisites apply to all Eucalyptus EE systems. Additional prerequisites are provided as applicable for optional Eucalyptus EE features, including Windows support, VMware support, and SAN support.

3.1.1 Baseline Prerequisites for Eucalyptus EE

For a standard installation of CentOS 5.5, the following steps will satisfy all prerequisites;

1. The front end, node(s), and client machine system clocks are synchronized (e.g., using NTP):

```
[root@clc]# yum install -y ntp
[root@clc]# ntpdate pool.ntp.org
```

2. Each node has a fully installed and configured Xen hypervisor that allows controlling the hypervisor via HTTP from *localhost*. You will also need a kernel with Xen support enabled. For more information, see *Appendix B: Installing and Configuring Hypervisors.*

3. 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, and 8774 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 using Eucalyptus facilities for enabling custom firewall rules on both the front end and nodes. For example, on CentOS 5.5:

   From a text console:

   - run **`system-config-securitylevel-tui`**
   - select `Security Level: Disabled`
   - select `OK`

   From an X terminal:

   - run **`system-config-securitylevel`**
   - select 'Disabled' for 'Firewall'
   - select the 'SELinux' tab
   - select either 'Permissive' or 'Disabled' for SELinux Setting

4. (Optional) The Eucalyptus CLC uses the systems existing SMTP services to deliver account information and system-level alerts. If you want these messages to be delivered as part of the Enterprise's Email infrastructure, you must configure SMTP services (e.g., *postfix, sendmail, dovecot, etc*.). Please contact your local Email Administrator for the proper configuration of the SMTP service that complies with local policies.

3.1.2 Additional Prerequisites for Windows VM Support

Eucalyptus EE with Windows VM support has these additional prerequisites:

1. A licensed installation copy (.iso image or CD/DVD disk) of a compatible Windows OS. Eucalyptus EE currently supports Windows virtual machines created from Windows Server 2003, Windows Server 2008, and Windows 7. For additional Windows-related licensing information, see the following links:

   Windows Volume Activation Overview Guide
   http://technet.microsoft.com/en-us/library/dd979803.aspx

   Windows Volume Activation Planning Guide
   http://technet.microsoft.com/en-us/library/dd878528.aspx

   Windows Volume Activation Deployment Guide
   http://technet.microsoft.com/en-us/library/dd772269.aspx

2. A VNC client such as RealVNC or Virtual Manager/Virtual Viewer (Centos/Xen) must be available for initial installation. (Subsequent Eucalyptus-hosted Windows instances will use RDP, but the initial installation requires VNC.)

   For information on preparing a Windows virtual machine image from your base Windows OS, see *Appendix A.*

3.1.3 Additional Prerequisites for VMware Support

Eucalyptus EE with VMware support has the following additional prerequisites:

1. Installed and configured VMware infrastructure software (ESX 4.0 and/or ESXi 4.0 hypervisors with or without vCenter Server 4.0)

2. The front-end machine must be able to reach the VMware hosts on ports 443, 902, and 903.

3. The VMware 'Administrator' account credentials must be provided to Eucalyptus when configuring VMware support (see section 'Configuring VMware Support' below), or an equivalent account with sufficient permissions must be created on VMware vCenter or ESX hosts:

   Option A: Administrator access to entire vSphere infrastructure:

The most straightforward way to configure vSphere for Eucalyptus—and the least likely to incur permissions-related complications—is to give Eucalyptus unrestricted (administrator-level) access to your vSphere endpoint(s). This can be accomplished either by using an existing administrative account and password or by creating a new account for Eucalyptus and associating it with vSphere's standard 'Administrator' role at the top level of the vSphere hierarchy as seen in vSphere client.

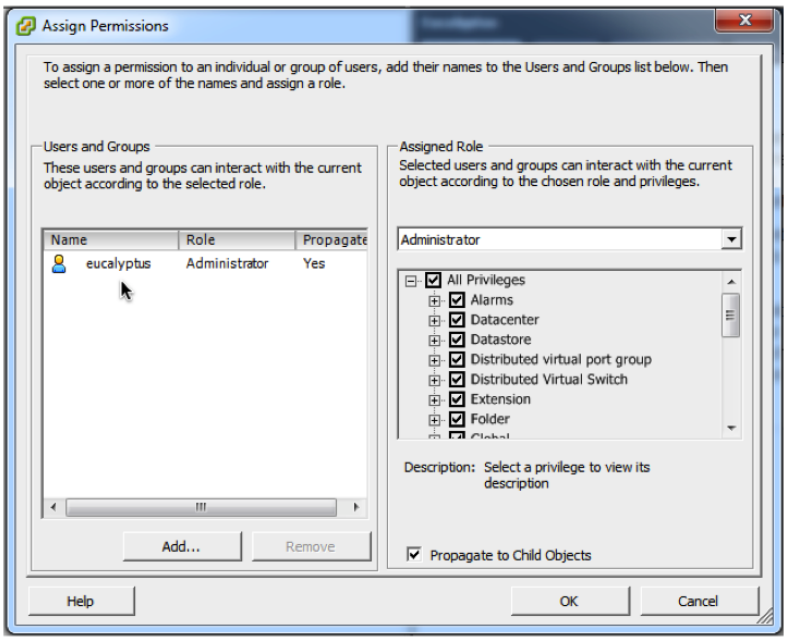


Figure 3.1 The vSphere client Assign Permissions window. Here you can associate user 'eucalyptus' with 'Administrator' role at the top of the vSphere hierarchy.

Option B: 'Least privilege' access to vSphere infrastructure:

To give the minimum required amount of control to Eucalyptus over your vSphere infrastructure, you will need to create one new user and two new roles. The new user, named, e.g., 'eucalyptus', and its password will be used when configuring Eucalyptus for VMware support (as described in *Section 4.4 Configuring VMware Support*). The new roles should be defined and associated with vSphere objects as follows:

(1) Top-level role, named, e.g., 'Eucalyptus vSphere', must be associated with the 'eucalyptus' user at the top level of the vSphere hierarchy only ("Propagate to Child Objects" does not need to be checked). This role should have only one privilege:

(2) Resource-level role, named, e.g., 'Eucalyptus', can be associated with the 'eucalyptus' user at the level(s) encapsulating the resources that you wish Eucalyptus to use. For example, you can create a new virtual datacenter for Eucalyptus to use, add to it the relevant hosts or clusters, and assign the 'eucalyptus' user 'Eucalyptus' role for the new datacenter (making sure to check "Propagate to Child Objects"). The 'Eucalyptus' role should have the following privileges:

Datastore
- Allocate Space
- Browse datastore
- Low level file operations

Folder
- Create folder

Host
- Configuration
  - Network configuration
  - Storage partition configuration

Network
- Assign network

Resource
- Assign virtual machine to resource pool

Virtual machine
- *all permissions*

4. Each node requires at least one *datastore* (either local or one shared by multiple nodes). If more than one datastore is available to a node, Eucalyptus will choose the datastore arbitrarily. Hence you may need to specify a datastore in Eucalyptus's configuration file for VMware (`vmware_conf.xml`).

To check datastores available on a host, perform the following steps with vSphere client pointed either at vCenter or at a particular ESX/ESXi node:

1. Select a host in left-hand-side panel.
2. Select the "Configuration" tab.
3. Select "Storage" in the secondary left-hand-side panel.
4. Select "View: Datastores" at the top of the panel.

Similarly, each node must have a network reachable by the node running Eucalyptus. If more than one network is available, you may need to specify a network in `vmware_conf.xml`. For information on `vmware_conf.xml` (including proper XML syntax), see *Section 4.4.1: Testing VMware Broker Configuration.*

To check the network settings and create a network (if necessary) perform the following steps with vSphere client pointed either at vCenter or at a particular ESX/ESXi node:

1. Select a host in left-hand-side panel.

2. Select the "Configuration" tab.

3. Select "Networking" in the secondary left-hand-side panel.

4. If there is no "VM Network" in the list, add it as follows:

    a. Click "Add Networking..." in the upper-right corner.
    b. Select "Virtual Machine" and click "Next."
    c. Select a switch (e.g., "Use vSwitch0") and click "Next."
    d. Enter "VM Network" for Network Label, leave VLAN ID blank, and click "Next."
    e. Check the summary and click "Finish".

5. To enable EBS support in Eucalyptus on VMware, each of the ESX/ESXi nodes in your infrastructure must be configured to support iSCSI. Given a node that is licensed for iSCSI support, this amounts to enabling and configuring the gateway for the VMkernel network. To accomplish that, perform the following steps with vSphere client pointed either at vCenter or at a particular ESX/ESXi node:

1. Select a host in left-hand-side panel.

2. Select the "Configuration" tab.

3. Select "Networking" in the secondary left-hand-side panel.

4. If there is no "VMkernel" network listed, add it as follows:

    a. Click "Add Networking..." in the upper-right corner.
    b. Select "VMkernel" and click "Next."
    c. Select a switch (e.g., "Use vSwitch0") and click "Next."
    d. Select the label, VLAN ID (if any) and click "Next."
    e. Select dynamic or static network configuration and click "Next",
    f. Click "Finish".

5. Select "DNS and Routing" in the secondary left-hand-side panel.

6. If "VMkernel" does not have a gateway, add it as follows:

    a. Select "Properties..." in upper-right corner,

b. Select the "Routing" tab, enter gateway's IP, and click “OK”.

For additional information on configuring vSphere, see the VMware website at http://www.vmware.com/support/pubs/vs_pubs.html.

6. Ensure that VMware Tools are installed in the images that you will instantiate through Eucalyptus. This allows Eucalyptus to discover an instance’s IP address in SYSTEM networking mode. This is also required for using the euca-bundle-instance command when running Windows VMs in Eucalyptus. For information on euca-bundle-instance, see the Eucalyptus EE User Guide. For information on installing VMware Tools, consult VMware documentation at www.vmware.com.

3.1.4 Additional Prerequisites for SAN Support

Eucalyptus EE with SAN support has these additional prerequisites:

- Configured SAN device(s). Eucalyptus currently supports these SAN devices:
  - Dell EqualLogic (PS4000 series, PS6000 series). For more information on Dell EqualLogic SANs, see *www.dell.com*.
  - NetApp (FAS2000 series, FAS6000 series) supported. For more information on NetApp SANs, see *www.netapp.com.*

For the latest information on Eucalyptus supported SAN devices, see the Eucalyptus website at *www.eucalyptus.com.*

3.2 Eucalyptus EE Installation Steps

This section outlines the steps required for installing Eucalyptus EE from *binary packages*:

Step 1: Download installation packages.
Step 2: Install OS dependencies.
Step 3: Install Eucalyptus dependencies
Step 4: Install Eucalyptus EE components: CLC, Walrus, CC, SC, NC, and optionally VMware broker (for VMware support)
Step 5: Install Eucalyptus EE license.
Step 6: Start Eucalyptus EE.

(Note that before you can run VMs in the Eucalyptus cloud, you must first register the Eucalyptus components (to connect them), as shown in Section *4: Configuring Eucalyptus*.)

The following sections 3.3 and 3.4 provide detailed instructions for installing Eucalyptus EE on CentOS 5.5 and openSUSE 11.2, respectively.

3.3 Installing Eucalyptus EE 2.0 on CentOS 5.5

This section shows you how to install Eucalyptus EE from binary packages on CentOS 5.5. In what follows, make sure the value of **`$VERSION`** is set to the version of Eucalyptus you wish to install. For example, we can set the value 2.0.1 using bash:

```
# export VERSION=2.0.1
```

3.3.1 Download Installation Packages

Eucalyptus provides installation packages bundled into two tarballs: one containing third-party dependencies that Eucalyptus relies on (e.g., Rampart, Axis2C, Java libraries, etc.), the other containing Eucalyptus components (CLC, Walrus, SC, CC, NC, and VMware Broker).

First, download the appropriate tarballs for your system architecture (32-bit or 64-bit) from the location provided to you by Eucalyptus.

Eucalyptus dependency packages:

32-bit machines: **eucalyptus-deps-centos-i386-${VERSION}eee.tar.gz**
64-bit machines: **eucalyptus-deps-centos-x86_64-${VERSION}eee.tar.gz**

Eucalyptus component packages:

32-bit machines: **eucalyptus-centos-i386-${VERSION}eee.tar.gz**
64-bit machines: **eucalyptus-centos-x86_64-${VERSION}eee.tar.gz**

Next, untar the bundles in a temporary location as shown:

```
[root@clc]# tar zxvf eucalyptus-deps-centos-*.tar.gz
[root@clc]# tar zxvf eucalyptus-centos-*.tar.gz
```

3.3.2 Installing OS Dependencies

Before installing Eucalyptus, you must first prepare your system by installing CentOS 5.5 operating system dependencies, as shown:

```
[root@clc]# 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 \
      mysql openldap-clients perl-Convert-ASN1 chkfontpath \
      scsi-target-utils fuse-libs libgcc.i386 bridge-utils
```

3.3.3 Installing Eucalyptus Dependencies

Install JDK

First, install the Sun JDK on the front-end machine (CLC, Walrus, SC and VMware Broker rely on this dependency) as shown:

Go to **http://java.sun.com/javase/downloads/index.jsp** and select Java Platform JDK.

In the drop-down menu under "Platform:" select either Linux (for i386) or Linux x64 (for x86_64). Once you have download the jdk*.bin file, install the package as follows:

```
[root@clc]# chmod 0700 jdk*.bin
[root@clc]# ./jdk*.bin
```

The `./jdk*.bin` command will install all the files in a folder labeled `jdk1.6.0_20` for example. Now create the directory`/opt/packages/jdk`, and move the contents of the folder `jdk1.6.0_20` to `/opt/packages/jdk`, as shown:

```
[root@clc]# mkdir -p /opt/packages/jdk
[root@clc]# mv jdk1.6.0_20/* /opt/packages/jdk/
```

For more information about installing Java Development Kit, please visit the following link: **http://java.sun.com/javase/6/webnotes/install/index.html#linux.**

Install Standard Dependencies

Second, install the Eucalyptus-prepared dependency RPMs on *all* machines (i.e., install all RPMs on *both* front-end and node machines) (Note that you must remove any preexisting copies of these dependencies prior to installation. Multiple copies of a dependency will return an error message.)

```
[root@clc]# cd eucalyptus-deps*
[root@clc]# yum install -y *.rpm --nogpgcheck

[root@node1]# cd eucalyptus-deps*
[root@node1]# yum install -y *.rpm --nogpgcheck
```

Install VDDK

Third, install the VMware's VIX DiskLib dynamic libraries (VDDK) on the front end. These libraries are contained within the vmware-vix-disklib-distrib folder within the eucalyptus dependences tarball.

Run the installer script, accepting the End User License Agreement and selecting the default install prefix `[/opt/packages/vddk]` as the root directory where all folders will be placed, as shown:

```
[root@clc]# cd eucalyptus-deps*/vmware-vix-disklib-distrib

[root@clc]# ./vmware-install.pl
Creating a new VMware VIX DiskLib API installer database using the tar4 format.
Installing VMware VIX DiskLib API.
You must read and accept the VMware VIX DiskLib API End User License Agreement
to continue. Press enter to display it.

Do you accept? (yes/no) yes

Thank you.
What prefix do you want to use to install VMware VIX DiskLib API?
The prefix is the root directory where the other
folders such as man, bin, doc, lib, etc. will be placed.

/opt/packages/vddk

The installation of VMware VIX DiskLib API 1.1.0 build-163495 for Linux
completed successfully. You can decide to remove this software from your system
at any time by invoking the following command:
"/usr/bin/vmware-uninstall-vix-disklib.pl".
Enjoy,

--the VMware team
```

3.3.4 Installing Eucalyptus Component Packages

On the front end, install the Eucalyptus component RPMs **(-cloud, -walrus, -sc, -cc)** (Note that if you are running a licensed version of Eucalyptus with VMware support, you will also install **–broker**) as shown. In the following examples, we use `x86_64`, which should be replaced with `i386` on 32-bit architectures.

```
[root@clc]# rpm -Uvh eucalyptus-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-common-java-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-cloud-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-walrus-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-sc-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-cc-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-gl-${VERSION}eee-0.*.x86_64.rpm

      # include the following RPM for Vmware support only
      eucalyptus-broker-${VERSION}eee-0.*.x86_64.rpm
```

On the node machine(s), install the NC RPMs **(-nc)**, as shown:

```
[root@node1]# rpm -Uvh eucalyptus-$VERSION-0.*.x86_64.rpm \
      eucalyptus-nc-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-gl-${VERSION}eee-0.*.x86_64.rpm
```

3.3.5 Installing the Eucalyptus EE License

Along with the Eucalyptus EE software you have received a license file. Please install this file in the following directory:

```
/etc/eucalyptus/
```

3.3.6 Starting Eucalyptus

Startup your Eucalyptus services. On the front end, enter:

```
[root@clc]# /etc/init.d/eucalyptus-cloud start
[root@clc]# /etc/init.d/eucalyptus-cc start
```

(Note that before starting the node controller, you must ensure that your hypervisor and libvirt are configured correctly to run Eucalyptus. For more information, see *Appendix B: Installing and Configuring Hypervisors*.)

On the node machine, start the node controller:

```
[root@node1]# /etc/init.d/eucalyptus-nc start
```

You have now installed Eucalyptus EE software on CentOS 5.5 and are ready to proceed with registering and configuring your Eucalyptus cloud. Note that before you can run VMs in Eucalyptus, you must first register the Eucalyptus components (to connect them), as described in *Section 4: Configuring Eucalyptus*.

3.4 Installing Eucalyptus EE 2.0 on openSUSE 11.2

To install Eucalyptus EE on openSUSE 11.2, follow the steps outlined in the above *Section 3.2: Eucalyptus EE installation steps*. Be sure to download and install the corresponding *openSUSE 11.2* packages. In addition, make sure the value of **`$VERSION`** is set to the version of Eucalyptus you wish to install. For example, we can set the value 2.0.1 using bash:

```
# export VERSION=2.0.1
```

3.4.1 Download Installation Packages

First, download the appropriate tarballs for your operating system at **http://open.eucalyptus.com/downloads.**

Eucalyptus dependency packages:

32-bit machines: **eucalyptus-deps-opensuse-i586-${VERSION}eee.tar.gz**
64-bit machines: **eucalyptus-deps-opensuse-x86_64-${VERSION}eee.tar.gz**

Eucalyptus component packages:

32-bit machines: **eucalyptus-opensuse-i586-${VERSION}eee.tar.gz**
64-bit machines: **eucalyptus-opensuse-x86_64-${VERSION}eee.tar.gz**

Next, untar the bundles in a temporary location as shown:

```
[root@clc]# tar zxvf eucalyptus-opensuse-*.tar.gz
[root@clc]# tar zxvf eucalyptus-deps-opensuse-*.tar.gz
```

3.4.2 Installing OS Dependencies

Install the following openSUSE 11.2 dependencies, as shown:

```
[root@clc]# zypper -n in curl bzr 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 openssl mysql openldap2-client libvirt \
      apache2 perl-Crypt-OpenSSL-Random \
      perl-Crypt-OpenSSL-RSA libfuse2 tgt libgcc44-32bit
```

3.4.3 Install Eucalyptus Dependencies

Install JDK

First, install the Sun JDK on the front-end machine (CLC, Walrus, SC and VMware Broker rely on this dependency) as shown:

Go to **http://java.sun.com/javase/downloads/index.jsp** and select Java Platform JDK.

In the drop-down menu under "Platform:" select either Linux (for i386) or Linux x64 (for x86_64). Once you have downloaded the jdk*.bin file, install the package:

```
[root@clc]# chmod 0700 jdk*.bin
[root@clc]# ./jdk*.bin
```

The `./jdk*.bin` command installs all the files in a folder labeled `jdk1.6.0_20` for example. Now create the directory`/opt/packages/jdk`, and move the contents of the folder `jdk1.6.0_20` to `/opt/packages/jdk`, as shown:

```
[root@clc]# mkdir -p /opt/packages/jdk
[root@clc]# mv jdk1.6.0_20/* /opt/packages/jdk/
```

For more information about installing Java Development Kit, please visit the following link: **http://java.sun.com/javase/6/webnotes/install/index.html#linux.**

Install Standard Dependencies

Second, install the Eucalyptus-prepared dependency RPMs on *all* machines (i.e., install all RPMs on *both* front-end and node machines) (Note that you must remove any preexisting copies of these dependencies prior to installation. Multiple copies of a dependency will return an error message.)

```
[root@clc]# cd eucalyptus-deps*
[root@clc]# rpm -Uvh *.rpm

[root@node1]# cd eucalyptus-deps*
[root@node1]# rpm -Uvh *.rpm
```

Install VDDK

Third, install the VMware's VIX DiskLib (VDDK) on the front end. These libraries are contained within the vmware-vix-disklib-distrib folder within the eucalyptus dependences tarball.

Run the installer script, accepting the End User License Agreement and specifying /opt/packages/vddk/ instead of the default install prefix, as shown:

```
[root@clc]# cd eucalyptus-deps*/vmware-vix-disklib-distrib

[root@clc]# ./vmware-install.pl
Creating a new VMware VIX DiskLib API installer database using the tar4 format.
Installing VMware VIX DiskLib API.
You must read and accept the VMware VIX DiskLib API End User License Agreement to
continue. Press enter to display it.

Do you accept? (yes/no) yes

Thank you.
What prefix do you want to use to install VMware VIX DiskLib API?
The prefix is the root directory where the other
folders such as man, bin, doc, lib, etc. will be placed.

/opt/packages/vddk

The installation of VMware VIX DiskLib API 1.1.0 build-163495 for Linux completed
successfully. You can decide to remove this software from your system at any time
by invoking the following command:
"/usr/bin/vmware-uninstall-vix-disklib.pl".
Enjoy,

--the VMware teacdm
```

3.4.4 Installing Eucalyptus Component Packages

On the front end, install the Eucalyptus component RPMs **(-cloud, -walrus, -sc,** and **-cc)** (Note that if you are running a licensed version of Eucalyptus with VMware support, you will also install **–broker**) as shown. In the following examples, we use `x86_64`, which should be replaced with `i386` on 32-bit architectures.

```
[root@clc]# rpm -Uvh eucalyptus-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-common-java-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-cloud-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-walrus-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-sc-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-cc-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-gl-${VERSION}eee-0.*.x86_64.rpm

      # include the following RPM for Vmware support only
      eucalyptus-broker-${VERSION}eee-0.*.x86_64.rpm
```

On the node machine(s), install the NC RPM **(-nc**) and common packages, as shown:

```
[root@node1]# rpm -Uvh eucalyptus-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-nc-${VERSION}eee-0.*.x86_64.rpm \
      eucalyptus-gl-${VERSION}eee-0.*.x86_64.rpm
```

3.4.5 Installing the Eucalyptus EE License

Along with the Eucalyptus EE software you have received a license file. Please install this file in the following directory: **`/etc/eucalyptus/`**

3.4.6 Starting Eucalyptus

Startup your Eucalyptus services. On the front end, enter:

```
[root@clc]# /etc/init.d/eucalyptus-cloud start
[root@clc]# /etc/init.d/eucalyptus-cc start
```

(Note that before starting the node controller, you must ensure that your hypervisor and libvirt are configured correctly to run Eucalyptus. For more information, see *Appendix B: Installing and Configuring Hypervisors*.)

On the node controller, enter:

```
[root@node1]# /etc/init.d/eucalyptus-nc start
```

You have now installed the Eucalyptus software on openSUSE 11.2 and are ready to proceed with registering and configuring your Eucalyptus cloud. Note that before you can run VMs in Eucalyptus, you must first register the Eucalyptus components (to connect them), as described in *Section 4: Configuring Eucalyptus*.

3.5 Upgrading from Eucalyptus version 1.6.2 to Eucalyptus 2.0.* EE

Eucalyptus provides convenient RPM packages for upgrading from Eucalyptus version 1.6.2 to Eucalyptus 2.0 EE. Note that your original Eucalyptus 1.6.2 installation must have been performed using RPM packages. The upgrade RPMs are not compatible with previous versions of Eucalyptus installed from source.

Before you begin the upgrade process, stop all services on all components of Eucalyptus. Begin the stopping process on the nodes. Then stop the services on the front end, as shown:

```
[root@node1]# /etc/init.d/eucalyptus-nc stop

[root@clc]# /etc/init.d/eucalyptus-cc stop
[root@clc]# /etc/init.d/eucalyptus-cloud stop
```

Next, make sure you have downloaded and untarred the appropriate packages for CentOS 5.5 or openSUSE 11.2 as shown in sections 3.3.1 or 3.4.1, respectively; and make sure you have installed all necessary CentOS 5.5 or openSUSE 11.2 operating system dependencies, as shown in sections 3.3.2, then:

1. Install Eucalyptus dependency RPMs on *all* machines (i.e., install all RPMs on front-end machine and install all RPMs on node machine(s)). Note that you must remove any preexisting copies of these dependencies prior to installation. Multiple copies of a dependency will return an error message.

   On CentOS 5.5:

   ```
   [root@clc]# cd eucalyptus-deps*
   [root@clc]# rpm -Uvh *

   [root@node1]# cd eucalyptus-deps*
   [root@node1]# rpm -Uvh *
   ```

   On openSUSE 11.2:

   ```
   [root@clc]# cd eucalyptus-deps*
   [root@clc]# rpm -Uvh *

   [root@node1]# cd eucalyptus-deps*
   [root@node1]# rpm -Uvh *
   ```

2. Install Eucalyptus components. Note that you must install the upgrade components on the same machine as the existing component. In the "single cluster" example, this means all components (except NC) are installed on the front-end machine, while NC is installed on the node machine(s). For both CentOS 5.5 and openSUSE 11.2:

   On the front-end machine:

```
[root@clc]# rpm -Uvh eucalyptus-2.0.0eee-*.x86_64.rpm \
      eucalyptus-common-java-2.0.0eee-*.x86_64.rpm \
      eucalyptus-cloud-2.0.0eee-*.x86_64.rpm \
      eucalyptus-walrus-2.0.0eee-*.x86_64.rpm \
      eucalyptus-sc-2.0.0eee-*.x86_64.rpm \
      eucalyptus-cc-2.0.0eee-*.x86_64.rpm \
      eucalyptus-gl-2.0.0eee-*.x86_64.rpm

      # include the following RPM for Vmware support only
      eucalyptus-broker-2.0.0eee-*.x86_64.rpm
```

   On the node machine(s):

```
[root@node1]# rpm -Uvh eucalyptus-2.0.0eee-0.1436.x86_64.rpm \
      eucalyptus-nc-2.0.0eee-*.x86_64.rpm \
      eucalyptus-gl-2.0.0eee-*.x86_64.rpm
```

3. Perform the upgrade process as follows:

   First, note the name of all backup directories (**`$BACKUPDIR`**) printed out when performing the RPM upgrade. Then enter the following on all machines:

```
/usr/share/eucalyptus/euca_upgrade --old $BACKUPDIR --new / \
      --conf --keys >/dev/null 2>&1
```

   On the front-end machine (machine running the CLC), enter:

```
[root@clc]# export JAVA_HOME=/opt/packages/jdk
[root@clc]# export PATH=$JAVA_HOME/bin:$PATH

[root@clc]# /usr/share/eucalyptus/euca_upgrade –old $BACKUPDIR \
      --new / --db --product > /dev/null 2>&1
```

4. Install the Eucalyptus EE license file you received along with the Eucalyptus EE software. Please install this file in the following directory: **`/etc/eucalyptus/`**

5. Start Eucalyptus services, as shown:

```
[root@clc]# /etc/init.d/eucalyptus-cloud start
[root@clc]# /etc/init.d/eucalyptus-cc start
[root@node1]# /etc/init.d/eucalyptus-nc start
```

Note that it is *not* necessary to reregister your cloud after starting your upgraded components.

3.5.1 Upgrading Volumes and Snapshots for EE 2.0 with SAN Support

If you have upgraded from 1.6.2 to Eucalyptus EE with SAN support, after performing the SAN setup (see *Section 4.5: Configuring SAN support*), you must also upgrade existing volumes and snapshots from 1.6.2 to EE, as shown:

First, create a volume to ensure your SAN setup is working correctly:

```
[root@clc]# euca-create-volume -z <clusterName> -s 1
```

To confirm that the volume has been created, run `euca-describe-volumes`.

Next, on the front end, for each registered storage controller, run the following command, where `sc_ip` is the IP address of the storage controller. Note that you can obtain a list of registered storage controllers (including IP address/hostnames) at `euca_conf –list-scs`.

```
[root@clc]# euca-convert-volumes -H sc_ip -B OverlayManager
```

For progress and status, observe the cloud output log at **`/var/log/eucalyptus/cloud-output.log`**

When upgrade is complete, you will see "Conversion Complete."

Note that the storage controller will not be available for servicing user requests during the upgrade process. This is an administrator maintenance operation that must be performed *before* your cloud is made available to users.

3.6 Upgrading from Eucalyptus 2.0 EE to Eucalyptus 2.0.1 EE

Eucalyptus provides convenient RPM packages for upgrading from Eucalyptus version 2.0 to Eucalyptus 2.0.1 EE. Before you begin the upgrade process, stop all services on all components of Eucalyptus. Begin the stopping process on the nodes. Then stop the services on the front end, as shown:

```
[root@node1]# /etc/init.d/eucalyptus-nc stop

[root@clc]# /etc/init.d/eucalyptus-cc stop
[root@clc]# /etc/init.d/eucalyptus-cloud stop
```

Next, make sure you have downloaded and untarred the appropriate packages for CentOS 5.5 or openSUSE 11.2 as shown in sections 3.3.1 or 3.4.1, respectively, then

1. Install Eucalyptus components. Note that you must install the upgrade components on the same machine as the existing component. In the "single cluster" example, this means all components (except NC) are installed on the front-end machine, while NC is installed on the node machine(s). For both CentOS 5.5 and openSUSE 11.2:

   On the front-end machine:

   ```
   [root@clc]# rpm -Uvh eucalyptus-2.0.1eee-*.x86_64.rpm \
         eucalyptus-common-java-2.0.1eee-*.x86_64.rpm \
         eucalyptus-cloud-2.0.1eee-*.x86_64.rpm \
         eucalyptus-walrus-2.0.1eee-*.x86_64.rpm \
         eucalyptus-sc-2.0.1eee-*.x86_64.rpm \
         eucalyptus-cc-2.0.1eee-*.x86_64.rpm \
         eucalyptus-gl-2.0.1eee-*.x86_64.rpm

         # include the following RPM for Vmware support only
         eucalyptus-broker-2.0.1eee-*.x86_64.rpm
   ```

   On the node machine(s):

   ```
   [root@node1]# rpm -Uvh eucalyptus-2.0.1eee-*.x86_64.rpm \
         eucalyptus-nc-2.0.1eee-*.x86_64.rpm \
         eucalyptus-gl-2.0.1eee-*.x86_64.rpm
   ```

2. Perform the upgrade process as follows:

First, note the name of all backup directories (**$BACKUPDIR**) printed out when performing the RPM upgrade. Then enter the following on all machines:

```
/usr/share/eucalyptus/euca_upgrade --old $BACKUPDIR --new / \
      --conf >/dev/null 2>&1
```

3. Install the new Eucalyptus EE license file you received along with the Eucalyptus EE 2.0.1 software. Please install this file in the following directory: **/etc/eucalyptus/**

4. Start Eucalyptus services, as shown:

```
[root@clc]# /etc/init.d/eucalyptus-cloud start
[root@clc]# /etc/init.d/eucalyptus-cc start
[root@node1]# /etc/init.d/eucalyptus-nc start
```

Note that it is *not* necessary to reregister your cloud after starting your upgraded components.

Section 4: Configuring Eucalyptus EE

This section shows you how to register and configure Eucalyptus EE once you have completed the software installation process as described in the previous section. We begin with instructions for registering Eucalyptus components and performing initial configuration. This is followed by more detailed instructions for configuring specific Eucalyptus EE features, including VMware support and SAN support. The first step is to register the Eucalyptus components so they can communicate with each other.

4.1 Registering Eucalyptus EE components

This section assumes that you have installed all Eucalyptus EE components and they are up and running. We also assume that your Eucalyptus setup consists of a "single cluster" (one front-end machine and one or more nodes).

First, you must register various front-end components. To do this, run the following commands on the front end:

```
[root@clc]# /usr/sbin/euca_conf \
      --register-walrus <front-end IPAddress>
[root@clc]# /usr/sbin/euca_conf --register-cluster \
     <clusterName> <front-end IPAddress>
[root@clc]# /usr/sbin/euca_conf --register-sc \
     <clusterName> <front-end IPAddress>
```

Note: To use Eucalyptus EE with VMware support you must also register the VMware Broker component, as shown:

```
[root@clc]# /usr/sbin/euca_conf \
      --register-vmwarebroker <clusterName> \
      <front-end IPAddress>
```

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

```
[root@clc]# /usr/sbin/euca_conf --register-nodes \
      "<Node0-IPAddress> ... <NodeN-IPAddress>"
```

At this point, you have successfully registered Eucalyptus EE components and are ready to proceed with an initial configuration of Eucalyptus EE.

4.1.1 Adding and Removing Nodes

Once your Eucalyptus system is registered and thus connected and running, you can add additional nodes or delete nodes. To add a node, enter:

```
[root@clc]# /usr/sbin/euca_conf --register-nodes \
      "<nodeName1> ... <nodeNameN>"
```

You will be prompted for a password to log into <nodeNameN>. This is needed to propagate the cryptographic keys.)

Similarly, to remove a node, enter:

```
[root@clc]# /usr/sbin/euca_conf \
      --deregister-nodes "<nodeName1> ... <nodeNameN>"
```

4.2 Initial Configuration of Eucalyptus EE

Open your Web browser to the Eucalyptus interface at

https://<front-endMachine>:8443

Since Eucalyptus uses a self-signed certificate, your browser is likely to prompt you to accept the certificate. It may take a few minutes after starting the Cloud Controller for the URL to respond 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.
3. Confirm the IP of the Cloud Controller host.

After clicking 'Submit', the *Configuration* tab appears (**Figure 4.1**). If necessary, you can adjust cloud configuration parameters by inputting new values in the fields provided. Configurable parameters include settings for *Cloud Host*, *DNS*, *Walrus*, *Clusters*, and *VMTypes*. Note that to change IP addresses of either Walrus host or Cluster host, you must first deregister Walrus and Cluster; then reregister each with a new IP address using the `euca_conf` command (see *Section 4.1: Registering Eucalyptus EE Components*).

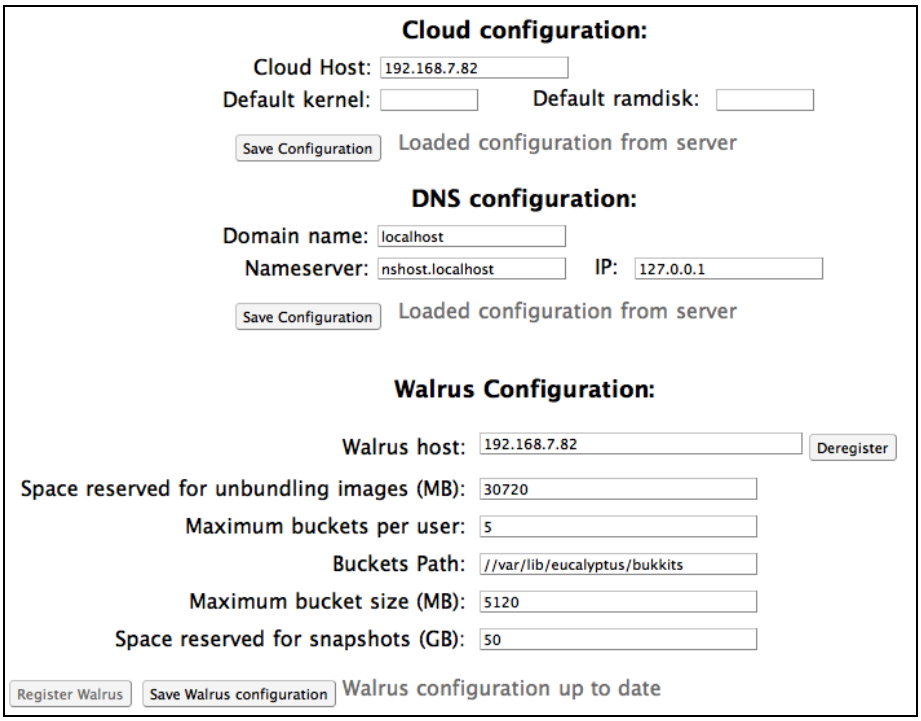


Figure 4.1 The Eucalyptus Configuration interface lets you adjust the configuration parameters for Cloud Host, DNS, Walrus, Clusters, and VM Types.

4.2.1 Reserving Space for the Image Cache in Walrus

Walrus reserves by default 30GB of space for an image cache. Thus, if the sum total size of all images to be cached in Eucalyptus exceeds 30 GB, you will want to increase the "Space reserved for unbundling images" parameter (in the Walrus section of the Configuration interface) to a value larger than the sum total size of all the cached images. For example, you may have a total of three images: two 10GB images and one 30 GB image. In order to ensure that all three images are cached and read to run in Eucalyptus, you will need to set the "Space reserved for unbundling images" in Walrus to 50 GB or larger.

4.3 Configuring Windows VM Support

There are no special configuration requirements for Eucalyptus EE with Windows support beyond the registration and initial configuration steps.

For information on preparing (bundling/uploading/registering) Windows images and running Windows instances in Eucalyptus EE, see the *Eucalyptus EE User Guide*.

For information on creating Windows images to run in Eucalyptus, see *Appendix A: Creating Windows Images for Eucalyptus.*

4.4 Configuring VMware Support

Configuring Eucalyptus EE with VMware Support involves a few special steps that enable the Eucalyptus VMware Broker component to properly interact with vSphere infrastructure components. In all cases the VMware Broker must be installed on the same machine as the Eucalyptus Cluster Controller (CC). (Note that this documentation focuses on a "single cluster" Eucalyptus setup where all components except node controller (NC) are installed on the front end. In more advanced Eucalyptus setups, the CC and VMware Broker may be installed on a machine other than the front end.)

Initial Configuration:

- On the front end, enter the following **`euca_conf`** command:

```
[root@clc]# euca_conf --configure-vmware
```

The output of the above command prompts you for the same parameters that the vSphere client application requests at startup. (Note that lines starting with "log4j:ERROR" can be ignored). Input the requested parameters, making sure to specify the full URL and not just the hostname. For example:

```
Please, supply vSphere endpoint URL: https://192.168.7.70/sdk
Please, supply vSphere username: Administrator
Please, supply vSphere password: ****
```

The endpoint can be either vCenter or ESX/ESXi node. (We recommend using vCenter as it is easier to configure and can be more efficient.)

If `euca_conf` has successfully connected to the endpoint, then a list of discovered hosts appears as shown in the following example:

```
[root@clc]# euca_conf –check vmware
Discovered hosts:
   192.168.7.207 login=Administrator datastoreName=datastore1 (4)
uploadViaHost=null
   192.168.7.213 login=Administrator datastoreName=datastore1 (3)
uploadViaHost=null
   192.168.7.198 login=Administrator datastoreName=datastore1 (5)
uploadViaHost=null
[root@clc]#
```

4.4.1 Testing VMware Broker Configuration

Running `euca_conf --configure vmware` generates an XML-based configuration file (named **`vmware_conf.xml),`** which directs the interaction of the VMware Broker with vSphere infrastructure components. If the VMware Broker does not find this file or is unable to parse it, the Broker will not become active and no vSphere resources will be available through Eucalyptus. You can check if the VMware Broker is reading **`vmware_conf.xml`** and connecting to the VMware endpoints using the **`euca_conf --check vmware`** command, as shown in the following section.

To test Vmware Broker configuration:

- Enter **`euca_conf --check vmware`**. The following example shows an error message indicating that VMware Broker is unable to contact the VMware endpoint:

```
[root@clc]# euca_conf --check vmware
Network mode: SYSTEM
ERROR: no vmware endpoints could be contacted (check url, login,
and password)
[root@clc]#
```

  To find the source of the above error, try scanning the contents of the **`cloud-output.log`** file at **`/var/log/eucalyptus/cloud-output.log`**.

  You may also want to examine (and if necessary edit) the contents of `vmware_conf.xml` to verify that the XML markup is structured properly and that parameters are specified correctly. For example, you may wish to specify the name of a vSphere datastore for the VMware Broker to use. (VMware Broker will choose the first datastore available to it by default). Or you may wish to specify the name of a vSphere network to use (again the Broker will select the first network available to it by default). For more information on configuring `vmware_conf.xml`, see *Section 4.4.2 Advanced Configuration of VMware Broker.*).

  When the VMware Broker is connected to the vSphere endpoints, **`euca_conf --check vmware`** will return a list of discovered hosts as shown in the preceding section. Note that after making changes to the `vmware_conf.xml` file you must restart the Cloud Controller as shown:

```
[root@clc]# /etc/init.d/eucalyptus-cloud restart
```

4.4.2 Advanced VMware Configuration

This section may be skipped if the simple configuration file produced by **`euca_conf --configure-vmware`** was sufficient to discover all the hosts in your vSphere infrastructure that you would like to use and if no adjustments to the configuration parameters are necessary. If, however, default parameter values for the VMware Broker are not sufficient or if more than one endpoint must be used (e.g., when targeting ESX/ESXi nodes directly) then the VMware Broker configuration file, **`vmware_conf.xml`**, will have to be edited manually.

As its extension indicates, **`vmware_conf.xml`** is written in XML. The validity of the XML syntax, including its structure and names of elements and attributes, is strictly enforced in that the Broker will refuse to work if the file is not valid. To check if the file is valid one can either check **`cloud-output.log`** for errors or run **`euca_conf`** as follows:

```
[root@clc root]# euca_conf --check vmware
```

(For more information on testing VMware Broker configuration, see *Section 4.4.1: Testing VMware Broker configuration*.)

XML File Structure (Hierarchy of Elements)

The part of the file that describes vSphere endpoints can be hierarchical, reflecting the hierarchy of abstractions defined within vSphere: endpoints may contain datacenters, datacenters may contain clusters, and clusters may contain hosts. However, just as parts of the hierarchy are optional in vSphere (e.g., there may be one default datacenter and no clusters) the hierarchy is optional in the VMware Broker configuration file. The only required element is **`<endpoint>`**. When other elements are present, they must be arranged relative to each other as follows:

```
<configuration>
   <vsphere cacheLimitMb=”......” CREDENTIALS EXTRAS>
      <endpoint url="https://..." CREDENTIALS EXTRAS discover=BOOLEAN>
         <datacenter name="....." CREDENTIALS EXTRAS discover=BOOLEAN>
            <cluster name="....." CREDENTIALS EXTRAS discover=BOOLEAN>
               <host name="....." CREDENTIALS EXTRAS />
            </cluster>
         </datacenter>
      </endpoint>
   </vsphere>
   <paths scratchDirectory="/path" scratchDirectoryLimitMb=”...”
          cacheDirectory="/path"   cacheDirectoryLimitMb=”...” />
</configuration>
```

For example, if a <datacenter> is specified, it must be contained by the <endpoint> to which it belongs. Likewise, any <cluster> must be contained within an <endpoint>, if any. And so on.

Specifying Element Attributes

The only parameter unique to the <vsphere> element is cacheLimitMb, which specifies how much space Eucalyptus is allowed to use on vSphere, cumulatively across all datastores, for caching VM templates. The default value is 50GB.

Each <datacenter>, <cluster>, and <host> element requires the '**name**' attribute, which must match the name of that abstraction in vSphere; whereas <endpoint> requires the '**url**' attribute, which is normally the IP of a vSphere endpoint prefixed by "https://".

CREDENTIALS and EXTRAS are categories of attributes. These attributes can be specified for any vSphere-related elements with values propagating from higher-level elements to lower-level elements, where the values can be overridden selectively. For example, if one were to specify maxCores="4" in the <endpoint> element, then all hosts belonging to that endpoint would advertise 4 cores instead of their actual number of physical cores. However, the lower-level parameter always overrides the higher-level parameter. So, if a <host> specifies swapSizeMB= "1024", that will override swapSizeMB="256" specified in the containing <datacenter>.

CREDENTIALS consist of 'login' and 'password' attributes, the latter of which can be specified in plaintext or encrypted (as produced by euca_conf --configure-vmware). At the very least they must be specified either for each <endpoint> or once in the enclosing <vsphere> element, in which case they will be used for all endpoints without explicitly specified credentials. If credentials are specified for any elements contained by <endpoint>, they will be used for the optional data transfer connections to individual ESX/ESXi nodes (see uploadViaHost attribute below).

Three elements – <endpoint>, <datacenter>, and <cluster> – may specify the **BOOLEAN** attribute 'discover' (with "true" and "false" as the only allowed values). Setting it to "true" implies that VMware Broker is allowed to add to its inventory any elements (clusters or hosts) contained therein even if they are not specified explicitly. Conversely, setting it to "false" implies that VMware Broker may not add to its inventory any containing elements that are not specified explicitly with <cluster> or <host> tags. If a host is not added to the inventory because discovery is forbidden and the host is not specified explicitly with a <host> element, that incident will be reported as "**DISALLOWED BY CONFIGURATION**".

By default Eucalyptus will attempt to use all resources that it discovers, such as memory, cores, and storage space on a datastore. Furthermore, when multiple

options are available, e.g., for a datastore or a network, it will make an arbitrary choice. Parameters in the **EXTRAS** category allow the administrator to restrict Eucalyptus's behavior as needed.

- 'datastore' - name of the vSphere datastore to use (first one found by default)
- 'network' - name of the vSphere network to use (first one found by default)
- 'maxCores' - number of virtual cores on a host (same as physical cores by default)
- 'maxMemMB' – memory, in MB, available for use by Eucalyptus instances (same as physical RAM by default)
- 'swapSizeMB' - swap size, in MB, to be created for instances (512MB by default)
- 'maxDiskMB' – disk size, in MB, available to Eucalyptus (free space by default)
- 'uploadViaHost' – upload directly to the host rather than through the endpoint ("false" by default)

The last option, uploadViaHost, applies only to setups using vCenter. The default behavior is to upload the VM's disk files through the endpoint. To avoid overloading the vCenter with I/O traffic, however, Eucalyptus can perform the upload directly to an individual host. In this case, if the credentials (either login or password) for the host are different from vCenter credentials, they must be specified explicitly in one or more elements contained by the `<endpoint>` (e.g., in each `<datacenter>` or each `<cluster>` or each `<host>` element).

Changing Cache Location and Size

You can change the two disk locations and the size limits used by VMware Broker for constructing and caching of disk images:

Disk Location:

- `scratchDirectory=/var/lib/eucalyptus/vmware/tmp`
- `cacheDirectory=/var/lib/eucalyptus/vmware/cache`

Disk Size:

- `scratchDirectoryLimitMb="50000"`
- `cacheDirectoryLimitMb="50000"`

The above values can be specified as attributes of the optional element `<paths>`. If the `<paths>` element is used, then both it and other vSphere-specific elements must be contained by the single top-level `<configuration>` element.

Adding ESX/ESXi Hosts Directly

When adding ESX/ESXi hosts directly, make sure `vmware_conf.xml` is properly formatted to show the additional ESX/ESXi hosts and includes the appropriate username, password, and datastore for each host, for example:

```
<configuration>
   <vsphere username='root' password='foobar'>
      <endpoint url="https://esx1/sdk" datastore='datastoreA'/>
      <endpoint url="https://esx2/sdk" datastore='datastoreB'/>
      …
```

4.5 Configuring SAN Support

4.5.1 Enabling SAN support

Using Eucalyptus EE with SAN support requires enabling the SANManager module, which directs the Storage Controller (SC) to manage appropriately credentialed SAN devices. Enabling the SANManager requires configuring the `CLOUD_OPTS` variable in `eucalyptus.conf` so that Eucalyptus loads the SANManager module and selects your SAN device at start time, as shown:

1. In your editor, open the file `$EUCALYPTUS/etc/eucalyptus/eucalyptus.conf`

2. Configure `CLOUD_OPTS` for your particular SAN device, as shown:

For Dell Equallogic SANs:

```
CLOUD_OPTS="--java-home=/opt/packages/jdk \
            -Debs.storage.manager=SANManager \
            -Debs.san.provider=EquallogicProvider”
```

For NetApp SANs:

```
CLOUD_OPTS="--java-home=/opt/packages/jdk \
            -Debs.storage.manager=SANManager \
            -Debs.san.provider=NetappProvider”
```

3. Start Eucalyptus (or the SC if SC is installed separately.) Note that if Eucalyptus is already started, you must stop the SC, edit `eucalyptus.conf,` then start the SC.

4.5.2 Configuring SAN Credentials

Now that you have enabled the SANManager module, you must login to the Eucalyptus Web interface and configure your cluster(s) to interact with your SAN device(s), as shown:

1. Login to the Eucalyptus web interface at https://<IPAddress>:8443 as “admin.”

2. Click on the Configuration tab.

3. In the Clusters section, enter the hostname or IP address of the SAN device, along with the administrative username and password: (**Figure 4.2**).

   If you are using a NetApp SAN, you will see an additional field "Reserved Aggregate." Enter here the name of the NetApp SAN aggregate you want Eucalyptus to use.

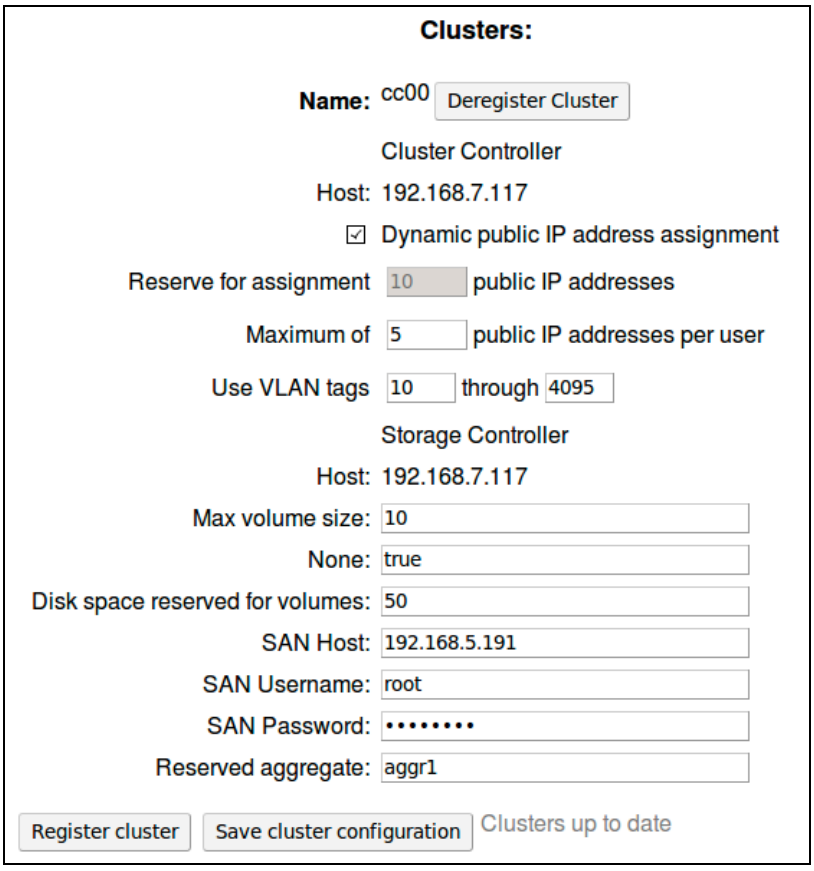


Figure 4.2 Cluster configuration section of the Eucalyptus Configuration interface window. To configure a cluster to interact with your SANs: enter SAN hostname or IP address, SAN username and password. For NetApp SANs: set "Reserved aggregate" to the aggregate you want Eucalytpus to use.

Your Eucalyptus cluster is now appropriately configured to interact with your SAN device(s). You can now proceed with running Eucalyptus block storage commands (e.g., `euca-create-volume`, etc.) with your integrated SAN device(s).

4.6 About Client-tools Configuration

To use the Eucalyptus system with client tools, you must obtain user credentials from the 'Credentials' tab in the Eucalyptus interface. 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 these

credentials with Euca2ools, Amazon EC2 tools and third-party tools like those from Rightscale.com.

For detail instructions on installing, administering, and using Eucalyptus client tools (Euca2ools) see the *Eucalyptus EE User Guide, Section 3: Getting Started.*

Part III – Managing Your Eucalyptus EE Cloud

Part III focuses on managing your Eucalyptus cloud environment. First we introduce the elements of the Eucalyptus Web interface. Then we take a closer look at the *User and Group* management interface, which provides convenient graphical tools for managing users and groups, and associating them with availability zones (clusters). Next, we introduce the *Accounting Reports* interface, where you can obtain statistical information on cloud operations as a means for auditing and a basis for allocating cloud resources. Finally we present alternate command line tools for performing user, group and system management tasks.

Who should read Part III?

Part III is directed at system, network and cloud administrators tasked with providing and managing cloud services for end users. It is also geared towards data center managers and other IT professionals involved in cloud resource allocation, auditing, acquisition, and management.

Part III discusses these topics:

- Eucalyptus EE Web interface
- Users and Groups interface
- Managing groups
- Managing users
- Associating groups with availability zones (clusters)
- Accounting Reports interface
- System Events logs
- Resource Usage reports (IP addresses, instances, and block storage)

Section 5: The Eucalyptus EE Web Interface

Eucalyptus EE provides a convenient Web interface, which provides cloud administrators with a graphical means of performing several cloud management tasks. To access the interface, the administrator must log in to Eucalyptus with username and password at:

https://<IPAddress>:8443

Once logged in, the *Your Eucalyptus Cloud* page appears, displaying six tabs across the top: *Credentials*, *Images*, *Users and Groups*, *Reports*, *Configuration*, and *Extras* (**Figure 5.1**). Each of these tab pages is introduced below.

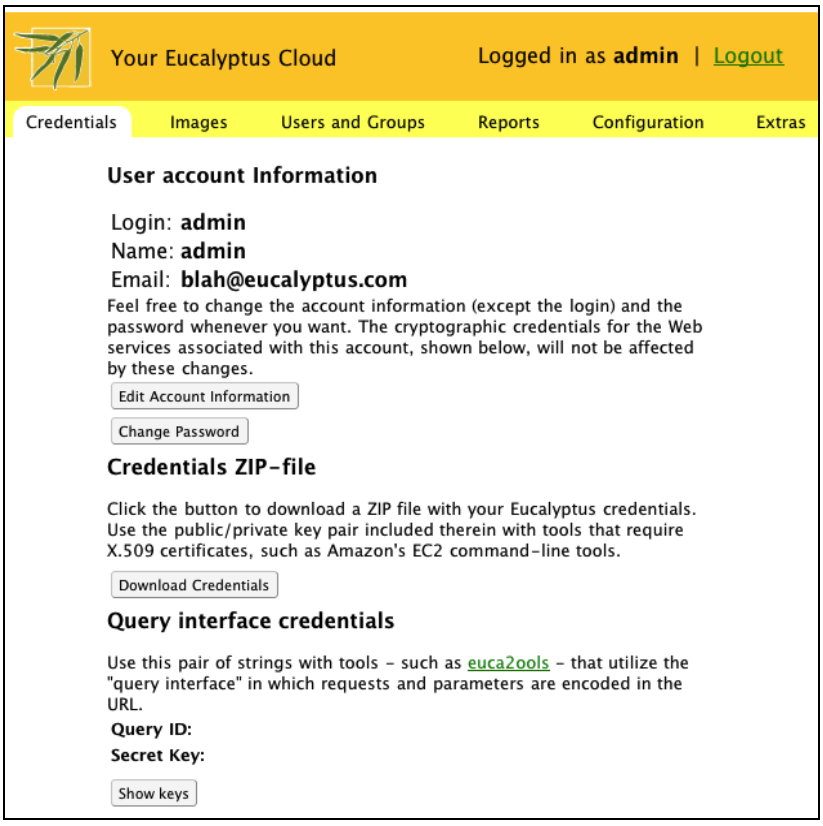


Figure 5.1 The Eucalyptus web interface.

5.1 Credentials

The *Credentials* page provides User account information for the administrator account; a "Download Credentials" button (for downloading a Credentials ZIP-file); and a "Show Keys" button for quickly viewing "Query interface credentials"(see **Figure 5.1**). For information on obtaining credentials, see the *Eucalyptus EE User's Guide, Section 3.3: Obtaining Credentials.*

5.2 Images

The *Images* page a list of images prepared by the administrator or users. (When you first install Eucalyptus, this list is empty.) User can select from this list to download and run as VMs in the Eucalyptus cloud. For more information on managing and working with images in Eucalyptus, see the *Eucalyptus EE User's Guide*.

5.3 Users and Groups

The *Users and Groups* page provides a graphical interface where administrators can quickly create groups that correspond to business entities (i.e., Engineering, Accounting, Publishing), add and delete users from groups, and associate availability zones (clusters) with groups. For more information, see *Section 6: Users and Groups management.*

5.4 Accounting Reports

The Accounting *Reports* page provides administrators with statistics of user activities and system resource usage. This lets administrators know what resources are being consumed by users and groups and provides a means for auditing and a basis for resource allocation. For more information, see *Section 7: Accounting Reports.*

5.5 Configuration

The *Configuration* page provides an interface where cloud administrators can set cloud configuration parameters for cloud host addresses, DNS, Walrus (bucket storage), clusters and SAN devices. For more information, see *Section 4.2: Initial Configuration of Eucalyptus EE.*

5.6 Extras

The *Extras* page includes miscellaneous downloads pertaining to Eucalyptus, including Eucalyptus-certified images and Eucalyptus-compatible tools.

Section 6: Users and Groups Management

6.1 Users and Groups Interface

Eucalyptus EE introduces a new *Users and Groups* interface that lets administrators add, modify, and delete groups; add, modify, and delete users; and associate availability zones (clusters) with groups. To access Users and Groups, log in to the Eucalyptus Web interface at **https://<IPAddress>:8443/** and click the Users and Groups tab.

6.2 Managing Users

6.2.1 User Self Sign-up

To activate the Eucalyptus website for use by others, the administrator must perform an initial login and designate an email address for user application requests. At this point an "Apply for account" link will appear on the login page.

When a user completes an application form, an email is sent to the administrator containing two URLs: one for accepting and one for rejecting the user. Note that no authentication is performed on people who fill out the form. The administrator must perform the 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.) Thus, account approval is at the administrator's discretion. Administrators can also use additional user-submitted information (e.g., telephone number, role, department) to help verify the user's identity.

Accepting or rejecting a signup request causes an email 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 must log in with the username and password submitted at signup. Users interested in joining the cloud should be directed to the Eucalyptus Web interface at

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

6.2.2 Activating User Accounts

After a user applies for an account via the Eucalyptus Web interface (by clicking the "apply for account" link and filling out the application form), the user's name will appear in the User list of the Users and Groups interface. (The user's name will

appear in gray text indicating the user is not yet approved.) The administrator can activate the user's account by clicking on the user name, clicking the edit button, and checking the "approved" checkbox in the Edit window to the right. The administrator can also check "skip email confirmation" to forgo the email confirmation process and check "enable" to allow the user immediate login access to Eucalyptus.

6.2.3 Adding Users

In addition to approving and rejecting user requests via email, administrators can quickly add, approve, enable, modify, and delete user accounts, as well as add users to specific groups, via the Users and Groups management interface, as shown:

1. In the Users column, click Add.

2. In the Adding user column, enter the requested user information, including the Group to which you wish to assign the user. The following example shows the adding of a new user "User4" to group "Engineering" (**Figure 6.1**):

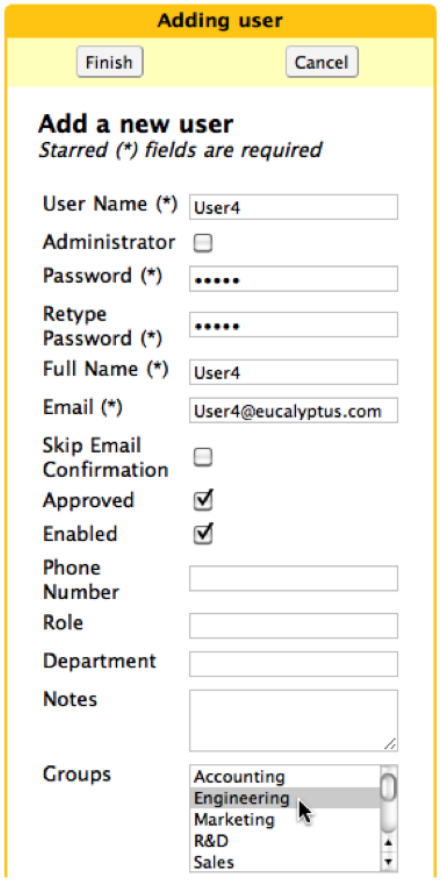


Figure 6.1 The Adding user window lets you add new user accounts and assign new users to groups.

Note that the names of users that have not yet been approved by the administrator (or whose accounts are disabled) appear in gray text in the Users list. In addition, users without a group specification are added to the group "default," and all users are added automatically to the group "all."

You can edit a user's account to disable or delete the user, or add the user to additional groups. To do so, click on the user name, and click the Edit button at the top of the window to the right. Note that you can use Ctrl-click to add a user to multiple groups simultaneously.

Editing Multiple Users at Once

You can also perform actions on multiple user accounts at the same time, including add users to groups, remove from groups, approve, enable, disable, and delete. To perform actions on multiple users, select the users names you wish to edit in the Users list. Next, in the Edit window to the right, click the appropriate button at the top of the window to perform the desired action.

For example, you may have multiple new user sign-up requests (indicated by new user names appearing in gray text in the User list). To grant approval to multiple users simultaneously: in the Users list, select the user names you wish to approve; then click the "approve" button at the top of the Edit window to the right (see **Figure 6.2**).

Figure 6.2 To perform actions on multiple users simultaneously: In the Users List, select the user names you wish to perform the action on; then click the button of the action you wish to perform in the Edit window to the right of the screen.

6.3 Working with Groups

Group is a new concept introduced in Eucalyptus EE. Administrators can create groups that correspond to business entities (e.g., Engineering, R&D, Marketing, Accounting) and assign the group "permissions" that grant access to specific clusters. By associating groups with clusters, and users with groups, the administrator can effectively restrict user access to cloud resources on a per group basis.

6.3.1 Adding Groups and Assigning Clusters to Groups

You can add, delete, and edit groups, and assign availability zones (clusters) to groups, using the Users and Groups interface, as shown:

1. In the Groups column click Add.

2. In the Adding group column, enter the name of the group, select an availability zone, and click Finish. The example shows the assigning of availability zone "testcluster" to a new group "Publishing" (**Figure 6.1**):

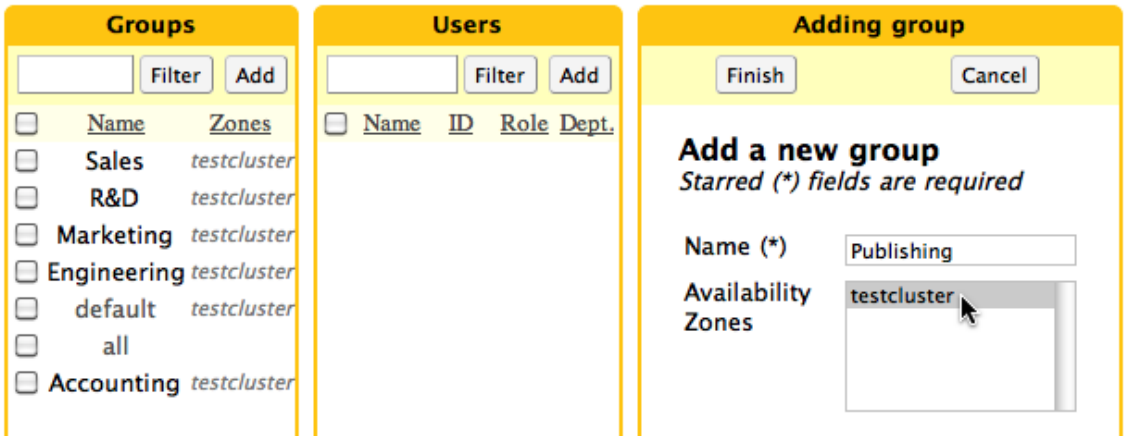


Figure 6.1 The Users and Groups management interface lets you create new groups, add users to groups, and assign availability zones to groups.

Two special groups appear in gray text in the Groups list: "default" and "all." Eucalyptus automatically adds users not assigned to a group to the "default" group; while the "all" group includes all users added to the cloud. In addition, you can modify a group to change availability zone assignments, or assign additional availability zones to the group. To do so, click on the group name, and click the edit button in the group profile column on the right side of the screen. To assign multiple availability zones to a group: Ctrl-click on the desired availability zone names.

Section 7: Accounting Reports

Eucalyptus 2.0 EE introduces a new *Accounting Reports* web interface, which lets administrators view statistics of Eucalyptus cloud activities and resource usage. Administrators can use this information as a basis for auditing, resource allocation, and cloud planning.

7.1 Report Types

The Accounting Reports interface gives you access to multiple types of reports on cloud operations, including *System Events*, *Resource Usage* (Storage, VM instances), and *Registered Components*. You can display a report over a selected time period by inputting a date range in the Reports control panel as shown below (**Figure 7.1**).

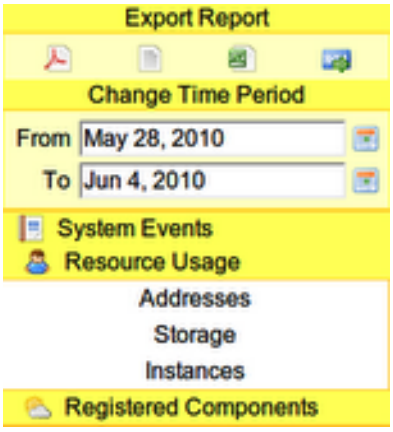


Figure 7.1 The reports control panel lets you select reports on cloud activities and resource usage over selected time periods.

The Accounting Reports interface lets you view the following reports:

7.2 System Events

System Events includes four interrelated types of logs that reflect the history of events the system has processed: The *System Event* log is an aggregated history of all events that change the state of the system (for example adding users/groups or restarting the system); The *Message Log* shows a history of all user requests (commands) and the amount of time elapsed in servicing each request. The *Component Log* and *Users & Groups log* show filtered subsets of the System Event log for system components and users & groups, respectively. System Events can be used to review changes made to the configuration of the system or in combination with other log files (e.g., `cloud-output.log`) to help troubleshoot problems.

7.3 Resource Usage

Resource Usage reports display the quantity and duration of use of two types of resources: *Instances*, and *Storage*. Given a selected reporting interval, these reports provide the number of allocations, the number of units allocated/de-allocated, and the amount of unit-time consumed by each user and group of a corresponding resource.

Instances reports show information about the usage of virtual machines and provide resource usage summaries with corresponding user and group aggregates of vmtype-hours. Storage reports display information about the usage of Block Storage volumes on storage controllers in the Eucalyptus cloud. Consider the following Block Storage report example (**Figure 7.2**):

Group	Block Storage				
	Volumes	GB Allocated	GB Deleted	GB-Hours	Snapshots
default	14	16	16	13	0
testgroup	13	17	17	13	0
testgroup2	2	4	4	1	0
User	**Block Storage**				
	Volumes	GB Allocated	GB Deleted	GB-Hours	Snapshots
admin	12	12	12	12	0
testuser2	2	4	4	1	0
testuser1	1	5	5	1	0

Figure 7.2 Storage reports let you view statistics on user and group usage of Block Storage resources in the Eucalyptus cloud.

The top table shows data aggregated at the scale of a group. The bottom table shows the same information on a per-user basis. The first column, “Volumes,” shows the number of volumes created during the selected time interval. The second and third columns (“GB Allocated” and “GB Deleted”) show the number of gigabytes, which were newly allocated and deleted, respectively. The third column shows the number of gigabyte-hours consumed. This value should be interpreted as: (# of hours allocated) × (size of volume in GB), where the number of hours is rounded up to the nearest hour. The fifth column, “Snapshots,” reports the number of snapshots created during the time interval.

Part IV – Advanced Configuration and Troubleshooting

Part IV shows you how to configure Eucalyptus' advanced VM networking subsystem. You will learn the capabilities of the four networking modes: SYSTEM, STATIC, MANAGED, and MANAGED NOVLAN; and how to use these modes to setup VM networks that achieve organizational security goals. The final section of this Administration guide addresses several common issues that arise and provides suggestions for troubleshooting Eucalyptus EE.

Who should read Part IV?

Part IV is geared for system administrators, network administrators, data center managers, and other IT professionals in positions of system, network, and security oversight and maintenance.

Part IV discuses these topics:

- Eucalyptus EE network configuration
- Networking modes
- Bridge names
- Configuring networking modes
- Troubleshooting Eucalyptus

Section 8: Eucalyptus EE Network Configuration

8.1 Overview of Eucalyptus Networking Modes

Eucalyptus EE includes a highly configurable VM networking subsystem that can be adapted to a variety of network environments. There are four high level *networking modes*:

- **SYSTEM**
- **STATIC**
- **MANAGED**
- **MANAGED NOVLAN**

Each mode has its own set of configuration parameters, features, benefits, and, in some cases, requirements placed on your local network. The administrator must select one of these four modes before starting Eucalyptus by modifying the *eucalyptus.conf* configuration file on each machine running a Eucalyptus component. A brief description of each mode follows:

SYSTEM

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

STATIC

This mode offers the Eucalyptus administrator more control over VM IP address assignment. Here, the administrator configures Eucalyptus with a 'map' of MAC address/IP address pairs. When a VM is instantiated, Eucalyptus sets up a static entry within a Eucalyptus controlled DHCP server, takes the next free MAC/IP pair, assigns it to a VM, and attaches the VMs Ethernet device to the physical Ethernet through the Xen bridge on the nodes (in a manner similar to SYSTEM mode). This

mode is useful for administrators who have a pool of MAC/IP addresses that they wish to always assign to their VMs.

Note that running Eucalyptus in SYSTEM or STATIC mode disables some key functionality, including the following: The ability to define network access for sets of VMs (termed *security groups* in Amazon EC2); user-controlled, dynamic assignment of IPs to instances at boot and run-time (termed *elastic IPs* in Amazon EC2); isolation of network traffic between VMs (i.e., VMs can become vulnerable to intrusion; and the availability of the meta-data service (use of the http://169.254.169.254/ URL to obtain instance specific information).

MANAGED

This mode provides the broadest range of features of the four networking modes. In MANAGED mode, the Eucalyptus administrator defines a large private network from which VM instances draw their IP addresses. As with STATIC mode, Eucalyptus maintains a DHCP server with static mappings for each VM instance that is created. Eucalyptus users can define a number of 'named networks', or 'security groups', to which they can apply network ingress rules. When a user runs a VM instance, they specify the name of a network to which a VM belongs and Eucalyptus selects a subset of the range of IPs where other VMs in the same 'network' can reside. A user can specify ingress rules that apply to a given 'network', such as allowing ping (ICMP) or SSH (TCP, port 22) traffic to reach their VMs. This lets Eucalyptus expose a capability similar to Amazon's 'security groups'. In addition, the administrator can specify a pool of public IP addresses that users may allocate, then assign to VMs either at boot or dynamically at run-time. This capability is similar to Amazon's 'elastic IPs'. Eucalyptus administrators that require security groups, elastic IPs, and VM network isolation must use MANAGED mode.

MANAGED-NOVLAN

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

WARNING: If you edit a networking related value in eucalyptus.conf, you will need to perform a clean restart of the CC (`$EUCALYPTUS/etc/init.d/eucalyptus-cc cleanrestart`) for changes to take effect. Make sure to terminate all VM instances before performing the clean restart.

8.1.1 About Network Configuration Examples

In the remainder of this section, we provide configuration examples for each of the four Eucalyptus networking modes. We use two different network configurations loosely based on our Eucalyptus Community Cloud. 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 8.1 has a very simple configuration: all the machines have one Ethernet device (eth0) and they are all connected directly to the public network.

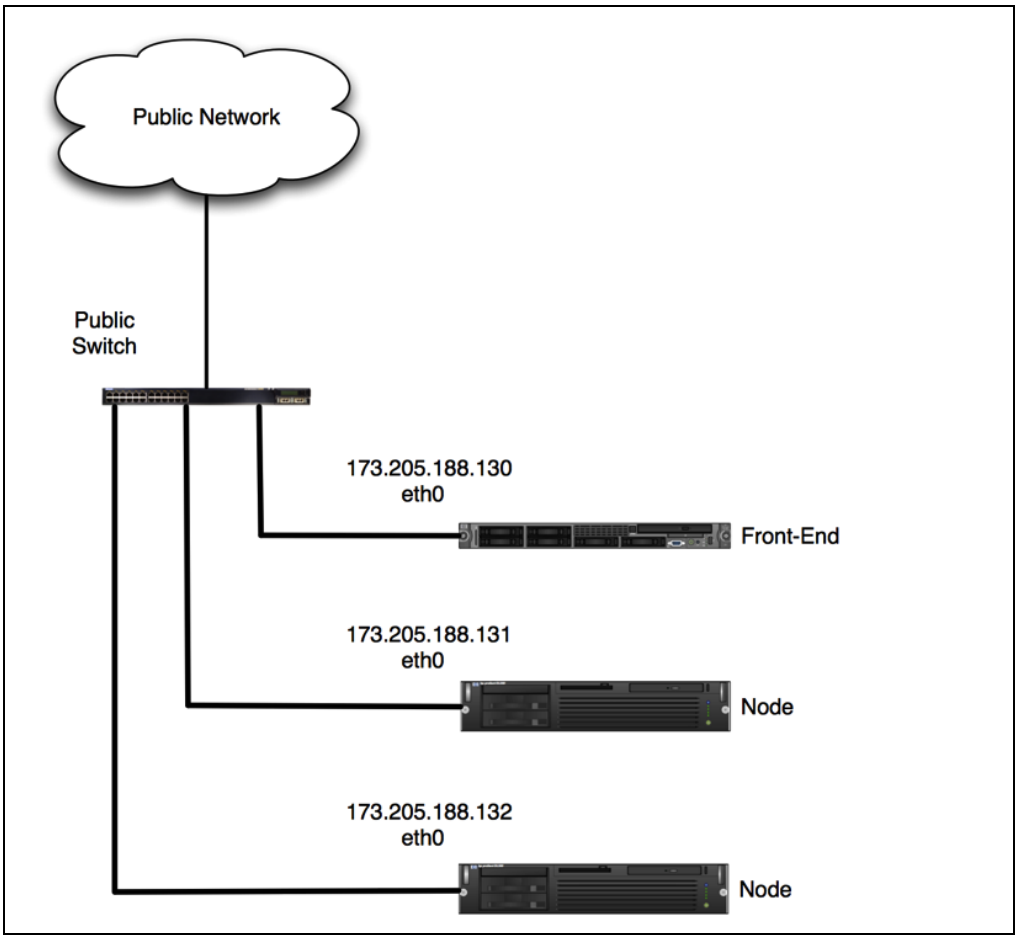


Figure 8.1 The above network configuration shows all machines directly attached to the Public Network.

In addition to the public network, the configuration in **Figure 8.2** uses a private network for some of the machines (which will be the node controller in these examples). One machine uses NAT to provide access to the public network to the machines on the private subnet acting as their gateway: that machine will be the front-end machine for our examples. The front-end machine has 2 Ethernet devices: eth0 is on the public network and eth1 is on the private network. The private network is in the 10.0.1.0/24 range.

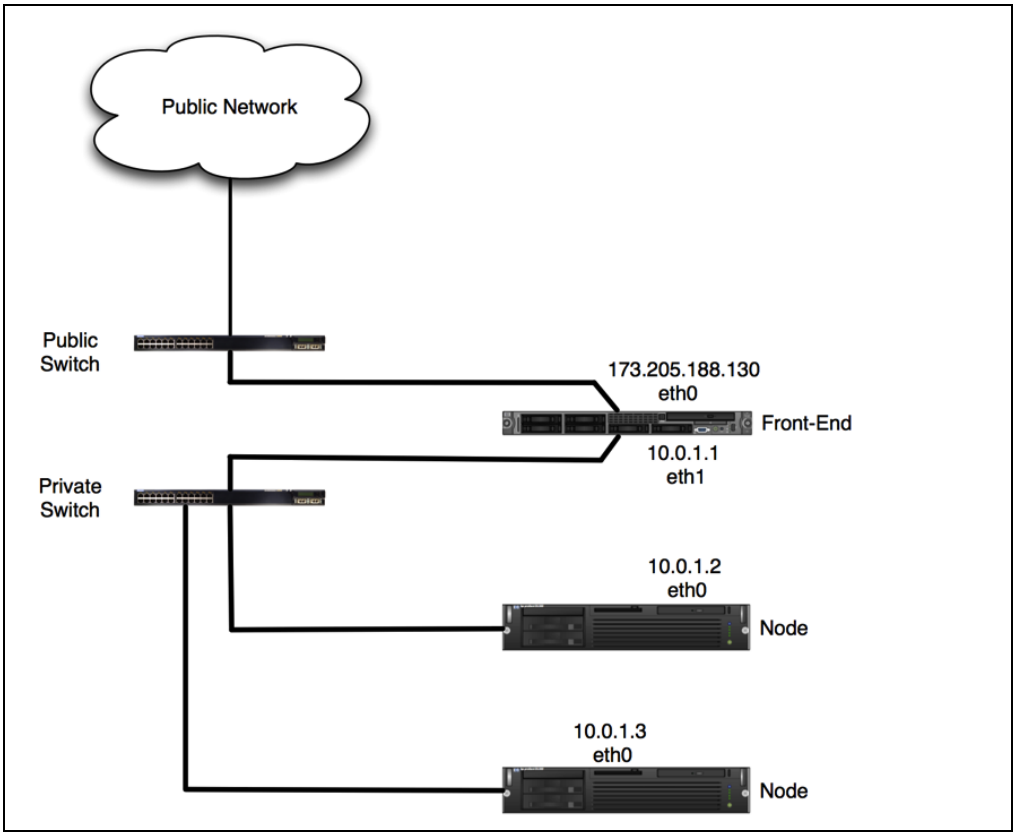


Figure 8.2 The above network configuration shows the node machines on private subnet using the front-end machine as the gateway to the Public Network. Note that the front-end machine has two Ethernet devices and uses NAT to allow the nodes access to the Public Network.

8.1.2 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
xenbr0          8000.feffffffffff       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 such bridge is named **xenbr0,** as shown above.

8.1.3 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 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. Note that this number 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_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.”

8.1.4 About DHCP Server

Most Eucalyptus networking modes require an installed DHCP server binary. The DHCP server package must be installed on the front end and must be compatible with ISC DHCP Daemon version 3.0 (this daemon is provided with most Linux distributions, for example with 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 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>"
```

8.2 Using Networking Modes

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

8.2.1 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 network isolation between instances

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 8.1**. 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"

8.2.2 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 network isolation between instances

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"
```

To configure Eucalyptus for your DHCP server, see *Section 8.1.4: About DHCP server.*

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 in *Section 8.1.3: 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 example:

Configuration Example:

For our STATIC mode example we refer to **Figure 8.1**. 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"

8.2.3 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, and elastic IPs. Note that in this mode each security group requires a separate VLAN, which Eucalyptus controls and maintains.

Requirements

- There is an available *private* subnet that is completely unused on the network (192.168..., 10....., other)
- 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 *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

- None.

Testing VLAN-clean

The admin must verify that the local network is VLAN clean (allows/forwards VLAN tagged packets) between machines running Eucalyptus components. To verify, perform the following test:

First, choose two IP addresses from the *private* subnet you plan to use with Eucalyptus. (In following example, those IP addresses are: 192.168.1.1 and 192.168.1.2). Next, on the front end, 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).

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 server, see *Section 8.1.4: About DHCP server.*

It is also necessary to instruct Eucalyptus about the unused *private* subnet. To do this, you must configure VNET_SUBNET, VNET_ADDRESSPERNET, etc., as described in *Section 8.1.3: 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 8.2**. 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"

Note that 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 simultaneous active named networks that can be in use at any one time (65536 / 32 = 2048). This directly correlates to the maximum number of security groups possible inside Eucalyptus. In this example, the maximum number of instances belonging to the same security group is 29 (32 less the network address, broadcast address, and router address).

8.2.4 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 *private* subnet that is completely unused on the network (192.168..., 10....., other)
- 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

- No VM network 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 *Section 8.2.3.*

Configuration Example

For this example, we refer to the network configuration shown in **Figure 8.2**. 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"
```

8.3 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 mode - Your CC will need to be configured with a bridge as its primary, public interface (VNET_PUBINTERFACE) in order for vtun tunneling to work in this mode.

MANAGED and MANAGED-NOVLAN modes - The CC attempts to auto-discover it's 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.

8.4 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 `VNET_LOCALIP` 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.

Section 9: Troubleshooting Eucalyptus EE

9.1 How to Troubleshoot Eucalyptus EE

To troubleshoot Eucalyptus EE, 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. 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

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/messages`**, **`/var/log/libvirt/`**, **`/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 convey more information than user credentials. For example **euca-describe-instances** give you additional information, including all instances running by all users on the system. Thus, make sure you have Euca2ools installed with proper administrator credentials.

9.2 Common Eucalyptus Problems and Solutions

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).

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
```

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.

Are the NCs running correctly?

First, check that the CC 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 `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.).

Is libvirt configured correctly?

For information on proper configuration of libvirt, see *Appendix B.3*.

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).

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.

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.

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).

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.).

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).

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.

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.

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/CPUs, and the NC has only 2 cores/CPUs 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).

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.

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`**.

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).

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.

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
```

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.

9.3 Troubleshooting VMware

Error messages appearing in the "Recent Tasks" section of vSphere client:

Create Virtual Machine*: The name 'i-XXXXXX' already exists.*

This may happen if Eucalyptus-created instances have been modified through vSphere client (which is not recommended!). The solution is to wait for Eucalyptus to reach steady state (no pending instances) and to delete, using the vSphere client, all non-running Eucalyptus instances, both in inventory and on disk. Eucalyptus instances start with 'i-' and 'emi-'. Browse the datastore to ensure that there are no files that are unaffiliated with a VM.

Software Internet SCSI enabled*: Error during the configuration of the host: Failed to enable software iSCSI: No vmkernel nic configured*

This may happen if the user attempted to attach an EBS volume to an instance running on the ESX/ESXi host that has not been configured for iSCSCI. For solution, see *"Configure ESX node for iSCSI" in Section 3.1.3: Additional Prerequisites for VMware support*.

Appendix A: Creating a Windows Image for Eucalyptus

This appendix shows you how to create a virtual machine image from a Windows OS for use in Eucalyptus Enterprise Edition (EE) with Windows VM Support. Eucalyptus EE is compatible with images created from licensed versions of Windows Server 2003, Windows Server 2008, and Windows 7. For instructions on running and managing Windows images in Eucalyptus EE, see *the Eucalyptus Enterprise Edition User Guide*.

About Windows Images

Windows OS is sensitive to physical and virtual hardware changes made after installation. Thus we advise creating the Windows image on the specific hypervisor (Xen, KVM, or ESX) where you plan to run instances of the Windows image. In addition, to support AWS methods for running Windows instances with randomized Administrative passwords (accessible through EC2 calls), a few extra steps must be followed when preparing the image, as detailed below:

Step 1: Install Eucalyptus with Windows VM Support

Before you begin, make sure that Eucalyptus EE with Windows VM support is installed and properly authenticated on your system. For installation and product licensing instructions, see *Section 3: Installing Eucalyptus.*

Step 2: Install Base Windows OS

It is important to install a base Windows OS using the same hypervisor technology and configuration you will be using within Eucalyptus. The best way to do this is to prepare your Windows VM directly on one of the Eucalyptus nodes. (i.e., if Eucalyptus is using Xen, then prepare the image on a node with Xen.)

Included are two libvirt.xml template files that closely match those that Eucalyptus will generate on VM instantiation time (one for Xen, one for KVM). We recommend you review these files to become familiar with required files/bridges/resources.

1. Create a blank disk file (For Windows 2003, an 8GB disk is sufficient.; for Windows 2008 and Windows 7 the disk size must be 15GB or larger.):

   - For Xen/KVM:

     Use the **dd** command to create your blank disk file. Make sure to specify the name of your Windows VM image during this step using the **of**

parameter (output to file). Note that Windows VM image names must begin with the string "windows" to be usable with Eucalyptus.) In the following example, we create a blank disk file for a Windows VM image named "windows.my2008Server.img":

```
[root@node1]# dd if=/dev/zero of=windows.my2008Server.img \
      bs=1M count=1 seek=16999
```

- For VMware:

  Create a new VM using the vSphere client following standard VMware process. **You must create image using the LSI Logic Parallel driver.**

2. Make available the .iso file on your Windows installation CD/DVD; and the Eucalyptus 'floppy' image file from your Eucalyptus install. The 'floppy' image file is located in:

   `$EUCALYPTUS/usr/share/eucalyptus/floppy`

   - For Xen/KVM:

     Copy the files to your local directory.

   - For VMware:

     Import the Windows .iso and Eucalyptus floppy into a datastore that is visible to the VM and configure the VM (using VSphere client) to use the .iso as the CD drive, and the floppy file as the floppy drive. Remember to select 'connect on power on' for the two drive devices. The first time you boot, make sure to enter BIOS and set the boot order to 'HDD, CDROM, then Removable Devices (floppy).'

3. Modify libvirt-xen.xml or libvirt-kvm.xml files so that paths/bridge names match the local directory. When modifying libvirt files, make sure to provide fully qualified path names. For example, in libvirt-xen.xml, a correct path name for the file 'floppy' in the .xml expression `<source file = 'floppy'/>` should have this form:

   `<source file = '/<path>/floppy'/>`

4. Start the windows install:

   - For Xen/KVM:

```
[root@node1]# virsh create libvirt-<xen/kvm>.xml
```

- For VMware:

  Power-up the VM.

5. Connect to the virtual console:

   - For Xen/KVM:

     Connect to the virtual console using VNC. Check to find the display number that has been allocated by looking at the process table (`ps axw`). For example, if the display number is '0,' then on a machine running Linux/Ubuntu, connect using a VNC client:

     ```
     [root@node1]# vinarge <machine-hosting-vm>:0
     ```

     You can also connect to your Windows image using the *Virtual Manager* and *Virtual Viewer* utilities with Centos/Xen systems)

   - For VMware:

     Connect to the virtual console using vSphere client.

6. Follow standard Windows installation procedure until the VM has completed installing Windows. (Note that it may be necessary to rerun the **`virsh create libvirt-<xen/kvm>.xml`** command (as shown in Step 4 above) multiple times during the Windows installation process.)

   (Note: Before proceeding with the following steps, install any patches, software, etc. on your VM that you wish to include in your instance.)

Step 3: Enable ACPI[1] Shutdown

The next step is to enable ACPI Shutdown using the Windows Group Policy Editor, as follows:

1. Start > Run: gpedit.msc

[1] Advanced Configuration and Power Interface

2. Navigate to: Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Option.

3. Select "Allow system to be shut down without having to log on."

4. Reboot the guest.

Step 4: Configure the Image to Support AWS Password/RDP Functionality

AWS Windows VM instances boot up with a single user (Administrator) provided a unique password for each instance with Microsoft's Remote Desktop Protocol (RDP) enabled.

After installing a Windows image, you must configure the image so that it generates a random Administrator password on boot; and you must enable RDP. This two-step process is slightly different for each version of Windows, as shown below:

Windows 2003:

1. Boot password set:

   a. Start > All Programs > Accessories > System Tools > Scheduled Tasks.

   b. Add New Task that runs "a:\euca.bat" on "When My Computer Starts" (as Administrator).

2. Enable Remote Desktop Protocol (RDP):

   a. Start > Right-click on My Computer > Select Properties > Remote Tab.

   b. Check the 'Enable Remote Desktop' checkbox.

Windows 2008:

1. Boot password set:

   a. Start > Administrative Tools > Task Scheduler.

   b. In the 'Actions' window (right pane) click on 'Create Basic Task.'

      The Create Basic Task wizard opens.

   c. In the 'Name' field enter: "Random Boot Password." Click Next.

   d. In the 'Trigger' menu, select the "When the computer starts" radio button.

Click Next.

e. In the 'Action menu, select the "Start a program" radio button. Click next.

f. In "Program/script" enter "a:\euca.bat" and click next.

g. Check "Open the Properties" dialog for this task when I click Finish. Click Finish.

h. In the Properties window, select these options: "Run whether user is logged in or not" and "Run with highest privileges".

i. Add current Admin password to ensure the script will run on boot.

2. Enable Remote Desktop Protocol (RDP):

a. Start > My Computer > Properties > Remote Settings.

b. Check the 'Allow connections from computers running any version of Remote Desktop' checkbox.

c. Ensure firewall is **disabled**. (If firewall is enabled, then ensure that RDP connections are permitted through firewall.)

Windows 7:

1. Boot password set:

a. Start > Administrative Tools > Task Scheduler.

b. In the 'Actions' window (right pane) click on 'Create Basic Task.'

The **Create Basic Task** wizard opens.

c. In the 'Name' field enter: "Random Boot Password." Click Next.

d. In the 'Trigger' menu, select the "When the computer starts" radio button. Click Next.

e. In the 'Action menu, select the "Start a program" radio button. Click next.

f. In "Program/script" enter "a:\euca.bat" Click next.

g. In the Properties window, select these options: "Run whether user is logged in or not" and "Run with highest privileges".

h. Add current Admin password to ensure the script will run on boot.

2. Enable Remote Desktop Protocol (RDP):

a. Start > My Computer > Properties > Remote Settings.

b. Check the 'Allow connections from computers running any version of Remote Desktop' checkbox.

c. Ensure firewall is **disabled**. (If firewall is enabled, then ensure that RDP connections are permitted through firewall.)

Step 5: Shut Down the Windows Image

The next step is to shut down the Windows image you have created in the above steps. The easiest way to shutdown your windows image is via libvirt using the **`virsh shutdown`** command, as shown:

```
[root@node1]# cd / ; virsh shutdown <domainName/ID>
```

Step 6: Enable Boot from Disk

Now that your Windows image is shutdown, you must edit the **`libvirt.xml`** file so the Windows image can boot from disk, as follows:

1. Go to **`//[home directory]/[local directory]/libvirt.xml`**

2. Remove the lines of XML code specifically directing Xen/KVM to boot the Windows image from CD-ROM as shown in this example:

```
   ...
      <boot dev='cdrom'/>
...
   <disk type='file' device='cdrom'>
      <source
   file='/root/en_windows_server_2008_datacenter_enterprise_standard_
   x64_dvd_X14-26714.iso'/>
      <target dev='hdc'/>
      <readonly/>
   </disk>
```

The revised libvirt.xml file allows the Windows image to boot from hard disk by default.

3. Run the Windows image using the **`virsh create`** command to confirm that your Windows image now boots from disk, as shown:

```
[root@node1]# cd / ; virsh create <libvirt.xml>
```

At this point, your Windows VM image should be ready to use in Eucalyptus. For instructions on managing and running Windows VM images in Eucalyptus, see *the Eucalyptus EE User Guide.*

Appendix B: Installing and Configuring Hypervisors (Xen/KVM)

Eucalyptus deploys instances (i.e., virtual machines) on a hypervisor. Eucalyptus can use either Xen or KVM hypervisors. The correct hypervisor to use depends on the Linux distribution on which you are running Eucalyptus. For example, if running Eucalyptus on CentOS 5.5 the Xen hypervisor is recommended.

To interact with these hypervisors, Eucalyptus employs the *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 distributions have better support for Xen, some for KVM), and personal preference.

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 may 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 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).

B.1 Installing Xen on CentOS 5.5

To install Xen and Xen tools on a CentOS 5.5 systems, please ensure that “yum” is functional; and (if using RHEL) that you have set up the required entitlements for RHN access. Then install these packages:

```
[root@node1]# yum install xen kernel-xen libvirt
```

This will install Xen tools, the kernel with Xen support, and the libvirt abstraction layer. The next step is to configure the Xen daemon.

B.1.1 Configuring Xen

After installing Xen and the Xen kernel, you may need to edit your “grub bootloader” configuration file to load the correct kernel at boot time. Go to:

`/etc/grub.conf`

Look for the "kernel" line that contains "xen" and make sure the "default" parameter corresponds to the Xen kernel. For example:

```
default=1
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-164.11.1.el5)
       root (hd0,0)
       kernel /vmlinuz-2.6.18-164.11.1.el5 ro
root=/dev/VolGroup00/LogVol00
       initrd /initrd-2.6.18-164.11.1.el5.img
title CentOS (2.6.18-164.11.1.el5xen)
       root (hd0,0)
       kernel /xen.gz-2.6.18-164.11.1.el5
       module /vmlinuz-2.6.18-164.11.1.el5xen ro
root=/dev/VolGroup00/LogVol00
       module /initrd-2.6.18-164.11.1.el5xen.img
```

In this case we ensure the correct kernel will be loaded by setting the “default” parameter to "default=1" (not “default=0"), since the 2nd kernel in the list is the one containing Xen.

Next, make sure that your **`xend-config.sxp`** file is setup correctly. Go to:

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

These are the recommended settings:

```
(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 '')
```

The default settings that ship with CentOS 5.5 are generally appropriate except you must enable **`xend-http-server`** (as shown above) and restart the daemon.

B.1.2 Running a Test VM with Xen Tools

After installing Xen it is a good idea to run a test VM from the command line. (If you cannot run a VM on Xen *outside* Eucalyptus, you will not be able to run VMs on Xen

through Eucalyptus.) Running a Xen VM typically involves creating a configuration file and passing it to the `xm create` command in Xen tools.

B.2 Installing KVM on CentOS 5.5

To install KVM via RPM packages on Centos 5.5 "yum" must be enabled; and (if using RHEL) you must have set up the required entitlements for RHN access. In addition, make sure that "hardware virtualization support" is enabled in the BIOS of any node machine on which you plan to install KVM. Then install these packages:

```
[root@node1]# yum install kvm kmod-kvm kvm-qemu-img \
                          kvm-tools libvirt
[root@node1]# ln –s /usr/libexec/qemu-kvm/usr/bin/kvm
```

This installs the KVM hypervisor and the libvirt abstraction layer.

B.2.1 Running a Test VM on KVM

Running a KVM VM typically involves invoking `kvm` with many parameters on the command-line. For further information see the hypervisor-related documentation accompanying your specific Linux distribution.

B.3 Configuring libvirt

Eucalyptus interacts with both Xen and KVM hypervisors through libvirt, thus it is important to confirm that libvirt is properly configured. To do so, access the libvirt configuration file at:

`/etc/libvirt/libvirtd.conf`

Remove the comment character (#) from each of the following lines:

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

Note that if you change the `libvirtd.conf` file configuration, you must restart the `libvirtd` daemon.

B.3.1 Testing libvirt Configuration with virsh

First, ensure that `libvirtd` daemon is running, as follows:

```
[root@node1]# /etc/init.d/libvirtd start
```

Next, ensure the "eucalyptus" user can communicate with `libvirt` using the **`virsh list`** command:

```
[root@node1]# su –c "virsh list" eucalyptus
Id Name                 State
----------------------------------
 0 Domain-0
[root@node1]#
```

If **`virsh list`** succeeds (i.e., it returns a Domain-0) for user "eucalyptus," `libvirt` is configured properly and you're ready to continue.

Appendix C: Eucalyptus Admin-tools

C.1 User Management

Command	Arguments	Description
euca-add-user	USERNAME	Add new user USERNAME.
	-e EMAIL, -email =EMAIL	Email of user.
	-a, -admin	Mark user as admin.
euca-describe-user	[USERNAME…]	Describe the users specified by the list of USERNAMES.
euca-delete-user	USERNAME	Delete the user USERNAME.

C.2 Group Management

Command	Arguments	Description
euca-add-user-group	GROUPNAME	Create GROUPNAME.
euca-describe-user-groups	[GROUPS…]	Describe listed GROUPNAME(s).
euca-delete-user-group	GROUPNAME	Delete GROUPNAME
euca-add-group-membership	GROUPNAME	Add the user USERNAME to the group GROUPNAME
	-u USERNAME	Name of User
euca-remove-group-membership	GROUPNAME	Remove the user USERNAME from the group GROUPNAME
	-u USERNAME	

C.3 Permissions

Command	Arguments	Description
euca-grant-zone-permission	GROUP	Grant permission to the group GROUP to use the availability zone ZONENAME
	-z ZONENAME	Name of the availability zone
euca-revoke-zone-permission	GROUP	Revoke permission from the group GROUP for availability zone ZONENAME
	-z ZONENAME	Name of the availability zone

C.4 Example Usage of User/Group Tools

```
## NOTE: -a flag adds administrative privileges to a user!
[admin@clc] euca-add-user -a exampalyptus
RESPONSE true
## NOTE: Below the fields USER-CODE and USER-WEB can be used to obtain
credentials and send manual confirmation emails.
[admin@clc] euca-describe-users exampalyptus
USER     exampalyptus   n/a   ADMIN ENABLED
USER-GROUP  exampalyptus    default
USER-CERT   exampalyptus   None  None
USER-KEYS   exampalyptus   phbK3VvVMLXhr2Mivnok8bs4kYVzExb1JAcQYw
   FSX6ct0Mn1YFlYa38L9I1t2339GsmMdDsiQ
USER-CODE   exampalyptus
   VYfghcxvzI3vMW6RcExqgnn4rPenFZdJXQxvMouHIuFh84OHEiwTg2acXRT6SYf4BaLDf
QNCoTdlMUbB2Q
USER-WEB    exampalyptus
   zjRMYZj642YACc18kxpTAc1PRtXLs5ChXLDYIjQkBHOtVKiWRS0XfK0iAYFzkcVduSacT
raGqqFRS5eFqKQ
## NOTE: Create a group
[admin@clc] euca-add-user-group e.g.roup
RESPONSE true
[admin@clc] euca-describe-user-groups e.g.roup
GROUP e.g.roup
USERS e.g.roup
AUTH  e.g.roup
## NOTE: Now, add the user to the group
[admin@clc] euca-add-group-membership -u exampalyptus e.g.roup
RESPONSE true
[admin@clc] euca-describe-user-groups e.g.roup
GROUP e.g.roup
USERS e.g.roup  exampalyptus
AUTH  e.g.roup
## NOTE: Now, grant permission to the group for one of the registered
clusters
[admin@clc] euca-grant-zone-permission -z cluster00 e.g.roup
RESPONSE true
[admin@clc] euca-describe-user-groups e.g.roup
GROUP e.g.roup
USERS e.g.roup  exampalyptus
AUTH  e.g.roup  Availability Zone Permission:cluster00
```

C.5 Other

Command	Arguments	Description
`euca-describe-properties`	[PROPERTY…]	Describe the system properties specified by the list of PROPERTY names. With no argument all properties are shown.
	-v, -verbose Show property descriptions	
`euca-get-credentials`	FILENAME	Download the admin credentials for the **locally** hosted Eucalyptus installation
	-s, -source	Unzip (and remove zipfile) and output 'eucarc'. If FILENAME is specified the archive will be extracted to 'dirname FILENAME'. Otherwise, the archive is extracted to the current working directory.
`euca-modify-property`	[PROPERTY…]	Change the values of the specified system properties. Please use `euca-describe-properties` to obtain a list of these properties and their descriptions.
	-p KEY=VALUE	Modify key to be VALUE. Can be given multiple times.
`euca-convert-volumes`	Convert volumes from one backend implementation to another	
	-H SC_HOST, -HOST SC_HOST	Hostname or IP of the storage controller.
	-B PROVIDER, -backend PROVIDER	Storage backend provider to convert from.

Appendix D: Overview of Cloud Computing

Today's IT organizations face a difficult challenge: They must continue to deliver the productivity-enhancing technical innovation that gives the enterprise its competitive edge, while simultaneously streamlining operations in response to severe budgetary constraints. As a result, rather than replacing infrastructure with expensive purchases, IT professionals are pursuing innovative technologies, such as cloud computing, to improve IT efficiency and enhance the performance and longevity of existing IT resources.

D.1 What is Cloud Computing?

Cloud computing is the delivery of computer resources through a Web service interface (e.g., SOAP or REST) on an as-needed basis. The term "cloud" refers to the organization of the underlying physical infrastructure remaining opaque (not visible) to the end user. In other words, cloud computing gives a user access to computer resources (i.e. machines, storage, operating systems, application development environments, application programs) over a network (Internet or intranet) through Web services, while the actual physical location and organization of the equipment hosting these resources—be it in the next room or spread across the globe—is not necessarily known to the user. As such, these resources appear to the user as being "in the cloud."

Improved Efficiencies

All types of cloud computing—public clouds, private clouds, and hybrid clouds—share an intrinsic set of characteristics. These characteristics help define cloud computing as a uniquely efficient model for delivering computer resources:

- *Underlying organization of physical computing resources unknown to end user, or "in the cloud."*
- *Computing resources available as-needed.*
- *Web-services-based interfaces.*
- *Dynamic Scalability (a.k.a. "Elasticity).*
- *Automated resource provisioning.*
- *Self-service.*

D.2 The Efficiency Benefits of Cloud Computing

The above characteristics of cloud computing work synergistically to bring a high level of efficiency to IT operations and significant benefits to the enterprise, as follows:

Computing Resources Located "In the Cloud"

Regardless of the type of cloud, public, private, or hybrid, redesigning the IT organization to provide computing resources from a cloud can provide substantial efficiency benefits. In the case of public clouds, the IT organization simply facilitates access to the cloud for users through Web service protocols. Thus capital that may otherwise be invested into building an internal data center infrastructure goes instead to paying for rental of the public cloud. In the case of private and hybrid clouds, existing data center infrastructure is highly optimized by consolidation and virtualization. This translates into reduced power consumption and enhanced server longevity, and thus reduced infrastructure-related expense.

Computing Resources Available "as-needed"

With computing resources available on an "as-needed" basis, cloud-computing customers simply use the resources they need when they need them. In the case of public clouds, such as AWS (Amazon Web Services), there are no proprietary license agreements required. Customers simply pay via credit card on a per time-unit basis. When services are no longer required, they are simply discontinued and no further charges incurred. In the case of private clouds, as-needed means that resources are provided to users when they are needed, then efficiently redistributed to other users when need ceases.

Web-services-based Interfaces

Using standard Web-based interface enables users to easily access computing resources over common networks using universally accepted network communication protocols. These protocols are highly compatible with existing networking devices. Thus cloud systems are designed for efficient operation over both the public Internet and private enterprise intranets.

Dynamic Scalability (a.k.a. Elasticity)

In a cloud system, computing resources are highly scalable. This means that many resources can be committed to a single application or user. *Elasticity* refers to the ability to scale up (add resources) or scale down (decommission them) "on the fly," dynamically in response to changing application or user needs. Elasticity gives the enterprise the ability to repurpose resources immediately as demand (generated by users or auto-scaling applications) fluctuates.

Automated Resource Provisioning

Clouds perform the distribution of computer resources immediately and automatically without direct human intervention. Thus users can quickly access the resources they require, while IT staff is freed from many repetitive acquisition, configuration, and servicing tasks.

Self-service

In a cloud setup, the end user directly selects the computer resources desired through a self-service Web interface. This gives the user direct control over computer resource deployment and configuration, and it helps assure that the users needs are more closely met. Self-service also frees the user from waiting in long IT service queues, and can thus enhance the productivity of the enterprise workforce.

In addition, **SLAs** (service-level agreements) combined with auditing and metrics tools help assure that users receive the computing resources expected.

D.3 Cloud Types (public, private, and hybrid)

While all cloud types share the same fundamental characteristics, there are some important distinctions. Each cloud type addresses a different set of needs for an organization:

Public Clouds

Public clouds provide computing resources over the Internet to the general public on a pay-as-you-go basis. Computing resources are provided to customers through self-serve interfaces. Customers generally "rent" access to resources as needed, on a per time-unit basis (e.g., per hour), or in some cases through subscription. Payments are generally made online via credit card.

Public clouds address the needs of an organization requiring access to highly scalable computing resources on a temporary or periodic basis. The main advantage of the public cloud is the ability to acquire access to high-quality pools of resources immediately and with minimal capital investment. The public cloud can thus serve the needs of individuals, small companies, or startups with limited finances; or any organization with periodic spikes in demand that require access to large-scale computing resources.

The disadvantage of the public cloud is that expenses accrue over time, and for an organization with significant computing needs the public cloud can become expensive. In addition, though encryption protocols can be used both for communication and storage, the current virtual machine technology requires that code and data be made available un-encrypted when it is executed and processed respectively. That is, if the data is processed in a public cloud, it must be "in the clear" when it is processed, as must the code that processes it, even if both are transmitted and stored in encrypted form. This requirement makes it impossible to create a "chain of trust" that does not involve the public cloud provider; hence public clouds are potentially inappropriate for particularly sensitive data and applications. Well-known public cloud providers include Amazon Web Services (AWS), Google Apps, Microsoft, IBM, Rackspace, and Salesforce.com.

Private Clouds

Private clouds provide computing resources to users within an enterprise over an intranet. Computing resources are distributed through an automated provisioning system accessed by users via a self-serve interface, just as they are in public clouds.

Private clouds address the needs of organizations requiring a more efficient IT infrastructure as well as a high level of security over sensitive data. In a private cloud, data center resources are consolidated through virtualization and highly optimized to reduce operational expense. Automated resource provisioning and user self-service let the IT workforce shift focus to planning and customization projects that support a more productive enterprise workforce. And because a private cloud operates within the security perimeter of the enterprise, it uses the physical and electronic security measures in place in the data center to ensure the security of code and data.

Private clouds are limited however by the resource capacity of the enterprise's data center, which might be an issue during large spikes in demand. In this case, a hybrid cloud that offers the ability to "spill out" or burst into one or more public clouds is a viable option.

In addition, it is important to note that due to the entrenched nature of many IT organizational practices, some resistance could be met by IT staff to the fundamental process changes required to successfully implement a private cloud

within an enterprise. While private clouds share many common characteristics with virtualized data centers, their scaling and self-service characteristics often require organizational process and policy changes where simply virtualizing data center resources does not.

Hybrid Clouds

Hybrid clouds combine elements of both private and public clouds. In a hybrid cloud, users inside an organization with a private cloud can breakthrough the boundaries of the enterprise firewall or "cloudburst" to access additional computing resources in the public cloud.

Hybrid clouds address the needs of organizations that require periodic access to highly scalable computing resources beyond the capacity of the enterprise's existing data center infrastructure. Organizations benefit from the enhanced efficiency and security of the private cloud, but are prepared for large and unexpected spikes in demand.

D.4 Cloud Service Styles (SaaS, PaaS, IaaS)

Computing resources accessed via any type of cloud are usually grouped into service "styles" according to the kind of resources provided—software, platform, or infrastructure. These styles correspond to different levels of programming and/or operational abstraction that applications deployed in each style must use. The three most common cloud service styles are generally referred to by the following acronyms:

- **SaaS** (Software as a Service): Provides network accessible access to software application programs.
- **PaaS** (Platform as a Service): Provides network accessible access to a programming or runtime environment with scalable compute and data structures embedded in it.
- **IaaS** (Infrastructure as a Service): Provides access to virtualized computer hardware resources, including machines, network resources, and storage.

Many public cloud companies tailor their market offerings to address specific needs of customers according to these service styles. As a SaaS example, Salesforce.com provides access to hosted business application software packages through network facing APIs. Google's App Engine (GAE) is a PaaS offering that allows users to upload Python or Java programs so that they can take advantage of Google's Big Table infrastructure through a set of hosted interfaces. Finally, Amazon.com's Amazon Web Services (AWS) is an example of IaaS style cloud computing, offering time-limited rental of virtualized machine collections via network accessible web services.

D.5 Private Clouds: The Benefits of Automated Self-service

Automated resource provisioning and user self-service allow users to access computer resources as needed through a simple Web-interface, without assistance from IT staff. Giving users direct control over resource provisioning helps ensure that users needs are more closely met. And it promotes smooth workflow by minimizing time spent by users waiting for help in long service queues.

For IT staff, automated resource provisioning and self-service means a reduction in time spent servicing user requests, as well as a degree of freedom from manual acquisition, setup, configuration, and service tasks. IT staff can focus instead on more innovative pursuits that further enhance efficiency and user productivity, including such things as data center capacity planning, enhancing automation, image management, maintaining archival provenance and automated legacy support, and providing users with customized Web interfaces and resource allocation options.

Automated resource provisioning and self-service deliver efficiency benefits through outsourcing responsibility for user service from IT staff directly to the end user. This increases the "concurrency "of the IT service system, which significantly improves its efficiency.

Another concept with roots in the e-commerce business model, concurrency refers to the simultaneous processing of multiple user requests. Increasing concurrency in the IT service system allows for multiple user requests to be processed immediately and simultaneously and thus reduces or eliminates long IT service queues. For example, if ten users each have a provisioning request that will take IT one-hour to complete, a user might wait ten hours before receiving help—then wait another hour for the task to be completed. With the increased concurrency of a self-service automated provisioning system, all ten users can be serviced immediately and simultaneously.

D.6 The Role of Open Source

Seeking more cost-effective options that provide greater adaptability and flexibility, IT professionals have gravitated towards open source. The term "open source" refers to software source code that is human-readable (as opposed to machine-readable) and thus open to use and change as desired by developers. Due to open source's unrestricted and highly malleable quality, Linux-based open source operating systems have become the platform of choice for many IT organizations, as well as a very active and innovative open source developer community.

About Linux

Many groups and companies provide and support one or more of the standard distributions of Linux, including Ubuntu, CentOS, RHEL, OpenSUSE, Debian, and Fedora. Most of these can be downloaded for free over the Internet. As a result of these efforts, Linux has become a stable, reliable, and cost-effective operating system platform and a viable alternative to more well-known proprietary operating systems.

Open Source Business Model

Open source solutions are generally available in both source code and binary forms on-line and without charge. Communities of users and developers contribute the labor necessary to maintain them. Often, however, a principle entity (a person or a company) coordinates and manages each specific open source system (the Linux kernel being a notable exception). Companies that create open source software generally give-away their core product, and instead generate income by offering paid support services and customized versions that extend the baseline capabilities of the core. This business model keeps the software open to the value-adding creativity and innovations of the open source developer community.

Benefits of Transparency and Extensibility

The transparency and extensibility of Linux-based open source solutions make them highly adaptable and flexible to the needs of IT organizations, as follows:

Transparency

Transparency refers to the ability to look into and fix elements of a system if necessary. Providing unrestricted human-readable code that can be modified by developers makes open source solutions highly transparent and thus highly adaptable and compatible with the broadest range of existing data center technologies.

Extensibility

Extensibility refers to extending the capabilities of a system. Open source systems are highly extensible and can be freely built-upon by developers as desired. This extensibility promotes experimentation and innovation—developers can build customized add-on features that extend the usefulness and value of a program. Open source's extensibility thus lets IT organizations develop efficiency-enhancing customizations that better serve the organization and users.

D.7 Benefits of the Eucalyptus Cloud

The Eucalyptus private cloud gives IT organizations the features so essential to improving the efficiency of an IT infrastructure, including the following:

- **Data center optimization.** Eucalyptus optimizes existing data center resources with consolidation through virtualization of all data center elements, including machines, storage and network. Eucalyptus is compatible with most widely used virtualization technologies, including Xen, KVM, and ESX hypervisors.

- **Automated self-service**. Eucalyptus automates computer resource provisioning by allowing users to access their own flexible configurations of machines, storage, and networking devices as needed through a convenient self-service Web interface.

- **Customizable Web-interface.** Eucalyptus uses universally accepted Web-based network communication protocols that allow users to access computing resources through a highly customizable Web-interface.

- **Scalable data center infrastructure.** Eucalyptus clouds are highly scalable, which enables an organization to efficiently scale-up or scale-down data center resources according to the needs of the enterprise.

- **Elastic resource configuration.** The elasticity of a Eucalyptus cloud allows users to flexibly reconfigure computing resources as requirements change. This helps the enterprise workforce remain adaptable to sudden changes in business needs.

- **Open source core**. Highly transparent and extensible, Eucalyptus' open source core architecture remains entirely open and available for value-adding customizations and innovations provided by the open source development community. The Eucalyptus open source software core is available for free download at www.eucalyptus.com.

- **Hybrid cloud capability.** Engineered to emulate Amazon Web Services (AWS), Eucalyptus interacts seamlessly with Amazon public cloud services, including EC2 and S3, with no software modification required. This allows IT organizations to quickly "cloudburst" into the public cloud space without purchasing additional data center hardware during very large spikes in enterprise resource demand. The vibrant eco system built around the Amazon AWS can be leveraged. For example, RightScale, CohesiveFT, Zmanda, rPath are just a few of the partners that deliver solutions for Amazon AWS that in turn work seamlessly with Eucalyptus

Glossary

ACPI (Advanced Configuration and Power Interface) – an open standard used to specify power management configuration for various hardware devices.

ami - Abbreviation for Amazon Machine Image. Used as a prefix in naming Amazon machine image files.

ATA (Advanced Technology Attachment) - Computer bus technology used primarily for transferring data to and from hard drives.

availability zones - An availability zone for Amazon denotes a large subset of their Cloud environment. Within Eucalyptus, we refine this definition to denote a subset of the cloud that shares a local area network. Each availability zone has its own cluster controller and storage controller. Properly credentialed administrators can view detailed information on availability zones with the following command:

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

bridge names - To properly configure Eucalyptus networking modes you must know the correct bridge name for your system. Bridge names for systems using Xen or KVM hypervisors include the following: Xen 3.2: `eth0;` Xen 3.0 or earlier: `xenbr0;` kvm: `br0.`

Bucket - A bucket is a storage container that stores objects. Objects are added or removed from a bucket via the PUT and GET commands. Any number of objects can be added to a bucket up to the maximum storage capacity of the bucket.

Bundling - Bundling prepares a VM image to be uploaded into the cloud. (Before you can run an instance you must first bundle the image (`euca-bundle-image`); then upload the bundled image (`euca-upload-bundle`). "Bundling" an image separates the image into multiple image parts and generates an XML manifest file containing metadata about the image. When you run an instance of the image (`euca-run-instances manifest.xml`), Eucalyptus uses the manifest file to reference image parts and kernel in assembling the virtual machine.

Cloudburst - Cloudburst refers to the capability for private cloud users to extend beyond the enterprise firewall and access computing resources in the public cloud. Eucalyptus' hybrid cloud capability lets users seamlessly cloudburst into the AWS public cloud should additional resources be required.

cluster - In Eucalyptus, a cluster is a physical collection of a number of resources within single network segment. Each cluster contains a Cluster Controller, Storage

Controller and a number of computer resources. An *availability zone* is a virtual abstraction of a cluster.

command-line tools (e.g., Euca2ools) **-** Text commands that let you control and manage VMs and other Eucalyptus cloud operations via a standard Terminal interface window.

credentials - Credentials refer to data values that are used to authenticate a user. Credentials can be provided in some cases by certificates and in other cases by keypairs.

Daemon - A daemon is a process that runs in the background to perform a specific function or system task. The CLC, CC, etc. are daemons that are part of the Eucalyptus Cloud environment.

ElasticFox - ElasticFox is an add-on for Firefox that lets you configure and manage both Amazon EC2 and Eucalyptus accounts through a graphical interface.

eki- - Abbreviation for Eucalyptus Kernel Image. Used as prefix in naming Eucalyptus kernel image files.

emi- - Abbreviation for Eucalyptus Machine Image. Used as prefix in naming Eucalyptus machine image files.

eri- - Abbreviation for Eucalyptus Ramdisk Image. Used as prefix in naming Eucalyptus ramdisk image files.

eucarc - The 'eucarc' file is a resource configuration file included with the credentials zip-file you download through the Eucalyptus Web interface. Before you can run Euca2ools against Eucalyptus you must first 'source' the eucarc file via a ssh-compatible shell (by entering `. eucarc` at the command line). By sourcing this file, a set of environment variables is defined to provide user-specific information to Euca2ools command line tools.

front end - The "front end" refers to the physical machine that hosts the Cloud Controller (CLC) and in most cases the Cluster Controller (CC) components. As such, the front end is the entry point into the cloud providing an interface through which underlying virtualized resources (servers, network, and storage) can be evaluated and managed. In this guide we use the terms "front end" and "front-end machine" interchangeably.

hypervisor - A hypervisor (e.g., Xen, KVM, ESX/ESXi) is a piece of software that enables multiple operating systems to run concurrently on a single host computer. Type 1 (aka native or bare-metal) hypervisors, including Xen, run directly on a host machine's hardware. KVM (Kernel-based Virtual Machine) is a variation on a Type 1 hypervisor that embeds in a platform's firmware. This way KVM transforms the

Linux kernel into a hypervisor. Type 2 hypervisors are software applications that allow multiple operating systems to run on top of conventional operating systems.

instance – An instance refers to an actively running virtual machine. Instances are deployed by instantiating virtual machine images. Multiple identical VM instances can be instantiated from a single virtual machine image.

instantiate (aka 'running instances') – Instantiate refers to the process of deploying a running virtual machine instance from an encrypted virtual machine image. To instantiate a virtual machine image in Eucalyptus use the following command:

```
euca-run-instances –k <private-key) –n <number> <emi-id>
```

kernel - The kernel is the central software component of a computer operating system. Kernels facilitate the functioning of application programs; and they bridge application processes with underlying data processing and management functions performed at the computer hardware level.

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

```
euca-add-keypair mykey >mykey.private
```

A pair of keys are created; one public key and one private key. The public key, "mykey" is stored in Eucalyptus. The private key "mykey.private" is stored within a file on a local machine. After creating a keypair, you must change access permissions to enable the private key in your local directory as follows:

```
chmod 0600 mykey.private
```

(Note: Use the file mykey.private when logging into VMs via the ssh command)

KVM (Kernel-based Virtual Machine) - KVM is a Linux kernel-based hypervisor that supports the instantiation of virtual machine instances and other virtualized computing resources.

libvirt - libvirt is a library, accessible via a C API. The library manages and supports the virtualization capabilities of Linux and other operating systems.

libvirt virsh - Command-line utility used for managing user domains via the libvirt library.

Linux - Linux refers to Unix-like operating systems based on the open source Linux kernel. Standard distributions of Linux include Ubuntu, CentOS, RHEL, OpenSUSE,

Debian and Fedora. Most Linux distributions can be downloaded for free over the Internet.

manifest file - The manifest file (`manifest.xml`) is a file generated when bundling images prior to uploading them to the cloud. The manifest file contains metadata used by Eucalyptus to reference associated image parts and kernel when instantiating a virtual machine.

networking mode - Networking modes, including SYSTEM, STATIC, MANAGED, and MANAGED-NOVLAN, offer the administrator varying degrees of control over VM network access and security. Each mode contains a unique set of configuration parameters and features. System mode is the default mode. Managed mode offers the greatest number of network configurable features.

node - A node is a single physical machine on which any number of VM instances can be hosted and managed. In Eucalyptus, the Node Controller (NC) executes on each node designated for hosting VM instances. The NC manages all VM operations on a particular node via query and control requests from the Eucalyptus Cluster Controller.

open source - open source refers to software source code that is human-readable (as opposed to machine-readable) and thus open to use and change as desired by the general public. Due to open source's unrestricted and highly malleable quality, Linux-based open source operating systems have become the platform of choice for many IT organizations, as well as a very active and innovative open source developer community.

persistent data – persistent data refers to data that remains in an unchanged state regardless of the change in state of the software or processes that created that data.

PEM (Privacy Enhanced Mail) – PEM is an ASCII Base64 encryption format frequently used for creating standard X.509 digital certificates and public keys.

ramdisk – ramdisk contains a set of drivers loaded temporarily into memory that instructs the kernel to launch appropriate system files when instantiating a virtual machine.

reservation – In Eucalyptus, a reservation refers to an ID tag that is created each time one or more VM instances are instantiated using the `euca-run-instances` command. You can view reservation ID tags associated with VM instances by querying the system with the `euca-describe-instances` command.

REST – REST (Representational State Transfer) is an HTTP-based query method used by Web services that uses encoded representational messages to conduct client-server (request and response) interactions. Rest/Query APIs use common directory-like URLs containing these encoded messages to prescribe specific

service requests and subsequent responses returned in any number of formats, including HTML, XML, media files, etc.

root directory– In Unix-based operating systems that use a hierarchical file system, root is the directory containing all other directories within a file system. The root directory in Unix-based systems is represented with a forward slash (/).

RPM – RPM is a software package file format used by Linux distributions. Initially developed by Red Hat for Red Hat Linux (RHEL), RPM is now used by many Linux-based open source distributions, including Centos, OpenSUSE, Fedora, and others.

security group – A security group is a set of VM instances that allow an administrator to control access to and usage of the associated VMs.

SLA (Service Level Agreement) – An SLA is an agreement that formally defines a measurable degree of service a customer can expect from a service provider. SLAs are frequently used to specify minimum and/or target levels of availability, performance, operation, and other service-related attributes, including accounting and billing.

SOAP (Simple Object Access Protocol) – SOAP is a structured messaging protocol used by Web services to exchange data over computer networks. SOAP messages generally rely on XML and HTTP protocols for message specification, transmission, and negotiation.

source code – 'source code' refers to the original 'human-readable' computer program written by a programmer in any programming language. To execute a program its source code must be compiled and translated into 'machine-readable' code understood by computers.

sourcing a file (`. eucarc`) - When a file is 'sourced' (by typing either **`source filename`** or **`. filename`** at the command line), the lines of code in the file are executed as if they were entered at the command line. Sourcing the 'eucarc' file (Eucalyptus resource configuration file) establishes necessary environmental variables and validates required user authentications for your Eucalyptus cloud.

SSH (TCP, port 22)– The Unix/Linux shell is used to make a secure connection between two network devices. By default, SSH utilizes TCP port 22.

snapshot – A snapshot is a copy of the total existing state of a volume. Eucalyptus lets you create instantaneous snapshots of a volume (**`euca-create-snapshot <vol-xxxxxxxx>`**). You can then create a new volume (with identical file system and/or data) from that snapshot (**`euca-create-volume --snapshot <snapshotId> -z <zone>`**).

YUM (Yellowdog Updater, Modified) – YUM is a metapackage management utility that computes dependencies and implements processes that install and maintain RPM based Linux distributions.

Ubuntu Enterprise Cloud – The Ubuntu Enterprise Cloud (UEC) is an open source cloud computing platform included with the Ubuntu Linux distributions. The Eucalyptus open source software core is the key component of UEC.

virtual machine image – A virtual machine image provides the root file system for a virtual machine instance. Machine images can be created from a variety of Linux operating systems, including Ubuntu, Centos, RHEL, OpenSUSE, Debian, and Fedora. You can create your own images or use one of the pre-made images available from the Eucalyptus Web interface or a variety of other sources.

virtual network – A virtual network is an abstraction of a computer network that combines hardware and software network resources and network functionality into a single, software-based administrative entity.

virtualization – Virtualization is a general term that refers to the abstraction of computer resources in software (or a combination of software and hardware).

virtual machine (VM) – A virtual machine is an abstraction of computer hardware within software. As such, virtual machines execute programs as if they were actual physical machines.

Web service – A Web service is a software system that supports automated machine-to-machine interaction over a network. Web services use interfaces described in Web Services Description Language (WSDL). Other systems interact with Web services using SOAP or REST messages via HTTP with XML and other web-related standards.

Please send comments, corrections, and questions regarding this Eucalyptus EE Administrator's Guide to documentation@eucalyptus.com. Thank you