3.0.2 Installation Guide

2012-05-01 Eucalyptus Systems

Contents

Welcome

Welcome to the Eucalyptus Installation Guide. This guide will help you understand, plan for, and install Eucalyptus. If you follow the recommendations and instructions in this guide, you will have a working version of Eucalyptus customized for your specific needs and requirements.

How to Read this Guide

We recommend that you read this guide in the order presented. There are no shortcuts for installing a customized installation of Eucalyptus. You have to understand what Eucalyptus is, what the installation requirements are, what your network configuration and restrictions are, and what Eucalyptus components and features are available based on your needs and requirements.

How do I?	Relevant topic
Understand what Eucalyptus is and does	*Introduction to Eucalyptus*
Decide how the installation will be done on your system	*Planning Your Installation*
Install Eucalyptus dependencies	*Configuring Dependencies*
Install Eucalyptus	*Installing Eucalyptus*
Configure Eucalyptus for your system	*Configuring Eucalyptus*
Start Eucalyptus and verify the installation	*Starting Eucalyptus*

Important: If you are upgrading from a previous version of Eucalyptus, see *Appendix: Upgrading Eucalyptus*.

How to Read this Guide

We recommend that you read this guide in the order presented. There are no shortcuts for installing a customized installation of Eucalyptus. You have to understand what Eucalyptus is, what the installation requirements are, what your network configuration and restrictions are, and what Eucalyptus components and features are available based on your needs and requirements.

How do I?	Relevent topic
Understand what Eucalyptus is and does	*Introduction to Eucalyptus*
Decide how the installation will be done on your system	*Planning Your Installation*
Configure Eucalyptus dependencies	*Configuring Dependencies*
Install Eucalyptus packages	*Installing Eucalyptus*
Configure Eucalyptus for your system	*Configuring Eucalyptus*
Start Eucalyptus	*Starting Eucalyptus*
Register Eucalyptus components	*Registering Eucalyptus*
Configure Eucalyptus runtime environment	*Configuring the Runtime Environment*
Find out more information about Eucalyptus	*Finding More Information*

Introduction to Eucalyptus

Eucalyptus is a Linux-based software architecture that implements scalable private and hybrid clouds within your existing IT infrastructure. Eucalyptus allows you to provision your own collections of resources (hardware, storage, and network) using a self-service interface on an as-needed basis.

You deploy a Eucalyptus cloud across your enterprise's on-premise data center. Users access Eucalyptus over your enterprise's intranet. This allows sensitive data to remain secure from external intrusion behind the enterprise firewall.

You can install Eucalyptus on the following Linux distributions:

- CentOS 5.6 and above
- Red Hat Enterprise Linux 5.6 and above
- Red Hat Enterprise Linux 6
- Ubuntu 10.04 LTS

Eucalyptus Overview

Eucalyptus was designed to be easy to install and as non-intrusive as possible. The software framework is modular, with industry-standard, language-agnostic communication. Eucalyptus provides 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). Also, Eucalyptus offers API compatability with Amazon's EC2, S3, and IAM services. This offers you the capability of a hybrid cloud.

Eucalyptus Components

Eucalyptus is comprised of six components: Cloud Controller, Walrus, Cluster Controller (CC), Storage Controller (SC), Node Controller (NC) and an optional VMware Broker. Other than the VMware Broker, each component is a stand-alone web service. This architecture allows Eucalyptus both to expose each web service as a well-defined, language-agnostic API, and to support existing web service standards for secure communication between its components.

A detailed description of each Eucalyptus Component follows.

Cloud Controller

The Cloud Controller (CLC) is the entry-point into the cloud for administrators, developers, project managers, and end-users. The CLC queries other components for information about resources, makes high-level scheduling decisions, and makes requests to the Cluster Controllers (CCs). As the interface to the management platform, the CLC is responsible for exposing and managing the underlying virtualized resources (servers, network, and storage). You can access the CLC through command line tools that are compatible with Amazon's Elastic Compute Cloud (EC2) and through a web-based Dashboard.

Walrus

Walrus allows users to store persistent data, organized as buckets and objects. You can use Walrus to create, delete, and list buckets, or to put, get, and delete objects, or to set access control policies. Walrus is interface compatible with Amazon's Simple Storage Service (S3), providing a mechanism for storing and accessing virtual machine images and user data. Walrus can be accessed by end-users, whether the user is running a client from outside the cloud or from a virtual machine instance running inside the cloud.

Cluster Controller

The Cluster Controller (CC) generally executes on a machine that has network connectivity to both the machines running the Node Controllers (NCs) and to the machine running the CLC. CCs gather information about a set of node machines and schedules virtual machine (VM) execution on specific node controllers. The CC also manages the virtual machine networks. All Node Controllers associated with a single CC must be in the same subnet.

Storage Controller

The Storage Controller (SC) provides functionality similar to the Amazon Elastic Block Store (Amazon EBS). The SC is capable of interfacing with various storage systems (NFS, iSCSI, SAN devices, etc.). Elastic block storage exports storage volumes that can be attached by a VM and mounted or accessed as a raw block device. EBS volumes persist past VM termination and are commonly used to store persistent data. An EBS volume cannot be shared between VMs and can only be accessed within the same availability zone in which the VM is running. Users can create snapshots from EBS volumes. Snapshots are stored in Walrus and made available across availability zones. Eucalyptus with SAN support lets you use your enterprise-grade SAN devices to host EBS storage within a Eucalyptus cloud.

Node Controller

The Node Controller (NC) executes on any machine that hosts VM instances. The NC controls VM activities, including the execution, inspection, and termination of VM instances. It also fetches and maintains a local cache of instance images, 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.

VMware Broker

VMware Broker (Broker) is an optional Eucalyptus component activated only in versions of Eucalyptus with VMware support. Broker enables Eucalyptus to deploy virtual machines (VMs) on VMware infrastructure elements. Broker mediates all interactions between the CC and VMware hypervisors (ESX/ESXi) either directly or through VMware vCenter.

System Requirements

To install Eucalyptus, your system must meet the following baseline requirements.

Note: The specific requirements of your Eucalyptus deployment, including the number of physical machines, structure of the physical network, storage requirements, and access to software are ultimately determined by the features you choose for your cloud and the availability of infrastructure required to support those features.

Compute Requirements

- Physical Machines: All Eucalyptus components must be installed on physical machines, not virtual machines.
- Central Processing Units (CPUs): We recommend that each machine in your Eucalyptus cloud contain either an Intel or AMD processor with a minimum of two, 2GHz cores.
- Operating Systems: Eucalyptus supports the following Linux distributions: CentOS 5, RHEL 5, RHEL 6, and Ubuntu 10.04 LTS.
- Machine Clocks: Each Eucalyptus component machine and any client machine clocks must be synchronized (for example, using NTP). These clocks must be synchronized all the time, not just at installation.
- Hypervisor: CentOS 5 and RHEL 5 installations must have Xen installed and configured on NC host machines. RHEL 6 and Ubuntu 10.04 LTS installations must have KVM installed and configured on NC host machines. VMware-based installations do not include NCs, but must have a VMware hypervisor pool installed and configured.
- Machine Access: Verify that all machines in your network allow SSH login, and that root or sudo access is available on each of them.

Storage and Memory Requirements

- Each machine in your network needs a minimum of 30 GB of storage.
- We recommend at least 100GB for Walrus and SC hosts running Linux VMs. We recommend at least 250GB for Walrus and SC hosts running Windows VMs.
- We recommend a range of 50-100GB per NC host running Linux VMs, and at least 250GB per NC host for running Windows VMs. Note that larger available disk space enables greater number of VMs.
- Each machine in your network needs a minimum of 4 GB RAM. However, we recommend more RAM for improved caching.

Network Requirements

- All NCs must have access to a minimum of 1Gb Ethernet network connectivity.
- All Eucalyptus components must have at least one Network Interface Card (NIC) for a base-line deployment. For better network isolation and scale, the CC should have two NICS (one facing the CLC/user network and one facing the NC/VM network). For HA configurations that include network failure resilience, each machine should have one extra NIC for each functional NIC (they will be bonded and connected to separate physical network hardware components).
- Some configurations require that machines hosting a CC have two network interfaces, each with a minimum of 1Gb Ethernet.
- Depending on the feature set that is to be deployed, the network ports connecting the Ethernet interfaces may need to allow VLAN trunking.
- In order to enable all of the networking features, Eucalyptus requires that you make available two sets of IP addresses. The first range is private, to be used only within the Eucalyptus system itself. The second range is public, to be routable to and from end-users and VM instances. Both sets must be unique to Eucalyptus, not in use by other components or applications within your network.
- The network interconnecting physical servers hosting Eucalyptus components must support UDP multicast for IP address 228.7.7.3. Note that UDP multicast is not used over the network that interconnects the CC to the NCs.

Once you are satisfied that your systems requirements are met, you are ready to plan your Eucalyptus installation.

Planning Your Installation

In order to get the most out of a Eucalyptus deployment, we recommend that you create a plan that provides a complete set of features, performance, scaling, and resilience characteristics you want in your deployment.

Attention: If you are upgrading from an existing Eucalyptus release, see *Appendix: Upgrading Eucalyptus*.

To successfully plan for your Eucalyptus installation, you must determine two things:

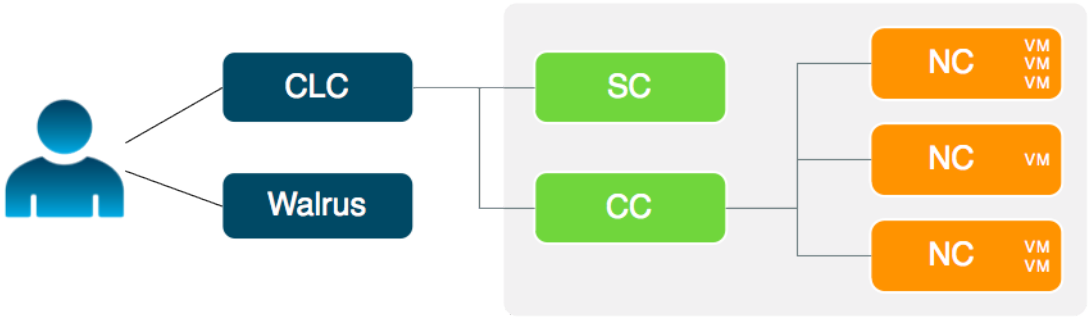
- **The infrastructure you plan to install Eucalyptus on:** Think about the application workload performance and resource utilization tuning. Think about how many machines you want on your system.
- **The amount of control you plan to give Eucalyptus on your network:** Use your existing architecture and policies to determine the Eucalyptus networking features you want to enable: elastic IPs, security groups, DHCP server, and Layer 2 VM isolation.

This section describes how to evaluate each tradeoff to determine the best choice to make, and how to verify that the resource environment can support the features that are enabled as a consequence of making a choice.

By the end of this section, you should be able to specify how you will deploy Eucalyptus in your environment, any tradeoffs between feature set and flexibility, and where your deployment will integrate with existing infrastructure systems.

Understanding the Eucalyptus Architecture

The following image depicts the logical relationship between Eucalyptus components in a generalized deployment.



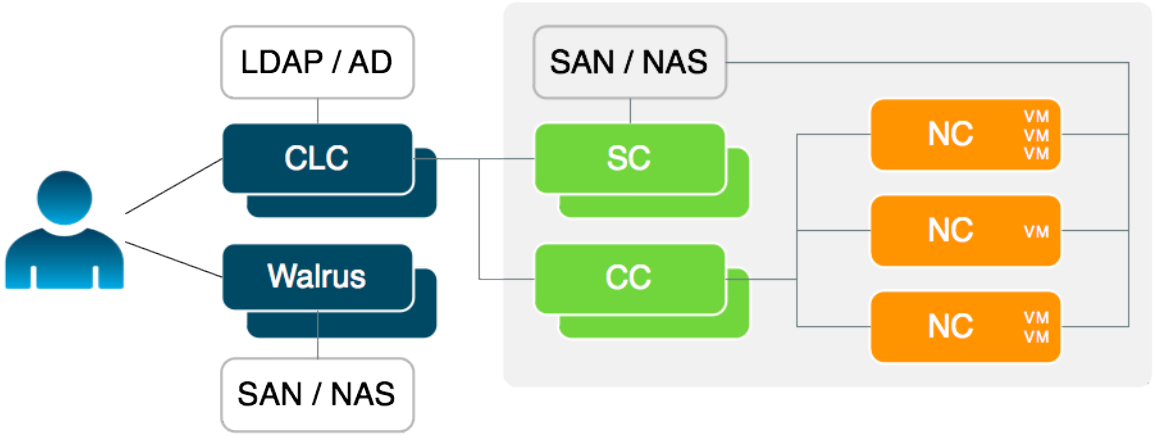
The cloud components, Cloud Controller (CLC) and Walrus, communicate with cluster components, the Cluster Controllers (CCs) and Storage Controllers (SCs). The CCs and SCs, in turn, communicate with the node controllers (NCs). The networks between machines hosting these components must be able to allow TCP connections between them.

However, if the CCs are on separate network interfaces (one for the network on which the cloud components are hosted and another for the network that NCs use) the CCs will act as software routers between these networks. So each cluster can use an internal private network for its NCs and the CCs will route traffic from that network to a network shared by the cloud components.

Virtual machines (VMs) run on the machines that host NCs. You can use the CCs as software routers for traffic between clients outside Eucalyptus and VMs. Or the VMs can use the routing framework already in place without CC software routers. However, depending on the layer-2 isolation characteristics of your existing network, you might not be able to implement all of the security features supported by Eucalyptus.

Eucalyptus HA

If you configure Eucalyptus for high availability (HA), the you will duplicate the cloud and cluster components. One component acts as the primary one unless there is a failure. In the event of a failure, the secondary component becomes the primary component.



Eucalyptus HA uses a service called Arbitrator that monitors connectivity between a user and a user-facing component (CLC, Walrus, and CC). An Arbitrator approximates reachability to a user. Each Arbitrator uses ICMP messages to periodically test reachability to an external entity (for example, a network gateway or border router) or to an external site (for example, google.com).

An Arbitrator is not required in HA. However, it is nice to have in order to test connectivity with a user.

If all Arbitrators fail to reach its monitored entity, Eucalyptus assumes there is a loss of connectivity between a user and the component. At that point a failover occurs. To allow for normal outages and maintenance, we recommend that you register more than one Arbitrator for each user-facing component.

Planning for Your Hardware

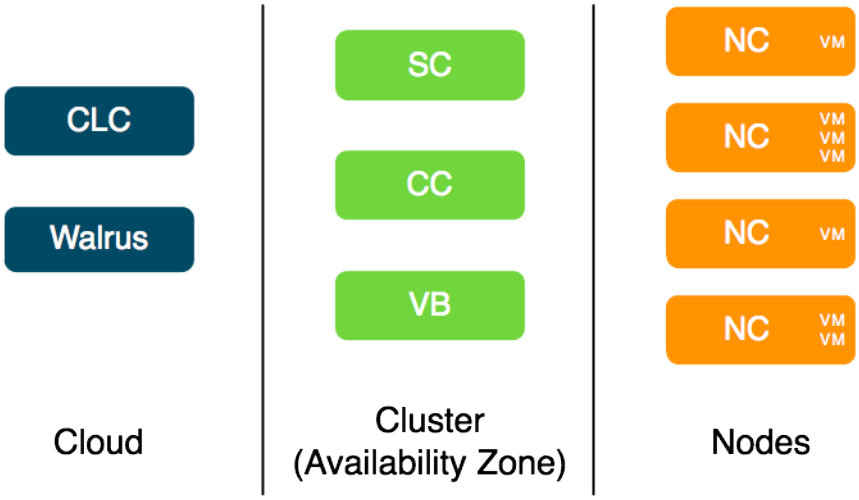
You can install Eucalyptus in various ways. You can install the CLC, Walrus, CC, and SC on one machine, and an NC on one or more machines. Or you can install each component on an independent physical server. This gives each component maximal local resource usage.

Often your decision about how to distribute Eucalyptus components across an installation must trade deployment simplicity for performance or high-availability. For example, placing all cloud and cluster components on a single machine can simplify administration because there is only one machine to monitor and control for the Eucalyptus control services. However, each of the components deploys as an independent web service. If these components must share a single cloud server, the physical resources that can be given to each service may become a performance bottleneck.

In general, the Eucalyptus components are designed to be run in any combination on the various physical servers in a data center. However, the majority of use cases can be satisfied by the below descriptions of deployment models.

Understanding Component Placement

A Eucalyptus deployment is a set of cloud services (CLC and Walrus) and one or more clusters, each of which contains a CC, an SC, an optional VMware Broker (located with the CC), and one or more NCs.



Cloud Components

The main decision for cloud components is whether to install the CLC and Walrus on the same server. If they are on the same server, they operate as separate web services within a single Java environment, and they use a fast-path for inter-service communication. If they are not on the same server, then they use SOAP and REST to work together.

However, when installed on the same server, the CLC and Walrus must share a common memory footprint, both managed by the Java memory manager. Walrus self-tunes its performance based on the memory pressure it perceives and runs faster with more memory. So, while separating the CLC and Walrus decreases the efficiency of the messaging between the two, it often increases the responsiveness of the overall Eucalyptus system when Walrus is given a large memory footprint.

Sometimes the key factor for cloud components is not performance, but server cost and data center configuration. If you only have one server available for the cloud, then you have to install the components on the same server.

The CLC and Walrus components are not designed to be separated by wide-area, common carrier networks. They use aggressive time-outs to maintain system responsiveness so separating them over a long-latency, lossy network link will not work.

The CLC and Walrus communicate with Eucalyptus clients independently. End-users typically interact with Eucalyptus through a client interface. They can use either our provided euca2ools Linux command line client tools, or the Eucalyptus AWS-compatible API, or a third-party client that is compatible with Eucalyptus. In all cases, the end-user client must be able to send messages via TCP/IP to the machine on which the CLC is deployed.

In addition, the CLC must have TCP/IP connectivity to all other Eucalyptus components except for node controllers (NCs), which may reside on their own private networks. In addition, NC servers must be able to send messages to the Walrus server because images are downloaded by the NC using the Walrus URL. That is, the CLC does not need to be able to route network traffic directly to the NCs but Walrus does for the purposes of image delivery.

Cluster Components

The Eucalyptus components deployed in the cluster level of a Eucalyptus deployment are the Cluster Controller (CC), Storage Controller (SC), and VMware Broker.

Tip: The VMware Broker is available by subscription only. You do not need the VMware Broker unless you are using VMware hypervisor.

You can install all cluster components on a single machine, or you can distribute them on different machines. The choice of one or multiple machines is dictated by the demands of user workload in terms of external network utilization (CC) and EBS volume access (SC).

CC Placement

If you plan to use elastic IPs and security groups, the CC physical machine becomes a software IP gateway between VM instances and the public network. Because of this software routing function, the physical server on which the CC is deployed should have fast, dedicated network access to both the NC network, and the public network.

If you don't plan to use elastic IPs or security groups, the CC physical machine will not act as a software gateway. Network traffic will be limited to small control messages.

In all cases, place the CC on a machine that has TCP/IP connectivity to the Eucalyptus cloud machines and the NC machines in its cluster.

SC Placement

The machine on which the SC is deployed must always have TCP/IP connectivity to the CLC. If you use one of Eucalyptus' provided SAN integration drivers, the SC must also have TCP/IP connectivity to the chosen SAN device. In this case, the SC only sends control messages to the SAN.

If you do not configure a SAN, the SC requires only TCP/IP connectivity to the NCs in the cluster. The SC will use this TCP/IP connectivity to provide the NCs network access to the dynamic block volumes residing on the SC's storage. SC storage should consist of a fast, reliable disk pool (either local file-system or block-attached storage) so that the SC can create and maintain volumes for the NCs. The capacity of the disk pool should be sufficient to provide the NCs with enough space to accommodate all dynamic block volumes requests from end-users

Subscription only: VMware Broker Placement

The VMware Broker resides on the CC. If you are using more than one cluster, make sure that the VMware Broker is installed on the CC in the cluster that will be using VMware components (vCenter Server or ESX/ESXi).

Make sure that the VMware Broker is able to communicate with the various VMware components (vCenter Server or ESX/ESXi) in its cluster.

Node Components

The Node Controllers are the components that comprise the Eucalyptus back-end. All NCs must have network connectivity to whatever hosts their EBS volumes. This host is either a SAN or the SC.

Verifying Component Disk Space

Eucalyptus components need disk space for log files, databases, buckets, and instances. The following table details the needs of each component. Verify that the machines you plan to install the components on have adequate space.

We recommend that you choose a disk for each Walrus that is large enough to hold all objects and buckets you ever expect to have, including all images that will ever be registered to your system, plus any Amazon S3 application data. For consistent performance, we recommend that you use identical disks for the primary and secondary Walrus.

Tip: We recommend that you use LVM (Logical Volume Manager). Should you run out of disk space, LVM allows you to add disks and migrate the data.

Component	Directory	Minimum Size
CLC	`/var/lib/eucalyptus/db`	2GB
CLC logging	`/var/log/eucalyptus`	1GB
Walrus	`/var/lib/eucalyptus/bukkits`	50GB
Walrus logging	`/var/log/eucalyptus`	1GB
SC	`/var/lib/eucalyptus/volumes` (EBS storage) This disk space on the SC is only required if you are not using a SAN driver.	50GB
CC	`/var/lib/eucalyptus/CC`	2GB

Component	Directory	Minimum Size
CC logging	`/var/log/eucalyptus`	500MB
NC	`/var/lib/eucalyptus/instances`	100GB
NC logging	`/var/log/eucalyptus`	500MB

If necessary, create symbolic links to larger filesystems from the above locations. Make sure that the eucalyptus user owns the directories.

Planning Networking Modes

Eucalyptus overlays a virtual network on top of your existing network. In order to do this, Eucalyptus supports four different networking modes: Managed, Managed (No VLAN), System, and Static. Each mode is designed to allow you to choose an appropriate level of security and flexibility. The purpose of these modes is to direct Eucalyptus to use different network features to manage the virtual networks that connect VMs to each other and to clients external to Eucalyptus.

A Eucalyptus installation must be compatible with local site policies and configurations (e.g., firewall rules). Eucalyptus configuration and deployment interfaces allow a wide range of options for specifying how it should be deployed. However, choosing between these options implies tradeoffs.

Your choice of networking mode depends on the following considerations:

- Do you plan to support elastic IPs and security groups?
- Do you plan to provide your own network DHCP server?
- Do you plan to support Layer 2 VM isolation?

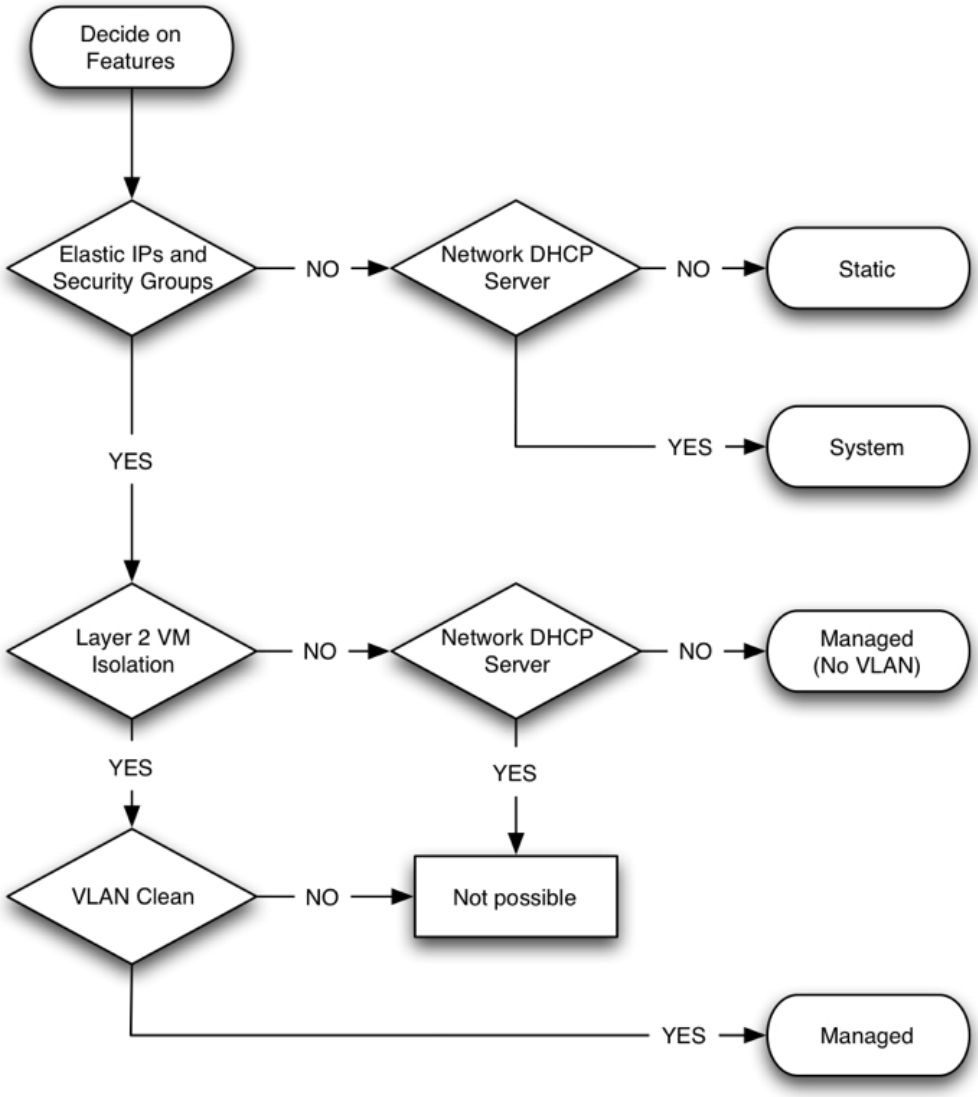
These networking features are described in the following table:

Feature	Description	Mode
Elastic IPs	Eucalyptus instances typically have two IPs associated with them: a private one and a public one. Private IPs are intended for internal communications between instances and are usually only routable within a Eucalyptus cloud. Public IPs are used for external access and are usually routable outside of Eucalyptus cloud. How these addresses are allocated and assigned to instances is determined by a networking mode. In System and Static modes, an instance is assigned only one IP address, which will be represented as both the private and public address assigned to the instance. Whether this address is routable outside of Eucalyptus is a property of the addresses that are set by the cloud administrator during Eucalyptus configuration. The distinction between public and private addresses becomes important in Managed and Managed (No VLAN) modes, which support elastic IPs. With elastic IPs the user gains control over a set of static IP addresses. Once allocated to the user, those same IPs can be dynamically associated to running instances, overriding pre-assigned public IPs. This allows users to run well-known services (for example, web sites) within the Eucalyptus cloud and to assign those services fixed IPs that do not change.	Managed Managed (No VLAN)
Security groups	Security groups are sets of networking rules that define the access rules for all VM instances associated with a group. For example, you can specify ingress rules, such as allowing ping (ICMP) or SSH (TCP, port 22) traffic to reach VMs in a specific security group. When you create a VM instance, unless otherwise specified at instance run-time, it is assigned to a default security group that denies incoming network traffic from all sources. Thus, to allow	Managed Managed (No VLAN)

Feature	Description	Mode
	login and usage of a new VM instance you must authorize network access to the default security group with the euca-authorize command.	
VM isolation	Although network traffic between VM instances belonging to a security group is always open, Eucalyptus can enforce isolation of network traffic between different security groups. This isolation is enforced using a VLAN tag per security group, thus, protecting VMs from possible eavesdropping by VM instances belonging to other security groups.	Managed
DHCP server	Eucalyptus assigns IP addresses to VMs in all modes except System. In System mode, you must allow a DHCP server outside of Eucalyptus to assign IPs to any VM that Eucalyptus starts.	Static Managed Managed (No VLAN)

If Eucalyptus can control and condition the networks its components use, your deployment will support the full set of API features. However, if Eucalyptus is confined to using an existing network, some of the API features might be disabled. So, understanding and choosing the right networking configuration is an important (and complex) step in deployment planning.

The following image shows which networking mode you should choose, depending on what networking features you want:



Each networking mode is detailed in the following sections.

Managed Mode

Managed mode offers the most features of the networking modes, but also carries with it the most potential constraints on the setup of the Eucalyptus administrator's network. In Managed mode, Eucalyptus manages the local network of VM instances and provides all networking features Eucalyptus currently supports, including VM network isolation, security groups, elastic IPs, and metadata service.

In Managed mode, you define a large network (usually private, unroutable) from which VM instances will draw their IP addresses. Eucalyptus maintains a DHCP server with static mappings for each VM instance that is created. When you create a new VM instance, you can specify the name of the network group to which that VM will belong. Eucalyptus then selects a subset of the entire range of IPs, to hand out to other VMs in the same network group.

You can also define a number of security groups, and use those groups to apply network ingress rules to any VM that runs within that network. In this way, Eucalyptus provides functionality similar to Amazon's security groups. In addition, the administrator can specify a pool of public IP addresses that users may allocate, then assign to VMs either at boot or dynamically at run-time. This capability is similar to Amazon's 'elastic IPs'. Eucalyptus administrators that require security groups, elastic IPs, and VM network isolation must use this mode.

Managed mode uses a Virtual LAN (VLAN) to enforce network isolation between instances. If your underlying physical network is also using a VLAN, there can be conflicts that prevent instances from being network accessible. So you have to determine if your network is VLAN clean (that is, if your VLANs are usable by Eucalyptus). To test if the network is VLAN clean, see VLAN Preparation.

Each VM receives two IP addresses: a public IP address and a private IP address. Eucalyptus maps public IP addresses to private IP addresses. Access control is managed through security groups.

Managed Mode Requirements

- There must be an available range of IP addresses for the virtual subnets. This range must not interfere with the physical network. Typically these IP addresses are selected from the private IP ranges: 192.168.x.x, 10.x.x.x, etc.
- The network must be VLAN clean, meaning that all switch ports that Eucalyptus components are connected to will allow and forward VLAN tagged packets.
- Any firewall running on the Cluster Controller must be 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).
- Any DHCP server on the subnet must be configured not to serve Eucalyptus instances.

Managed (No VLAN) Mode

In Managed (No VLAN) mode, Eucalyptus fully manages the local VM instance network and provides all of the networking features Eucalyptus currently supports, including security groups, elastic IPs, etc. However, it does not provide VM network isolation. Without VLAN isolation at the bridge level, it is possible in Managed (No VLAN) mode for a root user on one VM to snoop and/or interfere with the ethernet traffic of other VMs running on the same layer 2 network.

Tip: In Managed (No VLAN) mode, VM isolation is guaranteed by having different security groups on different subnets—this translates into Layer-3 only VM isolation.

Managed (No VLAN) Mode Requirements

- There must be an available range of IP addresses for the virtual subnets. This range must not interfere with the physical network. Typically these IP addresses are selected from the private IP ranges: 192.168.x.x, 10.x.x.x, etc.
- Any firewall running on the Cluster Controller must be 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.
- The CC must have a DHCP server daemon installed that is compatible with ISC DHCP Daemon version 3.0.X.

Managed (No VLAN) Mode Limitations

- No VM isolation.

System Mode

This is the simplest networking mode, but it 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 bridge. Then, VM instances can obtain an IP address using DHCP, the same way any machine using DHCP would obtain an address.

There is very little Eucalyptus configuration required to use System mode. Eucalyptus mostly stays out of the way in terms of VM networking. This mode requires a pre-configured DHCP server already active on the physical subnet. This server must be reachable by the machines hosting NC components. This mode is most useful for users who want to try out a simple Eucalyptus installation.

System Mode Requirements

- The physical Ethernet device on each NC that communicates with the CC must be bridged.
- A pre-existing DHCP server must be running and configured and reachable from the NCs.

System Mode Limitations

- No elastic IPs
- No security groups
- No VM isolation

Static Mode

Static mode is similar to System mode but offers you more control over instance IP address assignment. In Static mode, you configure 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 an instance, and attaches the instance's ethernet device to the physical ethernet through the bridge on the NCs (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.

In this mode, Eucalyptus manages VM IP address assignment by maintaining its own DHCP server with one static entry per VM. Static mode requires the Eucalyptus administrator to specify the network configuration each VM should receive from the Eucalyptus DHCP server running on the same physical server as the CC component.

Static Mode Requirements

- The Ethernet device on each NC that communicates with the CC must be bridged.
- There must be an available range of IP addresses for the virtual subnets. This range must not interfere with the physical network. Typically these IP addresses are selected from the private IP ranges: 192.168.x.x, 10.x.x.x, etc.
- Any DHCP server on the subnet must be configured not to serve Eucalyptus instances.

Static Mode Limitations

- No elastic IPs
- No security groups
- No VM isolation

Planning for Eucalyptus Features

Before you install Eucalyptus, we recommend that you think about the features you plan to implement with Eucalyptus. These features are detailed in the following sections.

Windows Host Support

Eucalyptus requires the following for Windows support:

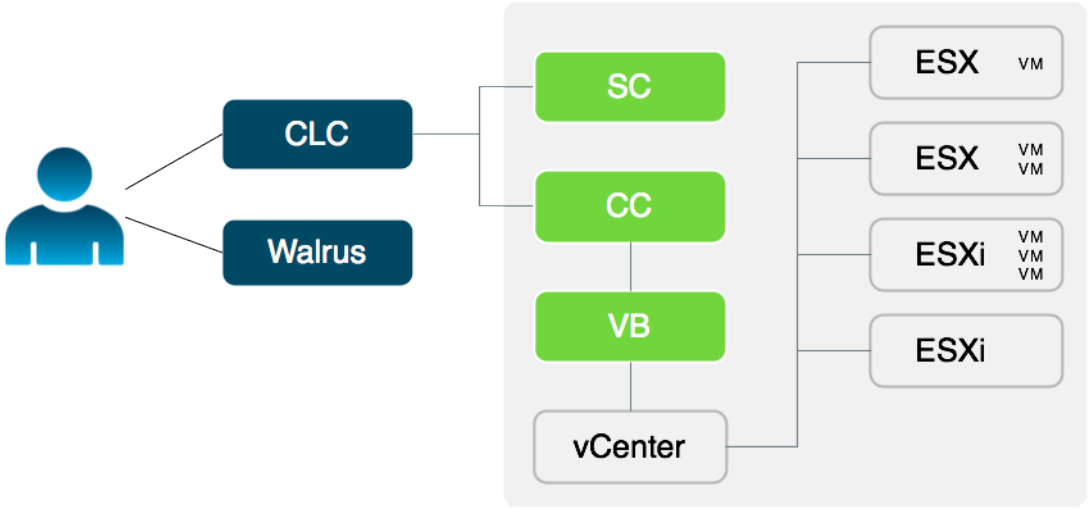
- A licensed installation copy (.iso image or CD/DVD disk) of a compatible Windows OS. Eucalyptus currently supports Windows virtual machines created from Windows Server 2003 R2 Enterprise (32/64 bit); Windows Server 2008 SP2, Datacenter (32/64 bit); Windows Server 2008 R2, Datacenter; and Windows 7 Professional.
- A VNC client such as RealVNC or Virtual Manager/Virtual Viewer (Centos/Xen) for initial installation. Subsequent Eucalyptus-hosted Windows instances will use RDP, but the initial installation requires VNC.

For additional Windows-related licensing information, see the following links:

- *http://technet.microsoft.com/en-us/library/dd979803.aspx*
- *http://technet.microsoft.com/en-us/library/dd878528.aspx*
- *http://technet.microsoft.com/en-us/library/dd772269.aspx*

VMware Support

Eucalyptus includes an optional subscription only component, the VMware Broker. The VMware Broker mediates all interaction between Eucalyptus and VMware infrastructure components (that is, ESX/ESXi, and vCenter).



Eucalyptus provides:

- Support for VMware vSphere infrastructure as the platform for deploying virtual machines
- Compatibility with VMware vSphere client, which can be used alongside Eucalyptus
- The ability to incorporate both VMware (ESX/ ESXi) and open-source (Xen and KVM) hypervisors within a single cloud infrastructure
- The ability to extend cloud-based features (for example, elastic IPs, security groups, Amazon S3, etc.) to a VMware infrastructure

The VMware Broker can run with either an administrative account or a minimally-privileged account on the VMware host.

VMware Support Prerequisites

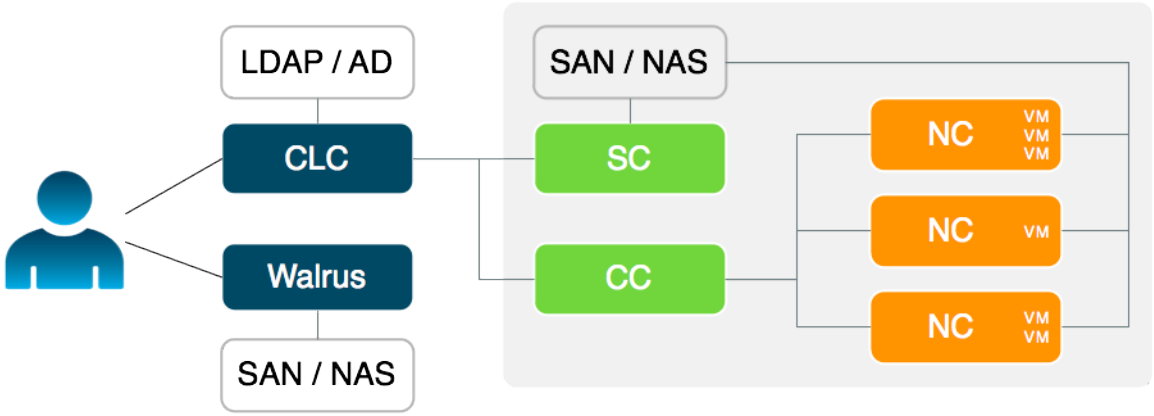
If you plan to use Eucalyptus with VMware, there are some additional prerequisites:

- You must install and configure the VMware infrastructure software (ESX and/or ESXi hypervisors with or without vCenter server).
- The CC server (that will also run the VMware Broker) must be able to route network traffic to and from the physical servers running VMware software on ports 443, 902, and 903. If there are internal firewalls present, these firewalls must be configured to open these ports so that the Eucalyptus cloud components can communicate with the VMware services and hypervisors.
- You must provide the VMware administrator account credentials to Eucalyptus when you configure VMware support, or an equivalent account with sufficient permissions must be created on VMware vCenter or ESX hosts.

For additional information on VMware support for Eucalyptus, contact Eucalyptus Systems, Inc.

SAN Support

Eucalyptus includes optional, subscription only support for integrating 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. Eucalyptus dynamically manages SAN storage without the need for the administrator to manually allocate and de-allocate storage, manage snapshots or set up data connections.



Eucalyptus with SAN support allows you to:

- 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 hypervisors

To use Eucalyptus with supported SAN storage, you must decide whether administrative access can be provided to Eucalyptus to control the SAN. If this is possible in your environment, Eucalyptus can automatically and dynamically manage SAN storage.

Currently, the Dell Equallogic series of SANs (PS 4000 and PS 6000) and NetApp Filer FAS 2000 and FAS 6000 series are supported. For Dell Equallogic, Eucalyptus requires SSH access to enable automatic provisioning. Eucalyptus will manage NetApp SANs via ONTAPI (version 7.3.3 and above).

SAN Support Prerequisites

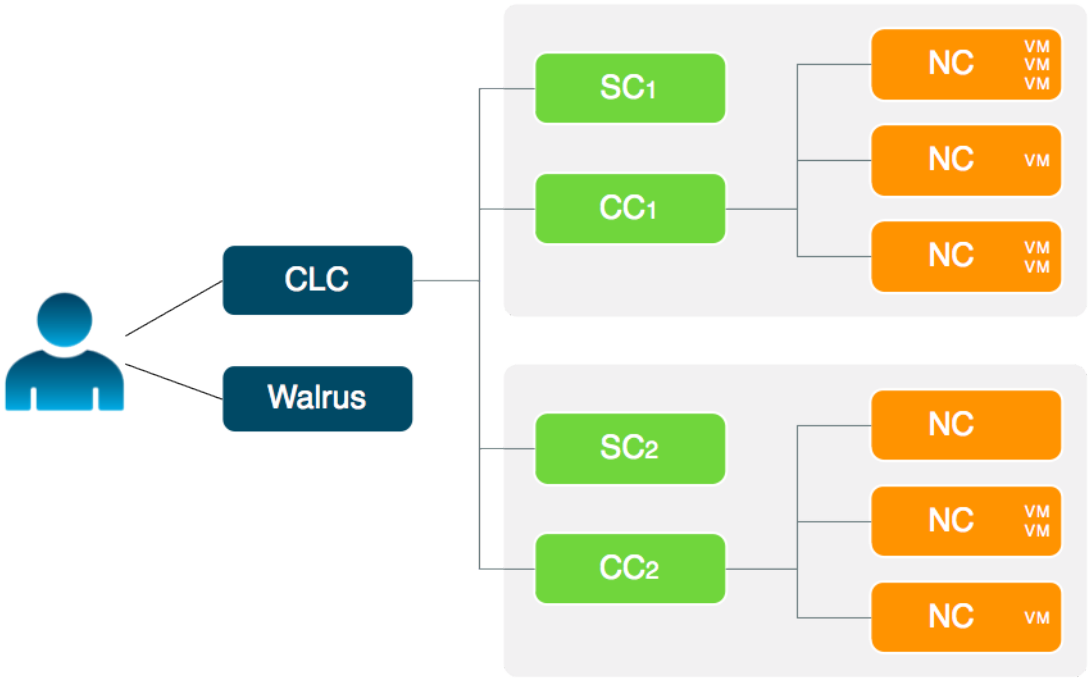
Eucalyptus supports the following SAN devices:

- Dell EqualLogic, PS4000 series and PS6000 series (For more information about Dell EqualLogic SANs, go to *http://www.dell.com*)
- NetApp, FAS2000 series and FAS6000 series (For more information about NetApp SANs, go to *http://www.netapp.com*
- Direct Attached Storage (JBOD)

For additional information on SAN support for Eucalyptus, contact Eucalyptus Systems, Inc.

Availability Zone Support

Eucalyptus offers the ability to create multiple availability zones. In Eucalyptus, an availability zone is a partition in which there is at least one available cluster.



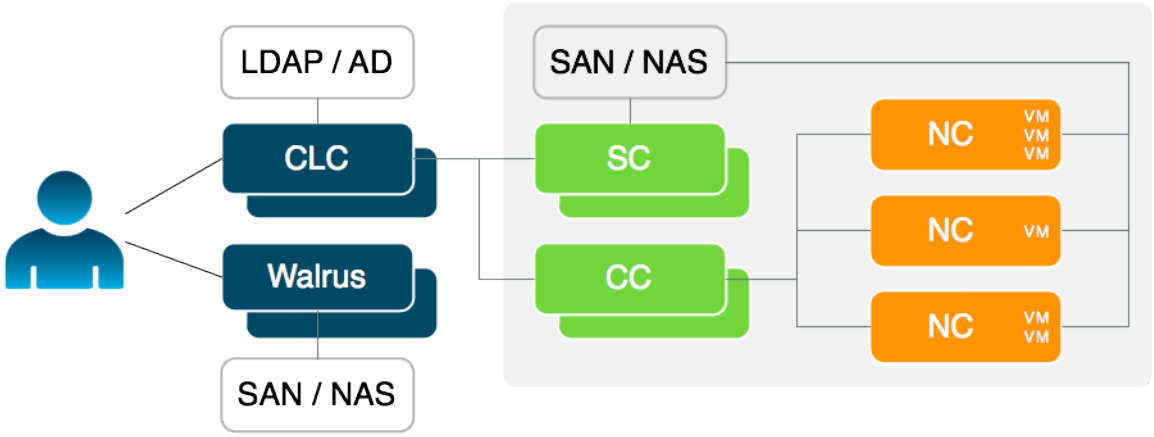
High Availability Support

Eucalyptus includes the ability to run redundant, hot swappable instances for the CLC, Walrus, CC, SC, and VMware Broker components. In a high availability (HA) configuration, a failure of any single component will not cause the system to halt. If your network configuration includes redundant networking hardware and routing paths, HA Eucalyptus can then tolerate a network component failure (e.g. the loss of a networking switch) without halting.

The deployment choices for HA Eucalyptus are similar to a regular Eucalyptus deployment, with the following additional considerations:

- You must host redundant Eucalyptus software components on separate hardware components in order to be able to tolerate a hardware failure. If, for example, you install redundant CLCs on the same machine and the machine crashes, both CLCs will become inoperable.
- The redundant components occur in pairs, one primary, the other secondary. These components must be able to communicate with each other through the network to which they are both attached while they are running. For example, both CLC components in an HA installation must be able to exchange messages. If you use a firewall to separate them, one will not detect a failure of the other and a hot failover will not occur. This ability for pairs of components to communicate is required for the CLC, Walrus, CC, SC, and the VMware Broker for HA to operate properly.

The following images shows a single cluster deployment with the component pairs at the cloud and cluster level. The NCs are not redundant.



Note that the same considerations for a regular Eucalyptus deployment with respect to networking mode and components placement apply to HA Eucalyptus in addition to the need for redundant component pairs to be able to communicate. Note also that the NC components are deployed redundantly in an HA Eucalyptus deployment. If a node running an NC fails, Eucalyptus will continue to be available for user requests. However, instances running on that specific node will be lost.

For HA: The installation and configuration sections will note instructions specific to HA deployment by the HA icon.

HA Requirements

HA Eucalyptus requires the same requirements as non-HA Eucalyptus. However, the infrastructure HA Eucalyptus will be deployed on must meet some additional requirements, listed in the following sections.

Redundant Physical Servers for Eucalyptus Components

Each cloud component (CLC and Walrus) and cluster component (CC, SC, and VMware Broker) in an HA deployment has a redundant hot backup. These redundant Eucalyptus components occur in pairs, and each member of a pair must be mapped to a separated physical server to ensure high availability.

Important: HA pairs must be able to connect to each other.

If the HA deployment is to be able to tolerate the failure of networking hardware, additional network interfaces are required for the physical servers that host Eucalyptus components. The physical servers hosting a CLC, Walrus, or CC and VMware Broker must each have three network interface cards (NICs). Each remaining physical server (except the NC components) requires two NICs.

DNS Round-Robin Support

The DNS entries for the externally visible IP addresses of the physical servers hosting CLC or Walrus components must be configured to change round-robin style in an HA deployment.

Storage Mirroring

HA Eucalyptus uses a kernel-level storage technology called DRBD for storage integrity. DRBD must be configured to mirror data operations between physical servers that host Walrus components. For more information about DRBD, go to *What is DRBD*.

Storage Controllers

For HA Storage Controllers, you must be using a supported SAN. Only use HA SCs with NetApp or Equallogic drivers, not with the iSCSI or JBOD SC driver.

HA Planning

High availability is the result of the combination of functionality provided by Eucalyptus and the environmental and operational support to maintain the systems proper operation. Eucalyptus provides functionality aimed at enabling highly available deployments:

1. **Detection of hardware and network faults which impact system availability:** Availability of the system is determined by its ability to properly service a user request at a given time. The system is available when there is at least a set of functioning services to perform the operations which result from a user request (i.e., system is distributed and operations require orchestration involving some, possibly all, services in the system).
2. **Deployment of redundant services to accommodate host failure:** A failure is the observed consequence of an underlying fault which compromises the systems function in some way (possibly compromising availability).
3. **Automated recovery from individual component failure:** Eucalyptus can take advantage of redundant host and network resources to accommodate singular failures while preserving the system's overall availability. As a result, the deployment of the system plays a large role in the level of availability that can be achieved.

To deliver services with high availability, Eucalyptus depends upon redundant hardware and network.

Considerations

A highly available deployment is able to mitigate the impact on system availability of faults from the following sources: = Machines hosting Eucalyptus services = Hardware faults on machines hosting Eucalyptus services can result in component services being unavailable for use by the system or users. The state of the hosting machine is monitored by the system and determines whether it can contribute to work done. In support of high availability, redundant component services can be configured. With redundant component services the system is able to isolate and mask the failure of one such service.

- **Inter-component networks:** Faults in the networks that connect the system's components to each other can prevent access to cloud resources and restrict the system's ability to process user requests. First, internal resources may become unavailable. For example, a single network outage could impact access to attached volumes or prevent access to running instances. Second, the coordination of services needed to process user requests may be impeded even if the service state is otherwise healthy.
- **User-facing network connections:** User-facing network faults can prevent access to an otherwise properly functioning system. The ability of a user to access the system is difficult to determine from the perspective of the system - can't look through the users eyes. Allowing for multiple inbound paths (i.e., multiple disjoint routes) decreases the possibility of an availability-impacting outage occurring w/in the scope of the environment within which Eucalyptus is deployed. (See also: registering arbitrators)

Recommendations

To ensure availability in the face of any single failure, we recommend the following deployment strategy:

- **Host/Service Redundancy:** Each component which is registered should have a complementary service registered on a redundant host. For example, the cloud and walrus services should be installed and registered on two hosts. Additionally, for example, each partition should have two cluster controllers and storage controllers (and vmware-brokers, if vmware is being used) configured. Each such complementary pair of services can suffer a single outage before system availability is compromised.
- **Inter-component Network Redundancy:** Each host of a component service should have redundant and disjoint network connections to other internal component services and supporting systems (e.g., SANs, vSphere). The recommended approach is to have two ethernet devices (each connected to a disjoint layer-2 network) on each host and bonding the devices. Such a configuration is also suggested on node controllers. Then, the outage of a either layer-2 network or ethernet device on a host does not impact service availability or access to cloud resources.
- **User-facing Network Redundancy:** The wide area (where users are) network connection should be redundant and disjoint. Each such path should have an independent arbitrator host whose liveness (as determined by ICMP echo) is used to approximate the users' ability to access the system. Redundant network connections from the local area network to the wide area network and user reachability approximation (arbitrator)
- **System Reachability Approximation:** The wide area (where users are) network connection(s) path should have an independent host (arbitrator) whose liveness (as determined by ICMP echo) can serve as a reasonable approximation

of users' ability to access the system. Ideally, the host “closest” to the user, but still within the domain of the deployment environment should be used (e.g., the border gateway of the hosting AS network). With such an arbitrator host in the network path between the user and the system, a failure by the user to reach an otherwise working service and allow the system to enable the complementary service (which should have a separate network route) restoring user access.

Preparing the Network

Decisions you make about the most appropriate deployment options imply different requirements that the underlying infrastructure must meet in order for Eucalyptus to deploy.

Prepare Internal Firewalls

Open ports 8443, 8773, 8774, and 8777 in any firewalls on your network to allow the various Eucalyptus components to communicate.

Verify TCP/IP Connectivity

Verify connectivity between the machines you’ll be installing Eucalyptus on. Some Linux distributions provide default TCP/IP firewalling rules that limit network access to machines. Disable these default firewall settings before you install Eucalyptus components to ensure that the components can communicate with one another.

Verify component connectivity by performing the following checks on the machines that will be running the listed Eucalyptus components.

1. Verify the connection from an end-user to the CLC on ports 8773 and 8443.
2. Verify the connection from an end-user to Walrus on port 8773.
3. Verify connection from the CLC to Walrus on ports 8773 and 8777.
4. Verify the connectiion from the CLC to the CC on port 8774.
5. Verify the connection from the CLC to the SC on ports 8773 and 8777.
6. Verify the connection from the CC to an NC on port 8775.
7. Verify the connection from an NC to Walrus on port 8773. Or, you can verify the connection from the CC to Walrus on port 8773, and from an NC to the CC on port 8776.
8. If you use VMware with Eucalyptus, verify the connection from the VMware Broker to VMware (ESX, VSphere).

Prepare VLAN

Tip: You only need to read this section if you are using Managed mode. If you aren’t using Managed mode, skip this section.

Managed networking mode requires that switches and routers be “VLAN clean.” This means that switches and routers must allow and forward VLAN tagged packets. If you plan to use the Managed networking mode, you can verify that the network is VLAN clean between machines running Eucalyptus components by performing the following test.

1. Choose two IP addresses from the subnet you plan to use with Eucalyptus, one VLAN tag from the range of VLANs that you plan to use with Eucalyptus, and the network interface that will connect your planned CC and NC servers. The examples in this section use the IP addresses 192.168.1.1 and 192.168.1.2, VLAN tag 10, and network interface eth3, respectively.
2. On the planned CC server, choose the interface on the local Ethernet and run:

```
vconfig add eth3 10
ifconfig eth3.10 192.168.1.1 up
```

3. On a planned NC server, choose the interface on the local network and run:

```
vconfig add eth3 10
ifconfig eth3.10 192.168.1.2 up
```

4. On the NC, ping the CC:

```
ping 192.168.1.1
```

5. On the CC, ping the NC:

```
ping 192.168.1.2
```

   - If this VLAN clean test fails, configure your switch to forward VLAN tagged packets. If it is a managed switch, see your switch's documentation to determine how to do this.
   - If the VLAN clean test passes, continue with the following steps to remove the test interfaces.

6. On the CC, remove the test interface by running:

```
vconfig rem eth3.10
```

7. On the planned NC, run:

```
vconfig rem eth3.10
```

Configuring Dependencies

Before you install Eucalyptus, make sure you have the following dependencies installed and configured.

Install Hypervisors

Eucalyptus deploys VM instances on a hypervisor. A hypervisor is a software abstraction of a physical hardware platform that enables multiple guest operating systems to run concurrently on a single physical machine. Eucalyptus is directly compatible with both Xen and KVM hypervisors. To interact with these hypervisors, Eucalyptus uses the libvirt virtualization API. Alternatively, you can configure Eucalyptus to use VMware virtualization technologies with which it communicates using web services.

While it is usually possible to use Xen and KVM with any of the supported distributions, the level of effort necessary, and the quality of the resulting platform for different distribution-hypervisor combinations varies. Before choosing an open source hypervisor and a Linux distribution, we recommend that you consider the level of support that the community reports for a specific combination.

Eucalyptus supports the Xen hypervisor for CentOS 5 and RHEL 5, and KVM for RHEL 6 and Ubuntu.

CentOS 5

Tip: Skip this section if you selected `Virtualization` as an option when you installed the kernel.

Before installing Eucalyptus, make sure that each NC server has a kernel with Xen support, and that this is the default kernel loaded at boot time.

To determine the current kernel, enter the following command on an NC server:

```
uname -r | grep xen | wc -l
```

If the command returns a zero, the Xen kernel is either not installed or not setup correctly. You must install a Xen kernel.

To install a Xen kernel:

1. Log in to a host you plan to use as an NC.
2. Confirm that YUM is working.
3. Install the Xen kernel package.

```
yum install kernel-xen
```

4. Repeat on each planned NC in your system.

RHEL 5

Tip: Skip this section if you selected `Virtualization` as an option when you installed the kernel.

Before installing Eucalyptus, make sure that each NC server has a kernel with Xen support, and that this is the default kernel loaded at boot time.

To determine the current kernel, enter the following command on an NC server:

```
uname -r | grep xen | wc -l
```

If the command returns a zero, the Xen kernel is either not installed or not setup correctly. You must install a Xen kernel.

To install a Xen kernel:

1. Log in to a host you plan to use as an NC.
2. Confirm that YUM is working.
3. Make sure that you have set up the required entitlements for RHN access.
4. Install the Xen kernel package.

```
yum install kernel-xen
```

5. Repeat on each planned NC in your system.

RHEL 6

When you install Eucalyptus from packages, KVM will be installed on all NCs. For more information about using KVM on RHEL 6, go to the *Virtualization* page in the Red Hat documentation.

Ubuntu 10.04 LTS

When you install Eucalyptus from packages, KVM will be installed on all NCs. For more information about installing and using KVM on Ubuntu, go to the *KVM* page in the Ubuntu documentation.

Configure Bridges

For Managed (No VLAN), Static, and System modes, you must configure a Linux ethernet bridge on all NC machines. This bridge connects your local ethernet adapter to the cluster network. Under normal operation, NCs will attach virtual machine instances to this bridge when the instances are booted.

The process for setting up a Linux ethernet bridge differs depending on Linux distribution, but the result is always the same.

If you use Xen, the distros typically set up a bridge for you, and you'll simply have to find its name. For Xen versions 3.0 or earlier the bridge name is typically `xenbr0`

If you use Xen 3.2 the bridge name is typically `eth0`

If you use KVM, or if you want to configure a bridge manually, the following sections describe how to set up a bridge on various distributions. In this section we show examples for configuring bridge devices that either obtain IP addresses using DHCP or statically.

Important: Before you configure a bridge, you must install the `bridge-utils` package.

CentOS 5

To configure a bridge in CentOS:

1. Install the `bridge-utils` package.

```
yum install bridge-utils
```

2. Create a new ethernet bridge configuration file `/etc/sysconfig/network-scripts/ifcfg-br0` and enter the following:

```
DEVICE=br0
BOOTPROTO=dhcp
ONBOOT=yes
TYPE=Bridge
```

3. Add your physical ethernet device to the bridge by editing your physical ethernet device configuration file (`/etc/sysconfig/network-scripts/ifcfg-eth0`).
   - If you are using DHCP, the configuration will look similar to the following example:

```
DEVICE=eth0
TYPE=Ethernet
BRIDGE=br0
```

   - If you are using a static IP address, the configuration will look similar to the following example:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=static
IPADDR=<static_IP_address>
NETMASK=<netmask>
GATEWAY=<gateway>
ONBOOT=yes
```

4. Enter the following command:

```
service network restart
```

RHEL 5

To configure a bridge in RHEL5.x, you need to create a file with bridge configuration (for example, ifcfg-brX) and modify the file for the physical interface (for example, ifcfg-ethX).

1. Install the `bridge-utils` package.

```
yum install bridge-utils
```

2. Go to the `/etc/sysconfig/network-scripts` directory:

```
cd /etc/sysconfig/network-scripts
```

3. Open the network script for the device you are adding to the bridge and add your bridge device to it. The edited file should look similar to the following:

```
DEVICE=eth0
# change the hardware address to match the hardware address your NIC uses
HWADDR=00:16:76:D6:C9:45
ONBOOT=yes
BRIDGE=br0
NM_CONTROLLED=no
```

4. Create a new network script in the `/etc/sysconfig/network-scripts` directory called `ifcfg-br0` or something similar. The br0 is the name of the bridge, but this can be anything as long as the name of the file is the same as the `DEVICE` parameter, and the name is specified correctly in the previously created physical interface configuration (ifcfg-ethX).
   - If you are using DHCP, the configuration will look similar to:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=dhcp
```

```
ONBOOT=yes
DELAY=0
```

- If you are using a static IP address, the configuration will look similar to:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=static
IPADDR=<static_IP_address>
NETMASK=<netmask>
GATEWAY=<gateway>
ONBOOT=yes
```

5. Enter the following command:

```
service network restart
```

RHEL 6

To configure a bridge in RHEL6.x, you need to create a file with bridge configuration (for example, ifcfg-brX) and modify the file for the physical interface (for example, ifcfg-ethX).

1. Install the `bridge-utils` package.

```
yum install bridge-utils
```

2. Go to the `/etc/sysconfig/network-scripts` directory:

```
cd /etc/sysconfig/network-scripts
```

3. Open the network script for the device you are adding to the bridge and add your bridge device to it. The edited file should look similar to the following:

```
DEVICE=eth0
# change the hardware address to match the hardware address your NIC uses
HWADDR=00:16:76:D6:C9:45
ONBOOT=yes
BRIDGE=br0
NM_CONTROLLED=no
```

4. Create a new network script in the `/etc/sysconfig/network-scripts` directory called `ifcfg-br0` or something similar. The br0 is the name of the bridge, but this can be anything as long as the name of the file is the same as the `DEVICE` parameter, and the name is specified correctly in the previously created physical interface configuration (ifcfg-ethX).

   - If you are using DHCP, the configuration will look similar to:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
DELAY=0
```

- If you are using a static IP address, the configuration will look similar to:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=static
IPADDR=<static_IP_address>
NETMASK=<netmask>
GATEWAY=<gateway>
ONBOOT=yes
```

5. Enter the following command:

```
service network restart
```

Ubuntu 10.04 LTS

To configure a bridge on Ubuntu Lucid:

1. Install the `bridge-utils` package.

```
apt-get install bridge-utils
```

2. Modify the `/etc/network/interfaces` file.

- If you are using DHCP, the configuration will look similar to the following example:

```
auto lo
iface lo inet loopback

auto br0
iface br0 inet dhcp
    bridge_ports eth0
```

- If you are using a static IP address, the configuration will look similar to the following example:

```
auto lo
iface lo inet loopback

auto br0
iface br0 inet static
    address <static_IP_address>
    network <network>
    netmask <netmask>
    broadcast <broadcast_IP_address>
    gateway <gateway>
    bridge_ports eth0
    bridge_stp off
```

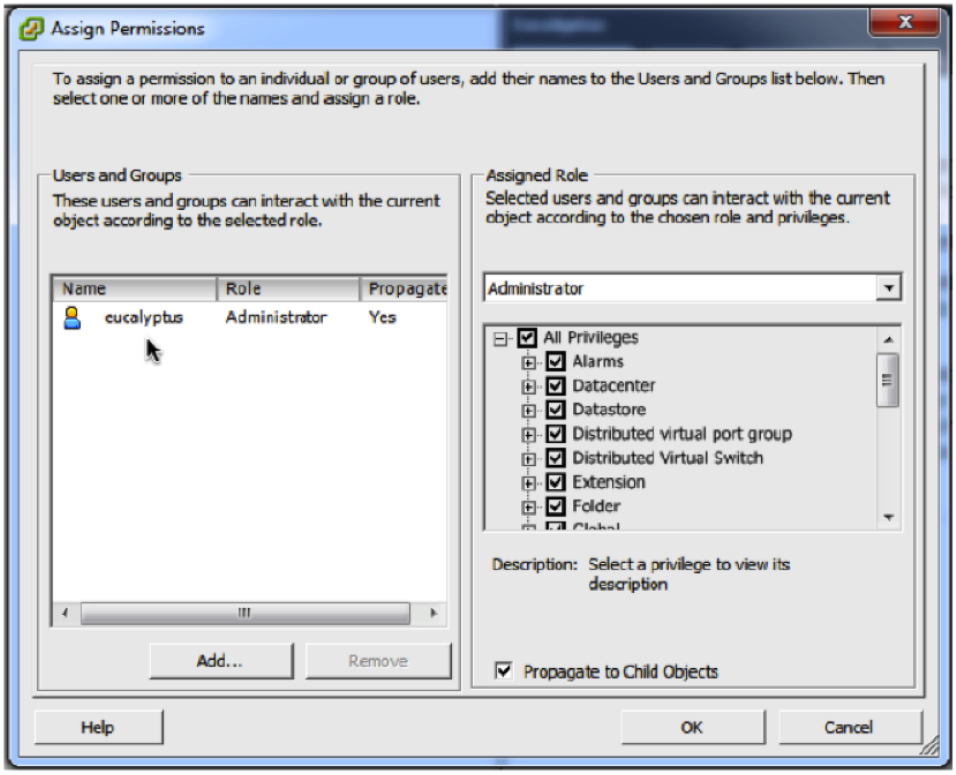
3. Enter the following command:

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

Subscription only: Configuring VMware

Tip: VMware support is available by subscription only. If you are not using VMware, skip this section.

The easiest way to configure vSphere for Eucalyptus is to give Eucalyptus unrestricted access to all vSphere endpoint(s). This way does not require complex modifications to local access permission settings. You can grant this access to Eucalyptus 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 the vSphere client.



To give a more limited amount of control to Eucalyptus over your vSphere infrastructure, create one new user and two new roles. The new user, eucalyptus, and its password are used when configuring Eucalyptus for VMware support. Define and associate these new roles with vSphere objects by doing the tasks listed in this section.

Create New User

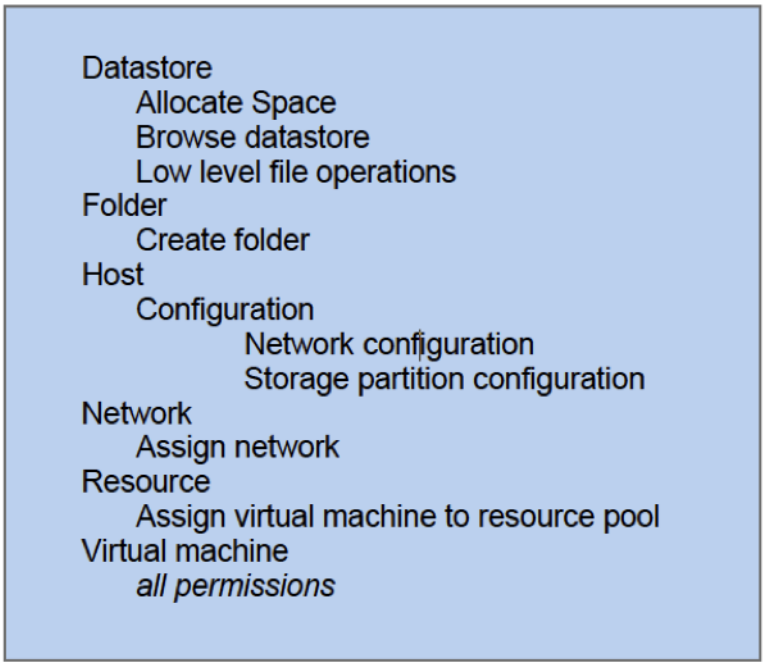
To give a more limited amount of control to Eucalyptus over your vSphere infrastructure, create one new user and two new roles. The new user (e.g., eucalyptus) and its password are used when configuring Eucalyptus for VMware. Define and associate these new roles with vSphere objects by doing the following tasks:

1. Associate a top-level role (e.g., Eucalyptus vSphere) with the eucalyptus user at the top level of the vSphere hierarchy only. You don't need to check **Propagate to Child Objects**. This role should have only one privilege:

2. Associate a resource-level role (e.g., Eucalyptus) with the eucalyptus user at the level(s) encapsulating the resources that you want 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. Check **Propagate to Child Objects**. The Eucalyptus role should have the following privileges:



You're now ready to set up a datastore.

Set Up a Datastore

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. If Eucalyptus is to be restricted in its use of available datastores, specify a datastore in Eucalyptus's configuration for VMware.

To determine the datastores that are available on a host, perform the following steps with vSphere client referencing either at vCenter Server or at a specific ESX/ESXi node:

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

You're now ready to create a network.

Create a Network

Each node must have a network reachable by the node running the Eucalyptus VMware Broker.

Tip: If more than one network is available, specify the network name in Eucalyptus configuration explicitly. Eucalyptus assumes that this network resides on the switch named "vSwitch0". If that is not the case, the switch name also must be specified in the configuration. For more information about

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

1. Click a host in left-hand side panel.
2. Click the **Configuration** tab.
3. Click **Networking** in the secondary left-hand-side panel.
4. If there is no VM Network in the list, add it by performing these steps:

a) Click **Add Networking...** in the upper-right corner.
b) Click **Virtual Machine** and click **Next**.
c) Click 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**.

Enable EBS Support

To enable VMware support for dynamic block volume support (like Amazon's Elastic Block Store) in Eucalyptus, configure each of the ESX/ESXi nodes in your infrastructure 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. Click a host in left-hand-side panel.
2. Click the **Configuration** tab.
3. Select **Networking** in the secondary left-hand-side panel.
4. If there is no **VMkernel** network listed, add it by performing the following tasks:
   a) Click **Add Networking...** in the upper-right corner.
   b) Click **VMkernel** and click **Next**.
   c) Click a switch (e.g., **Use vSwitch0**) and click **Next**.
   d) Click the label **VLAN ID** and make sure that **None(0)** is selected, then click **Next**.
   e) Choose either dynamic network config or static IP assignment, depending on your environment. When your are done, click **Next**.
   f) Click **Finish**.
5. Click **DNS and Routing** in the secondary left-hand-side panel.
6. If VMkernel does not have a gateway, add it by performing these steps:
   a) Click **Properties...** in upper-right corner.
   b) Click the **Routing** tab, enter the gateway's IP, and click **OK**.

For more information about configuring vSphere, go to the VMware website at *http://www.vmware.com/support/pubs/vs_pubs.html*.

Install VMware Tools

Ensure that VMware Tools are installed in the images that will be installed and run within the Eucalyptus cloud. These tools allow Eucalyptus to discover an instance's IP address in System networking mode. They also are required for using the `euca-bundle-instance` command when running Windows VMs in Eucalyptus. For information about installing VMware Tools, go to the VMware documentation at *http://www.vmware.com*.

Install VDDK

If you are using VMware, install the VMware's VIX DiskLib dynamic libraries (VDDK) on the machine that you plan to install the VMware Broker on (i.e., the CC machine).

As part of your Eucalyptus subscription, you will receive an entitlement certificate and a private key that allow you to download Eucalyptus. You will also receive a GPG public key to be used to verify the Eucalyptus software's integrity. The entitlement certificate file is named for the license holder and is appended with an x.y.z flag that indicates the count of certificate and the Eucalyptus version number. For example, `<license_name>-1.3.0.crt` indicates that the file is your first certificate for 3.0. The private key file is named after the license holder but does not include version or numbering information. For example: `<license_name>-1.3.0.key`. The GPG public key file is called `c1240596-eucalyptus-release-key.pub`.

Warning: The VMware VDDK library installer will not run on a system running a Xen kernel.

1. Log in to the machine that you will install the CC on.

2. Download the VDDK tarball.

```
wget --certificate <license_name>-1.3.0.crt --private-key <license_name>.key

https://downloads.eucalyptus.com/software/enterprise/3.0/dependencies/eucalyptus-deps-3.0.tar.gz
```

3. Extract the installer for the VDDK library using the following commands:

```
tar xzf eucalyptus-deps-3.0.tar.gz
cd eucalyptus-deps-3.0
tar xzf vmware-vix-disklib-distrib.tgz
cd vmware-vix-disklib-distrib
```

4. Run the installer script, accepting the End User License Agreement and selecting the default install prefix as the root directory where all folders will be placed, as shown:

```
./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
```

Configure the Firewall

If you have existing firewall rules on your hosts, you must allow Eucalyptus access.

Tip: If you are installing on Ubuntu 10.04 LTS, you can skip this section.

To enable your firewall:

1. UDP multicast to the hosts that will run the CLC, Walrus, SC, and VMwareBroker.

```
-A RH-Firewall-1-INPUT -p udp -d <network/mask> -j ACCEPT
```

2. Allow UDP between these hosts by, for example, adding the following rule to `/etc/sysconfig/iptables`.

```
-A RH-Firewall-1-INPUT -p udp -s <network/mask> -j ACCEPT
```

3. Repeat on each host that will run a Eucalyptus component: CLC, Walrus, CC, SC, and NC.

Configure SELinux

Security-enabled Linux (SELinux) is security feature for Linux that lets you to set access control through policies. Eucalyptus is not compatible with SELinux.

Tip: If you are installing on Ubuntu 10.04 LTS, you can skip this section.

To configure SELinux to allow Eucalyptus access:

1. Open /etc/selinux/config and edit the line `SELINUX=enabled` to `SELINUX=permissive`.
2. Save the file.
3. Run the following command:

```
setenforce 0
```

Configure NTP

Eucalyptus requires that each machine have the Network Time Protocol (NTP) daemon started and configured to run automatically on reboot.

To use NTP:

1. Install NTP on the machines that will host Eucalyptus components.
   - For CentOS 5, RHEL 5, and RHEL 6, run `yum install ntp`
   - For Ubuntu 10.04 LTS, run `apt-get install openntpd`
2. Open the `/etc/ntp.conf` file (`/etc/openntpd/ntpd.conf` in Ubuntu) and add NTP servers, as in the following example.

```
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
```

3. Save and close the file.
4. Configure NTP to run at reboot. (In Ubuntu, NTP is configured after you install it.)

```
chkconfig ntpd on
```

5. Start NTP. (In Ubuntu, NTP starts after you install it.)

```
service ntpd start
```

6. Synchronize your server.

```
ntpdate -u <your_ntp_server>
```

7. Synchronize your system clock, so that when your system is rebooted, it does not get out of sync.

```
hwclock --systohc
```

8. Repeat on each host that will run a Eucalyptus component.

Configure an MTA

All hosts running the CLC must run a mail transport agent server (MTA) on port 25. Eucalyptus uses the MTA to deliver or relay email messages to cloud users' email addresses. You can use Sendmail, Exim, postfix, or something simpler. The MTA server does not have to be able to receive incoming mail.

Many Linux distributions satisfy this requirement with their default MTA. For details about configuring your MTA, go to the documentation for your specific product.

To test your mail relay for localhost, send email to yourself from the terminal using `mail`.

Installing Eucalyptus

Eucalyptus installation packages are available for CentOS 5, RHEL 5, RHEL 6, and Ubuntu 10.04 LTS. The following sections show installation steps on each supported Linux distribution.

As part of your Eucalyptus subscription, you will receive an entitlement certificate and a private key that allow you to download Eucalyptus. You will also receive a GPG public key to be used to verify the Eucalyptus software's integrity. The entitlement certificate file is named for the license holder and is appended with an x.y.z flag that indicates the count of certificate and the Eucalyptus version number. For example, `<license_name>-1.3.0.crt` indicates that the file is your first certificate for 3.0. The private key file is named after the license holder but does not include version or numbering information. For example: `<license_name>-1.3.0.key`. The GPG public key file is called `c1240596-eucalyptus-release-key.pub`.

For HA: If you are installing Eucalyptus HA, pay attention to the extra steps noted in the instructions.

Install on CentOS 5

If you plan to install Eucalyptus HA, we recommend that you install each Eucalyptus component on a separate host. For example, if you are installing CLC, Walrus, CC, and SC, you will install each of these components on a separate host. You will also install each secondary component (the secondary CLC, Walrus, CC, and SC) on a separate host. In this case, you will need eight machines. Each additional cluster needs four more machines for its CCs and SCs. This does not account for NCs, which are not redundant.

To install Eucalyptus on servers running CentOS 5:

1. Move your credentials so the package managers can locate them.
   a) Place the entitlement certificate in `/etc/pki/tls/certs`.

```
mv <license_name>-1.3.0.crt /etc/pki/tls/certs/<license_name>-1.3.0.crt
```

   b) Place the private key in `/etc/pki/tls/private`.

```
mv <license_name>.key /etc/pki/tls/private/<license_name>.key
```

   c) Place the Eucalyptus GPG key in `/etc/pki/rpm-gpg`.

```
mv c1240596-eucalyptus-release-key.pub
/etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
```

2. On all servers, create a file in `/etc/yum.repos.d` called eucalyptus-enterprise.repo with the following content:

```
[eucalyptus-enterprise]
name=Eucalyptus Enterprise 3.0
baseurl=https://downloads.eucalyptus.com/software/enterprise/3.0/centos/$releasever/$basearch
gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
gpgcheck=1
sslclientcert=/etc/pki/tls/certs/<license_name>-1.3.0.crt
sslclientkey=/etc/pki/tls/private/<license_name>.key
```

3. On all systems that will run either Eucalyptus or Euca2ools, create a file in `/etc/yum.repos.d` called euca2ools.repo with the following content:

```
[euca2ools]
name=Euca2ools 2.0
baseurl=http://downloads.eucalyptus.com/software/euca2ools/2.0/centos/$releasever/$basearch
gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
gpgcheck=1
```

4. Enable the EPEL repository with the following two commands:

```
wget
http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm

rpm -Uvh epel-release-5-4.noarch.rpm
```

5. If your planned Walrus server is running Xen, install `kmod-drbd83-xen` package. Otherwise, skip this step.

```
yum install kmod-drbd83-xen
```

6. Install Eucalyptus packages. The following example shows most components being installed all on the same server. You can use different servers for each component.

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

For HA: If you are deploying HA, you must install these packages on pairs of systems. For instance, "eucalyptus-cloud" is installed on the primary CLC and the secondary CLC.

7. On each planned NC server, install the NC package:

```
yum install eucalyptus-nc
```

Important: If you are using VMware, you can skip this step. Eucalyptus software is not installed on the node machines. The nodes are running VMware.

8. If you plan to use VMware, install the subscription only VMware Broker package on each CC:

```
yum install eucalyptus-broker
```

9. After you have installed Eucalyptus, test multicast connectivity between the CLC and Walrus, SC, and the VMware Broker.

   a) Run the following receiver command on the CLC:

```
java -classpath /usr/share/eucalyptus/jgroups-2.11.1.Final.jar
org.jgroups.tests.McastReceiverTest -mcast_addr 224.10.10.10 -port 5555
```

   b) Run the following sender command on Walrus:

```
java -classpath /usr/share/eucalyptus/jgroups-2.11.1.Final.jar
org.jgroups.tests.McastSenderTest -mcast_addr 224.10.10.10 -port 5555
```

   c) Repeat the previous step on the SC and then on the VMware Broker.

For HA: If you are installing an HA environment, repeat these tasks on secondary CLC, Walrus, SC, and VMware Broker.

Your installation is complete.

You are now ready to *Configure Eucalyptus*.

Install on RHEL 5

If you plan to install Eucalyptus HA, we recommend that you install each Eucalyptus component on a separate host. For example, if you are installing CLC, Walrus, CC, and SC, you will install each of these components on a separate host. You will also install each secondary component (the secondary CLC, Walrus, CC, and SC) on a separate host. In this case, you will need eight machines. Each additional cluster needs four more machines for its CCs and SCs. This does not account for NCs, which are not redundant.

To install Eucalyptus on servers running RHEL 5:

1. Move your credentials so the package managers can locate them.
   a) Place the entitlement certificate in `/etc/pki/tls/certs`.

   ```
   mv <license_name>-1.3.0.crt /etc/pki/tls/certs/<license_name>-1.3.0.crt
   ```

   b) Place the private key in `/etc/pki/tls/private`.

   ```
   mv <license_name>.key /etc/pki/tls/private/<license_name>.key
   ```

   c) Place the Eucalyptus GPG key in `/etc/pki/rpm-gpg`.

   ```
   mv c1240596-eucalyptus-release-key.pub
   /etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
   ```

2. On all servers, create a file in `/etc/yum.repos.d` called `eucalyptus-enterprise.repo` with the following content:

   ```
   [eucalyptus-enterprise]
   name=Eucalyptus Enterprise 3.0
   baseurl=https://downloads.eucalyptus.com/software/enterprise/3.0/rhel/$releasever/$basearch
   gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
   gpgcheck=1
   sslclientcert=/etc/pki/tls/certs/<license_name>-1.3.0.crt
   sslclientkey=/etc/pki/tls/private/<license_name>.key
   ```

3. On all systems that will run either Eucalyptus or Euca2ools, create a file in `/etc/yum.repos.d` called `euca2ools.repo` with the following content:

   ```
   [euca2ools]
   name=Euca2ools 2.0
   baseurl=http://downloads.eucalyptus.com/software/euca2ools/2.0/rhel/$releasever/$basearch
   gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
   gpgcheck=1
   ```

4. Enable the Cluster-Storage repository in Red Hat Network for each SC:
   a) Go to *https://rhn.redhat.com*
   b) Navigate to the system that will run the SC.

c) Click **Alter Channel Subscriptions**.
d) Make sure the **RHEL Cluster Storage** checkbox is checked.
e) Click **Change Subscriptions**.

5. Enable the Virtualization repository in Red Hat Network for each NC:
   a) On *http://rhn.redhat.com*, navigate to the system that will run the NC.
   b) Click **Alter Channel Subscriptions**.
   c) Make sure the **RHEL Virtualization** checkbox is checked.
   d) Click **Change Subscriptions**.
6. On the machine(s) that will run Walrus, create a file in `/etc/yum.repos.d` called `centos-extras.repo` with the following content:

```
[centos-extras]
name=CentOS 5 - Extras
mirrorlist=http://mirrorlist.centos.org/?release=5&arch=$basearch&repo=extras

gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
```

7. Download the key that CentOS uses to sign their packages:

```
wget http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
```

8. Copy the resulting file to `/etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5`:

```
mv RPM-GPG-KEY-CentOS-5 /etc/pki/rpm-gpg
```

9. Enter the following command to enable the EPEL repository:

```
rpm -Uvh
http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
```

10. If your planned Walrus server is running Xen, install `kmod-drbd83-xen` package. Otherwise, skip this step.

```
yum install kmod-drbd83-xen
```

11. On all servers, enter:

```
yum update
```

12. Install Eucalyptus packages. The following example shows most components being installed all on the same server. You can use different servers for each component.

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

For HA: If you are deploying HA, you must install these packages on pairs of systems. For instance, "eucalyptus-cloud" is installed on the primary CLC and the secondary CLC.

13. On each planned NC server, install the NC package:

```
yum install eucalyptus-nc
```

Important: If you are using VMware, you can skip this step. Eucalyptus software is not installed on the node machines. The nodes are running VMware.

14. If you plan to use VMware, install the subscription only VMware Broker package on each CC server:

```
yum install eucalyptus-broker
```

15. After you have installed Eucalyptus, test multicast connectivity between the CLC and Walrus, SC, and the VMware Broker.

a) Run the following receiver command on the CLC:

```
java -classpath /usr/share/eucalyptus/jgroups-2.11.1.Final.jar
org.jgroups.tests.McastReceiverTest -mcast_addr 224.10.10.10 -port 5555
```

b) Run the following sender command on Walrus:

```
java -classpath /usr/share/eucalyptus/jgroups-2.11.1.Final.jar
org.jgroups.tests.McastSenderTest -mcast_addr 224.10.10.10 -port 5555
```

c) Repeat the previous step on the SC and then on the VMware Broker.

For HA: If you are installing an HA environment, repeat these tasks on secondary CLC, Walrus, SC, and VMware Broker.

Your installation is complete.

You are now ready to *Configure Eucalyptus*.

Install on RHEL 6

If you plan to install Eucalyptus HA, we recommend that you install each Eucalyptus component on a separate host. For example, if you are installing CLC, Walrus, CC, and SC, you will install each of these components on a separate host. You will also install each secondary component (the secondary CLC, Walrus, CC, and SC) on a separate host. In this case, you will need eight machines. Each additional cluster needs four more machines for its CCs and SCs. This does not account for NCs, which are not redundant.

To install Eucalyptus on servers running RHEL 6:

1. Move your credentials so the package managers can locate them.

a) Place the entitlement certificate in `/etc/pki/tls/certs`.

```
mv <license_name>-1.3.0.crt /etc/pki/tls/certs/<license_name>-1.3.0.crt
```

b) Place the private key in `/etc/pki/tls/private`.

```
mv <license_name>.key /etc/pki/tls/private/<license_name>.key
```

c) Place the Eucalyptus GPG key in `/etc/pki/rpm-gpg`.

```
mv c1240596-eucalyptus-release-key.pub
/etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
```

2. On all servers, create a file in `/etc/yum.repos.d` called `eucalyptus-enterprise.repo` with the following content:

```
[eucalyptus-enterprise]
name=Eucalyptus Enterprise 3.0
baseurl=https://downloads.eucalyptus.com/software/enterprise/3.0/rhel/$releasever/$basearch
gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
gpgcheck=1
sslclientcert=/etc/pki/tls/certs/<license_name>-1.3.0.crt
sslclientkey=/etc/pki/tls/private/<license_name>.key
```

3. On all servers that will run either Eucalyptus or Euca2ools, create a file in `/etc/yum.repos.d` called `euca2ools.repo` with the following content:

```
[euca2ools]
name=Euca2ools 2.0
baseurl=http://downloads.eucalyptus.com/software/euca2ools/2.0/rhel/$releasever/$basearch
gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
gpgcheck=1
```

4. Install the ELRepo repository on the machine that will run Walrus:

```
yum --nogpg install
http://elrepo.org/linux/elrepo/el6/x86_64/RPMS/elrepo-release-6-4.el6.elrepo.noarch.rpm
```

5. Install the EPEL repository by entering the following command:

```
yum --nogpg install
http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-5.noarch.rpm
```

6. On all servers, enter:

```
yum update
```

7. Install Eucalyptus packages. The following example shows most components being installed all on the same server. You can use different servers for each component.

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

For HA: If you are deploying HA, you must install these packages on pairs of systems. For instance, "eucalyptus-cloud" is installed on the primary CLC and the secondary CLC.

8. On each planned NC server, install the NC package:

```
yum install eucalyptus-nc
```

Important: If you are using VMware, you can skip this step. Eucalyptus software is not installed on the node machines. The nodes are running VMware.

9. If you plan to use VMware, install the subscription only VMware Broker package on each CC server:

```
yum install eucalyptus-broker
```

10. After you have installed Eucalyptus, test multicast connectivity between the CLC and Walrus, SC, and the VMware Broker.

    a) Run the following receiver command on the CLC:

```
java -classpath /usr/share/eucalyptus/jgroups-2.11.1.Final.jar
org.jgroups.tests.McastReceiverTest -mcast_addr 224.10.10.10 -port 5555
```

    b) Run the following sender command on Walrus:

```
java -classpath /usr/share/eucalyptus/jgroups-2.11.1.Final.jar
org.jgroups.tests.McastSenderTest -mcast_addr 224.10.10.10 -port 5555
```

    c) Repeat the previous step on the SC and then on the VMware Broker.

For HA: If you are installing an HA environment, repeat these tasks on secondary CLC, Walrus, SC, and VMware Broker.

Your installation is complete.

You are now ready to *Configure Eucalyptus*.

Install on Ubuntu 10.04 LTS

If you plan to install Eucalyptus HA, we recommend that you install each Eucalyptus component on a separate host. For example, if you are installing CLC, Walrus, CC, and SC, you will install each of these components on a separate host. You will also install each secondary component (the secondary CLC, Walrus, CC, and SC) on a separate host. In this case, you will need eight machines. Each additional cluster needs four more machines for its CCs and SCs. This does not account for NCs, which are not redundant.

To install Eucalyptus on servers running Ubuntu 10.04 LTS:

1. Copy the entitlement certificate to the `/etc/ssl/certs` directory on each server that you want to install Eucalyptus on.

```
mv <license_name>-1.3.0.crt /etc/ssl/certs/<license_name>-1.3.0.crt
```

2. Copy the private key file to the /etc/ssl/private directory on each server that you want to install Eucalyptus on.

```
mv <license_name>.key /etc/ssl/private/<license_name>.key
```

Important: Make sure that the private key's file permissions are restricted to only the root user and ssl-certs group.

3. Add the public key to the list of trusted keys:

```
apt-key add c1240596-eucalyptus-release-key.pub
```

4. On each server that you want to install Eucalyptus on, go to `/etc/apt/apt.conf.d` and create a new file (for example, `eucarepo`) with the following content:

```
Acquire {
https {
VerifyPeer "true";
SslCert "/etc/ssl/certs/<license_name>-1.3.0.crt";
```

```
SslKey "/etc/ssl/private/<license_name>.key";
};
};
```

5. Create a file in `/etc/apt/sources.list.d` called `eucalyptus-enterprise.list` with the following content:

```
deb https://downloads.eucalyptus.com/software/enterprise/3.0/ubuntu lucid
universe
```

6. On all machines that will run either Eucalyptus or Euca2ools, create a file in /etc/apt/sources.list.d called euca2ools.list with the following content:

```
deb http://downloads.eucalyptus.com/software/euca2ools/2.0/ubuntu lucid
universe
```

7. Enter the following command on all machines:

```
apt-get update
```

8. Install Eucalyptus packages and dependencies. The following example shows a package install all on the same server. You can install each component on a different server.

```
apt-get install eucalyptus-cloud eucalyptus-cc eucalyptus-sc eucalyptus-walrus
```

For HA: If you are deploying HA, you must install these packages on pairs of systems. For instance, "eucalyptus-cloud" is installed on the primary CLC and the secondary CLC.

9. On each planned NC server, install the NC package:

```
apt-get install eucalyptus-nc
```

Important: If you are using VMware, you can skip this step. Eucalyptus software is not installed on the node machines. The nodes are running VMware.

10. If you plan to use VMware, install the subscription only VMware Broker package on each CC server:

```
apt-get install eucalyptus-broker
```

11. After you have installed Eucalyptus, test multicast connectivity between the CLC and Walrus, SC, and the VMware Broker.

    a) Run the following receiver command on the CLC:

```
java -classpath /usr/share/eucalyptus/jgroups-2.11.1.Final.jar
org.jgroups.tests.McastReceiverTest -mcast_addr 224.10.10.10 -port 5555
```

    b) Run the following sender command on Walrus:

```
java -classpath /usr/share/eucalyptus/jgroups-2.11.1.Final.jar
org.jgroups.tests.McastSenderTest -mcast_addr 224.10.10.10 -port 5555
```

    c) Repeat the previous step on the SC and then on the VMware Broker.

For HA: If you are installing an HA environment, repeat these tasks on secondary CLC, Walrus, SC, and VMware Broker.

Your installation is complete.

You are now ready to *Configure Eucalyptus*.

Configuring Eucalyptus

This section describes the parameters that need to be set in order to launch Eucalyptus for the first time. The first launch of Eucalyptus is different than a restart of a previously running Eucalyptus deployment in that it sets up the security mechanisms that will be used by the installation to ensure system integrity.

Eucalyptus configuration is stored in a text file, `/etc/eucalyptus/eucalyptus.conf`, that contains key-value pairs specifying various configuration parameters. Eucalyptus reads this file when it launches and when various forms of reset commands are sent it the Eucalyptus components.

Important: Perform the following tasks after you install Eucalyptus software, but before you start the Eucalyptus services.

Configure Network Modes

This section provides detailed configuration instructions for each of the four Eucalyptus networking modes. Eucalyptus requires network connectivity between its clients (end-users) and the cloud components (CC, CLC, and Walrus). In Managed and Managed (No VLAN) modes, traffic to instances pass through the CC. So, in these two modes clients must be able to connect to the CC. In System and Static modes, clients need to connect directly to the NC. The CC does not act as a router in these two modes.

The `/etc/eucalyptus/eucalyptus.conf` file contains all network-related options in in the “Networking Configuration” section. These options use the prefix VNET_. The most commonly used VNET options are described in the following table. The set of networking settings that apply to a cloud varies based on its networking mode. Each setting in this section lists the modes in which it applies. Unless otherwise noted, all of these settings apply only to CCs.

The `/etc/eucalyptus/eucalyptus.conf` file contains all network-related options in in the Networking Configuration section. These options use the prefix VNET_. The most commonly used VNET options are described in the following table.

Option	Description	Modes
`VNET_MODE`	The networking mode in which to run. The same mode must be specified on all CCs and NCs in the entire cloud. Valid values: `STATIC`, `SYSTEM`, `MANAGED`, `MANAGED-NOVLAN`, Default: `SYSTEM`	All
`VNET_PRIVINTERFACE`	The name of the network interface that is on the same network as the NCs. In Managed and Managed (No VLAN) modes this must be a bridge for instances in different clusters but in the same security group to be able to reach one another with their private addresses. Default: `eth0`	Static Managed Managed (No VLAN)
`VNET_PUBINTERFACE`	**On a CC**, this is the name of the network interface that is connected to the “public” network. **On an NC**, this is the name of the network interface that is connected to the same network as the CC. Depending on the hypervisor's configuration this may be a bridge or a physical interface that is attached to the bridge. Default: `eth0`	Managed

Option	Description	Modes
`VNET_BRIDGE`	On an NC, this is the name of the bridge interface to which instances' network interfaces should attach. A physical interface that can reach the CC must be attached to this bridge. Common settings include `xenbr0` for older Xen versions, `eth0` for newer Xen versions, and `br0` for KVM.	Static System Managed (No VLAN)
`VNET_MACMAP`	A map of MAC addresses to IP addresses that Eucalyptus should allocate to instances when running in Static mode. Separate MAC addresses and IP addresses with = characters. Separate pairs with spaces. Example: `VNET_MACMAP="00:01:02:03:04:05=192.168.1.1 A1:A2:A3:A4:A5:A6=192.168.1.2"`	Static
`VNET_PUBLICIPS`	A space-separated list of individual and/or hyphenated ranges of public IP addresses to assign to instances. If this is undefined then instances will receive only private IP addresses. For example: `VNET_PUBLICIPS="173.205.188.140-173.205.188.254"`	Managed Managed (No-VLAN)
`VNET_SUBNET`, `VNET_NETMASK`	The address and network mask of the network the cloud should use for instances' private IP addresses.	Static Managed Managed (No VLAN)
`VNET_ADDRSPERNET`	Eucalyptus assigns a distinct subnet of private IP addresses to each security group. This setting dictates how many addresses each of these subnets should contain. Specify a power of 2 between 16 and 2048. This is directly related, though not equal, to the number of instances that may reside in each security group.	Managed , Managed (No VLAN)
`VNET_DNS`	The address of the DNS server to supply to instances in DHCP responses.	Static Managed Managed (No VLAN)
`VNET_BROADCAST`, `VNET_ROUTER`	The network broadcast and default gateway to supply to instances in DHCP responses.	Static
`VNET_LOCALIP`	By default the CC automatically determines which IP address to use when setting up tunnels to other CCs. Set this to the IP address that other CCs can use to reach this CC if tunneling does not work.	Managed Managed (No-VLAN)
`VNET_DHCPDAEMON`	The ISC DHCP executable to use. This is set to a distro-dependent value by packaging. The internal default is `/usr/sbin/dhcpd3`.	Static Managed Managed (No VLAN)
`VNET_DHCPUSER`	The user the DHCP daemon runs as on your distribution. For CentOS 5, RHEL5, and RHEL 6 this is typically `root`. In Ubuntu 10.04 LTS, this is typically `dhcpd`. Default: `dhcpd`	Static Managed Managed (No VLAN)

Managed Mode

In Managed mode, Eucalyptus manages the local network of VM instances and provides all networking features Eucalyptus currently supports, including VM network isolation, security groups, elastic IPs, and metadata service. Configure each CC to use an Ethernet device that lies within the same broadcast domain as all of its NCs.

Important: In Managed mode, each security group requires a separate subnet and a separate VLAN that Eucalyptus controls and maintains. So the underlying physical network must be "VLAN clean." For more information about VLAN clean, see *Prepare VLAN*.

To configure for Managed mode:

CLC Configuration

No network configuration required.

CC Configuration

Important: You must set `VNET_PUBLICIPS` identically on all CCs in a multi-cluster configuration.

1. Log in to the CC and open the `/etc/eucalyptus/eucalyptus.conf` file.
2. Go to the **Network Configuration** section, uncomment and set the following:

```
VNET_MODE="MANAGED"

VNET_SUBNET="<subnet for instances' private IPs. Example: 192.168.0.0>"
VNET_NETMASK="<your netmask for the vnet_subnet. Example: 255.255.0.0>"
VNET_DNS="<your DNS server's IP>"
VNET_ADDRSPERNET="<# of simultaneous instances per security group>"

VNET_PUBLICIPS="<your_free_public_ip1 your_free_public_ip2 …>"

VNET_LOCALIP="<the IP of the local interface on the cc that is reachable from
 CLC>"

VNET_DHCPDAEMON="<path to DHCP daemon binary. Example: /usr/sbin/dhcpd3>"

VNET_DHCPUSER="<DHCP user name. Example: dhcpd>"
```

3. If your NCs are not reachable from end-users directly and the CC has two (or more) Ethernet devices of which one connects to the client/public network and one connects to the NC network, or the single Ethernet device that the CC uses to connect to both clients and NCs is NOT 'eth0', then you must also uncomment and set:

```
VNET_PRIVINTERFACE="<Ethernet device on same network as NCs. Example: eth1>"

VNET_PUBINTERFACE="<Ethernet device on 'public' network. Example: eth0>"
```

4. Save the file.
5. Repeat on each CC in your system.

Important: Each CC must have the same configuration with the exception of the `VNET_LOCALIP` value, which should be machine-specific.

NC Configuration

1. Log into an NC machine and open the `/etc/eucalyptus/eucalyptus.conf` file.

2. Go to the **Network Configuration** section, uncomment and set the following:

```
VNET_MODE="MANAGED"
VNET_PUBINTERFACE="<Ethernet device/bridge reachable from cc machine. Example:
 eth0>"
```

3. Save the file.
4. Repeat on each NC.

Managed (No-VLAN) Mode

In Managed (No-VLAN) mode, Eucalyptus does not use VLANs to isolate the network bridges attached to VMs from each other. Configure each CC to use an Ethernet device that lies within the same broadcast domain as all of its NCs.

To configure for Managed (No VLAN) mode:

CLC Configuration

No network configuration required.

CC Configuration

Important: You must set `VNET_PUBLICIPS` identically on all CCs in a multi-cluster configuration.

1. Log in to the CC and open the `/etc/eucalyptus/eucalyptus.conf` file.
2. Go to the **Network Configuration** section, uncomment and set the following:

```
VNET_MODE="MANAGED-NOVLAN"
VNET_SUBNET="<your subnet for instance's private IPs. Example: 192.168.0.0>"
VNET_NETMASK="<your netmask for the vnet_subnet. Example: 255.255.0.0>"
VNET_DNS="<your DNS server IP>"
VNET_ADDRSPERNET="<# of simultaneous instances per security group>"

VNET_PUBLICIPS="<your_free_public_ip1 your_free_public_ip2 …>"

VNET_LOCALIP="<the IP of the local interface on the cc that is reachable from
 CLC>"

VNET_DHCPDAEMON="<path to DHCP daemon binary. Example: /usr/sbin/dhcpd3>"

VNET_DHCPUSER='<DHCP user. Example: dhcpd>"
```

3. If your NCs are not reachable from end-users directly and the CC has two (or more) Ethernet devices of which one connects to the client/public network and one connects to the NC network, or the single Ethernet device that the CC uses to connect to both clients and NCs is NOT 'eth0', then you must also uncomment and set:

```
VNET_PRIVINTERFACE="<Ethernet device on same network as NCs. Example: eth1>"

VNET_PUBINTERFACE="<Ethernet device on 'public' network. Example: eth0>"
```

4. Save the file.
5. Repeat on each CC in your system.

Important: Each CC must have the same configuration with the exception of the `VNET_LOCALIP` value, which should be machine-specific.

NC Configuration

1. Log into an NC machine and open the `/etc/eucalyptus/eucalyptus.conf` file.
2. Go to the **Network Configuration** section, uncomment and set the following:

```
VNET_MODE="MANAGED-NOVLAN"
VNET_BRIDGE="<bridge name. Example: xenbr0>"
```

3. Save the file.
4. Repeat on each NC.

System Mode

In System mode, Eucalyptus mostly stays out of the way in terms of VM networking, relying on your local DHCP service to configure VM networks. The NC has to specify a bridge, and that it is the bridge that is connected to an Ethernet network that has a reachable DHCP server running elsewhere that is configured to hand out IP addresses dynamically.

To configure for System mode:

CLC Configuration

No network configuration required.

CC Configuration

1. Log in to the CC and open the `/etc/eucalyptus/eucalyptus.conf` file.
2. Go to the **Network Configuration** section, uncomment and set the following:

```
VNET_MODE="SYSTEM"
```

3. Save the file.
4. Repeat on each CC in your system.

NC Configuration

1. Log into an NC machine and open the `/etc/eucalyptus/eucalyptus.conf` file.
2. Go to the **Network Configuration** section, uncomment and set the following:

```
VNET_MODE="SYSTEM"
VNET_BRIDGE="<name of bridge on same network as the DHCP server. Example:
xenbr0>"
```

3. Save the file.
4. Repeat on each NC.

Static Mode

Static mode requires you to specify the network configuration each VM should receive from the Eucalyptus DHCP server running on the same physical server as the CC component. Configure each CC to use an Ethernet device that lies within the same broadcast domain as all of its NCs.

To configure for Static mode:

CLC Configuration

No network configuration required.

CC Configuration

1. Log in to the CC and open the `/etc/eucalyptus/eucalyptus.conf` file.
2. Go to the **Network Configuration** section, uncomment and set the following:

```
VNET_MODE="STATIC"

VNET_SUBNET="<subnet for private IP addresses for instances. Example:
192.168.1.0>"

VNET_NETMASK="<netmask for the subnet in vnet_subnet. Example: 255.255.255.0>"

VNET_BROADCAST="<broadcast IP to supply to instances in DHCP responses.
Example: 192.168.1.255>"

VNET_ROUTER="<subnet router IP/gateway IP to supply to instances in DHCP
responses>"

VNET_DNS="<IP of your DNS server>"

VNET_MACMAP="<MAC-to-IP mapping for your VMs. Example:
AA:BB:CC:DD:EE:FF=192.168.1.1
A1:B1:C1:D1:E1:F1=192.168.1.2>"

VNET_PRIVINTERFACE="<Ethernet device on same network as the NCs. Example:
eth0>"

VNET_DHCPDAEMON="<path to DCHP daemon binary. Example /usr/sbin/dhcp3d>"

VNET_DHCPUSER="<DHCP user name. Example: dhcpd>"
```

3. Save the file.
4. Repeat on each CC in your system.

NC Configuration

1. Log into an NC machine and open the `/etc/eucalyptus/eucalyptus.conf` file.
2. Go to the **Network Configuration** section, uncomment and set the following:

```
VNET_MODE="STATIC"
VNET_BRIDGE="<name of bridge on the same network as the CC. Examples: xenbr0
 or eth0>"
```

3. Save the file.
4. Repeat on each NC.

Configure Hypervisors

Eucalyptus interacts with both Xen and KVM hypervisors through libvirt. This section details steps to configuring these hypervisors.

Important: Make sure that you enable hardware virtualization before you start these steps.

CentOS 5

The default settings that ship with CentOS 5.6 are generally appropriate except you must enable `xend-http-server` and restart the daemon.

To make sure the Xen daemon is set up correctly:

1. Log in to an NC and open the `/etc/xen/xend-config.sxp` file.
2. Verify these Eucalyptus-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 '')
```

3. Open the `/etc/libvirt/libvirtd.conf` file.
4. Uncomment the following lines and change the value for `unix_sock_group` from `libvirt` to `eucalyptus`:

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

5. Save the file and restart the libvirtd daemon.
6. Use the virsh list command to confirm that the eucalyptus user can communicate with libvirt.

```
su -c "virsh list" eucalyptus
Id Name                 State
----------------------------------
0 Domain-0
```

   The command returns a Domain-0 for user eucalyptus. If the command doesn't succeed, double-check the steps and setting.

7. If you are running Xen on your NC, the `euca-get-console-output` command will not work, unless you do the following:
   a) On the NC, open the `/etc/sysconfig/xend`.
   b) Uncomment the following line:

```
XENCONSOLED_LOG_GUESTS=yes
```

   c) Save the file and restart xend.
8. Repeat for each NC server in your system.

RHEL 5

To make sure the Xen daemon is set up correctly:

1. Log in to an NC and open the `/etc/xen/xend-config.sxp` file.
2. Verify these Eucalyptus-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 '')
```

3. Open the `/etc/libvirt/libvirtd.conf` file.
4. Uncomment the following lines and change the value for `unix_sock_group` from `libvirt` to `eucalyptus`:

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

5. Save the file and restart the libvirtd daemon.
6. Use the virsh list command to confirm that the eucalyptus user can communicate with libvirt.

```
su -c "virsh list" eucalyptus
Id Name                 State
----------------------------------
0 Domain-0
```

   The command returns a Domain-0 for user eucalyptus. If the command doesn't succeed, double-check the steps and setting.
7. If you are running Xen on your NC, the `euca-get-console-output` command will not work, unless you do the following:
   a) On the NC, open the `/etc/sysconfig/xend`.
   b) Uncomment the following line:

```
XENCONSOLED_LOG_GUESTS=yes
```

   c) Save the file and restart xend.
8. Repeat for each NC server in your system.

RHEL 6

No additional configuration is required for KVM or libvirt on RHEL 6.

Ubuntu 10.04 LTS

No additional configuration is required for KVM or libvirt on Ubuntu 10.04 LTS.

Configure Loop Devices

To ensure that Eucalyptus starts new instances, you must configure the number of loop devices you expect to use for SC and NC components. An SC with insufficient loop devices fails to create new EBS volumes. An NC with insufficient loop devices fails to start new instances. This section tells you how to configure loop devices for your distribution.

We recommend that you err on the side of configuring too many loop devices. Too many loop devices result in a minor amount of memory tie-up and some clutter added to the system's `/dev` directory. Too few loop devices make Eucalyptus unable to use all of a system's resources.

Eucalyptus installs with a default loop device amount of 256. However, Eucalyptus cannot control the number of loop devices on either CentOS 5 or RHEL 5. These systems default to eight loop devices, so you must supply a option to the loop driver by writing a configuration file. For more information, see the instructions for either *CentOS 5* or *RHEL 6*.

If you want to change the default loop device number in RHEL 6 or Ubuntu 10.04 LTS, see the instructions for either *RHEL 5* or *Ubuntu 10.04 LTS*.

Tip: We recommend a minimum of 50 loop devices. If you have fewer than 50, the startup script will complain.

CentOS 5

1. Log in to the SC server.
2. Create and open a file, `/etc/modprobe.d/eucalyptus-loop`.
3. Enter the following line:

```
options loop max_loop=N
```

   where N is an integer from 1 to 256

4. Reload the loop driver if it is already loaded.

```
rmmod loop
modprobe loop
```

5. Repeat for each SC and NC server.

RHEL 5

1. Log in to the SC server.
2. Create and open a file, `/etc/modprobe.d/eucalyptus-loop`.
3. Enter the following line:

```
options loop max_loop=N
```

   where N is an integer from 1 to 256

4. Reload the loop driver if it is already loaded.

```
rmmod loop
modprobe loop
```

5. Repeat for each SC and NC server.

RHEL 6

You don't need to make a configuration change for RHEL 6 machines unless you want to change the default loop device value of 256. To change the default value, perform the tasks that follow.

1. Log in to the SC server and open the `/etc/eucalyptus/eucalyptus.conf` file.
2. Uncomment the following line:

```
# CREATE_SC_LOOP_DEVICES=256
```

3. Replace 256 with the number of loop devices.
4. Repeat for each SC on your system.
5. Log in to an NC server and open the `/etc/eucalyptus/eucalyptus.conf` file.
6. Uncomment the following line:

```
# CREATE_NC_LOOP_DEVICES=256
```

7. Replace 256 with the number of loop devices.

8. Repeat for each NC on your system.

Ubuntu 10.04 LTS

1. Log in to the SC server and open the `/etc/eucalyptus/eucalyptus.conf` file.
2. Uncomment the following line:

```
# CREATE_SC_LOOP_DEVICES=256
```

3. Replace 256 with the number of loop devices.
4. Repeat for each SC on your system.
5. Log in to an NC server and open the `/etc/eucalyptus/eucalyptus.conf` file.
6. Uncomment the following line:

```
# CREATE_NC_LOOP_DEVICES=256
```

7. Replace 256 with the number of loop devices.
8. Repeat for each NC on your system.

Configure Multi-Cluster Networking

Eucalyptus supports multiple clusters within a single Eucalyptus cloud. This section briefly describes how Eucalyptus manages the networking aspect of a multi-cluster setup.

In System or Static networking modes, Eucalyptus does not perform any special configuration for a multi-cluster setup. In Managed and Managed (No VLAN) modes, Eucalyptus sets up Layer 2 Tunneling Protocol (L2TP) between your clusters. This means that virtual machines in the same security group, but distributed across clusters (potentially each in their own broadcast domain), can communicate with one another. Eucalyptus uses the VTun package to handle all L2TP tunnels between clusters. If VTun is installed on each of your CCs, multi-cluster tunneling is automatically handled by each CC.

Depending on the networking mode and network topology, keep the following network configuration considerations in mind.

Managed Mode:	During normal operation, you will see many tunnel interfaces being created and destroyed as virtual networks are constructed and torn down.
Managed (No VLAN) Mode:	In order for VTun tunneling to work in this mode, you must configure each CC with a bridge as its primary, public interface (`VNET_PUBINTERFACE`).
Managed Mode and Managed (No VLAN) Mode:	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 the `eucalyptus.conf` file. Do not run two CCs in the same broadcast domain with tunneling enabled, as this will potentially lead to a broadcast storm as tunnels start forwarding packets in a loop on your local network.

If you want to disable tunneling altogether, set `DISABLE_TUNNELING=y` in `eucalyptus.conf`.

Manage IP Tables Rules

In Managed and Managed (No VLAN) modes, Eucalyptus flushes the CC's iptables rules for both `filter` and `nat`, then it sets the default policy for the `FORWARD` chain in `filter` to `DROP`. At run time, the CC adds and removes rules from `FORWARD` as users add and remove ingress rules from their active security groups. In addition, the `nat` table is

configured to allow VMs access to the external network using IP masquerading, and dynamically adds/removes rules in the `nat` table as users assign and unassign public IPs to VMs at instance boot or run-time.

If you have rules you want to apply on the CC, make the following edit on the CC before you start Eucalyptus or while Eucalyptus is stopped:

```
iptables-save > /var/run/eucalyptus/net/iptables-preload
```

Caution: Performing this operation to define special iptables rules that are loaded when Eucalyptus starts, you could cause Eucalyptus VM networking to fail. We recommend that you only do this if you are completely sure that it will not interfere with the operation of Eucalyptus.

Starting Eucalyptus

Make sure that each host you installed a Eucalyptus component on resolves to an IP address. Edit the `/etc/hosts` file if necessary.

Start the Eucalyptus components in the order presented in this guide.

Start the CLC

1. Log in to the CLC.
2. Enter the following command to initialize the CLC:

```
/usr/sbin/euca_conf --initialize
```

Note: This command might take a minute or more to finish.

3. Enter the following command to start the CLC:

```
service eucalyptus-cloud start
```

For HA: For an HA environment, start the secondary CLC. Do not initialize the secondary CLC. Just start it.

Start Walrus

Important: If you installed Walrus on the same host as the CLC, skip this step.

To start Walrus:

Log in to the Walrus server and enter the following command:

```
service eucalyptus-cloud start
```

For HA: For an HA environment, repeat this task on the secondary Walrus.

Start the CC

To start the CC:

1. Log in to the CC server and enter the following:

```
service eucalyptus-cc start
```

2. If you have a multi-cluster setup, repeat this step on the CC in each cluster.

For HA: For an HA environment, repeat this task on the secondary CC in each cluster.

Start the VMware Broker

If you are using Eucalyptus with VMware support, start the VMware Broker. The VMware Broker is located on the CC. Otherwise, skip this step.

1. Log in to the CC server and enter the following:

```
service eucalyptus-cloud start
```

2. If you have a multi-cluster setup, repeat this step on the CC in each cluster.

For HA: For an HA environment, repeat this task on the secondary CC in each cluster.

Start the SC

Important: If you installed SC on the same host as the CLC, skip this step.

To start the SC:

1. Log in to the SC server and enter the following command:

```
service eucalyptus-cloud start
```

2. If you have a multi-cluster setup, repeat this step on the SC in each cluster.

For HA: For an HA environment, repeat this task on the secondary SC in each cluster.

Start the NCs

1. Log in to an NC server and enter the following command:

```
service eucalyptus-nc start
```

2. Repeat for each NC server.

Verify the Startup

At this point, all Eucalyptus components are enabled and starting up. Some of these services perform intensive initialization at start-up, particularly the first time they are started. You might have to wait a few minutes until they are fully operational.

One quick way to determine if the components are running is to run netstat on the various hosts and look to see when the service ports are allocated to a process. Specifically, the CLC, Walrus, the SC, and the VMware Broker allocate ports 8773. The CC listens to port 8774, and the NC uses port 8775.

Verify that everything has started without error. Expected outcomes include:

- The CLC is listening on ports 8443 and 8773
- Walrus is listening on port 8773
- The SC is listening on port 8773
- The CC is listening on port 8774
- VMware Broker is listening on port 8773
- The NCs are listening on port 8775
- Log files are being written to `/var/log/eucalyptus/`

Registering Eucalyptus

Eucalyptus implements a secure protocol for registering separate components so that the overall system can't be tricked into including a component run by an unauthorized administrator or user. You only need to register components the first time Eucalyptus is started after it was installed.

Most registration commands run on the CLC server. NCs, however, are registered on each CC. You must register each node controller on every CC for the cluster on which the node controller participates.

Note that each registration command will attempt an SSH as root to the remote physical host where the registering component is assumed to be running. The registration command also contacts the component so it must be running at the time of the command is issued. If a password is required to allow SSH access, the command will prompt the user for it.

Except for NCs, each registration command requires four pieces of information:

- The **component** you are registering, because this affects where the commands must be executed.
- The **partition** the component will belong to. The partition is the same thing as availability zone in AWS.
- The **name** ascribed to the component. This is the pretty name used to identify the component in a human friendly way. This name when reporting about system state changes which require attention. This name must be globally-unique with respect to other component registrations.
- The **IP address** of the service being registered.

NCs only have two pieces of information: component name and IP address.

Register the Secondary Cloud Controller

For HA: If you installed HA, register the secondary CLC. Otherwise, skip this section.

Log in to the primary CLC and enter the following command to register the secondary CLC:

```
/usr/sbin/euca_conf --register-cloud --partition eucalyptus
--host <clc_#2_IP_address> --component <clc_name>
```

The partition name for the CLC has to be `eucalyptus`.

Register Walrus

To register Walrus:

On the CLC server, enter the following command:

```
/usr/sbin/euca_conf --register-walrus --partition walrus --host
<walrus_IP_address> --component <walrus_name>
```

For HA: For HA, register the secondary Walrus the same way, using the secondary Walrus IP address and secondary Walrus name. Use the same partition name as the primary Walrus.

Register the CC

To register the CC:

1. On the CLC, enter the following command:

```
/usr/sbin/euca_conf --register-cluster --partition <partition_name>
 --host <CC_IP_address> --component <cc_name>
```

Tip: We recommend that you set the `partition` name to a descriptive name for the availability zone controlled by the CC. The `component` is a unique name. We recommend that you use a short-hand name of the hostname or IP address of the machine, like `cc-host` or `cc-ip`. We recommend hostname.

2. Repeat for each cluster, replacing the CC name, partition name, CC IP address, and CC name.

For HA: For HA, register the secondary CC the same way, replacing the CC IP address and CC name, but using the same partition name as the primary CC.

Register the VMware Broker

Tip: If you aren't using VMware, skip this section.

To register the VMware Broker

1. On the CLC, enter the following command:

```
/usr/sbin/euca_conf --register-vmwarebroker --partition <partition_name>
 --host <CC_IP_address> --component <vmwarebroker_name>
```

Important: Register the VMware Broker component using the CC IP address, not the CLC IP address.

2. Repeat for each cluster, replacing the VMware Broker name, partition name, CC IP address, and CC name.

For HA: For HA, register the secondary VMware Broker the same way, using the secondary CC IP address and CC name, but using the same partition name as the primary CC.

Register the SC

To register the SC:

1. On the CLC, enter the following command:

```
/usr/sbin/euca_conf --register-sc --partition <partition_name> --host
<SC_IP_address>
 --component <SC_name>
```

Important: An SC must have same partition name as the CC in the same cluster.

2. Repeat for each cluster, replacing the SC name, partition name, SC IP address, and SC name.

For HA: For HA, register the secondary SC the same way, using the secondary SC IP address and SC name, but using the same partition name as the primary SC.

Register the NCs

Important: If you are using VMware, you can skip this task. You do not have to register the NCs. Eucalyptus software is not installed on the node machines. The nodes are running VMware.

1. On a CC, register all NCs using the following command with the IP address of each NC server:

```
/usr/sbin/euca_conf --register-nodes "<node0_IP_address> ...
<nodeN_IP_address>"
```

2. Repeat each cluster in your cloud.

The IP addresses of the NCs are space delimited, as in the following example:

```
/usr/sbin/euca_conf --register-nodes "192.168.71.154 192.168.71.155
 192.168.71.159"
```

For HA: For HA, you must also register the NCs with the secondary CC.

Register Arbitrators

Eucalyptus uses a periodic ICMP echo test to an Arbitrator. This test approximates an end user's ability to access the system. If Eucalyptus determines that it cannot reach the host associated with a registered Arbitrator, all Eucalyptus services operating on that host attempt to failover to the alternate hosts running those services.

For HA: In HA, you can register each Arbitrator service on the primary and secondary CLC and Walrus. If you are using either Managed or Managed (No VLAN) mode, you can also register Arbitrator services on both the primary CC and the secondary CC.

We recommend that you register more than one Arbitrator for each Eucalyptus component. This will allow for normal outages and maintenance. There is no limit on the number of Arbitrators on a CLC and a Walrus. You can only register up to three on a CC.

Register an Arbitrator service on each host that has a cloud component (CLC or Walrus) installed. An Arbitrator is a host-wide component: when an Arbitrator is registered on a host, it is registered with all cloud components enabled on that host. A separate arbitrator has to be registered per each network entity that needs to be monitored from the host.

To register an Arbitrator:

1. Log in to the primary CLC.
2. Enter the following command to register an arbitrator:

```
/usr/sbin/euca_conf --register-arbitrator --partition <ID>
 --component <ID> --host <target_host>
```

where:

- `<ID>` is a globally unique ID that identifies an Arbitrator. Note that you must use the same <ID> as both a partition and component ID.

- `<target_host>` is the IP or hostname running the Eucalyptus component that will run the Arbitrator.

For example:

```
euca_conf --register-arbitrator --partition EXAMPLE_ARB --component EXAMPLE_ARB
 --host 192.168.1.10
```

3. Repeat for the secondary CLC and for both Walrus servers.
4. Define the gateway for each Arbitrator:

```
/usr/sbin/euca-modify-property -p <ID>.arbitrator.gatewayhost=<gateway>
```

where:

- `<ID>` is the globally unique ID of the registered Arbitrator.
- `<gateway>` is an external hostname or IP address used to approximate connectivity to the end user.

For example:

```
euca-modify-property -p EXAMPLE_ARB.arbitrator.gatewayhost=192.168.1.1
```

5. Repeat for each registered Arbitrator.
6. To register on each CC, log in to the primary CC, and open the `/etc/eucalyptus/eucalyptus.conf` file.
7. Provide a list of Arbitrators (up to three) as values for the `CC_ARBITRATORS` property. For example:

```
CC_ARBITRATORS="192.168.48.11 192.168.48.12"
```

8. Save the file and restart the CC.

```
service eucalyptus-cc restart
```

9. Repeat on the secondary CC.

In the following example, the primary CLC is on <CLC_host_p>, the secondary CLC is on <CLC_host_s>, the primary Walrus is on <Walrus_host_p>, and the secondary Walrus is on <Walrus_host_s>.

```
/usr/sbin/euca_conf --register-arbitrator --host <CLC_host_p>
--component ARB00 --partition ARB00
/usr/sbin/euca_conf --register-arbitrator --host <CLC_host_p>
--component ARB01 --partition ARB01
/usr/sbin/euca_conf --register-arbitrator --host <Walrus_host_p>
--component ARB02 --partition ARB02
/usr/sbin/euca_conf --register-arbitrator --host <Walrus_host_p>
--component ARB03 --partition ARB03
/usr/sbin/euca_conf --register-arbitrator --host <CLC_host_s>
--component ARB04 --partition ARB04
/usr/sbin/euca_conf --register-arbitrator --host <CLC_host_s>
--component ARB05 --partition ARB05
/usr/sbin/euca_conf --register-arbitrator --host <Walrus_host_s>
--component ARB06 --partition ARB06
/usr/sbin/euca_conf --register-arbitrator --host <Walrus_host_s>
--component ARB07 --partition ARB07
```

Configuring the Runtime Environment

After Eucalyptus is installed and registered, perform the tasks in this section to configure the runtime environment.

Generate Administrator Credentials

Now that you have installed and configured Eucalyptus, you're ready to start using it. To do so, you must generate credentials.

Important: When you run the `euca_conf --get-credentials` command, you are requesting the access and secret keys and an X.509 certificate and key. You cannot retrieve an existing X.509 certificate and key. You can only generate a new pair.

To generate a set of credentials:

1. Generate administrator credentials.

```
/usr/sbin/euca_conf --get-credentials admin.zip

unzip admin.zip
```

2. Source the `eucarc` file.

```
source eucarc
```

You are now able to run Eucalyptus commands.

Configure SAN Support

To use the SAN integration feature that Eucalyptus provides, configure supported SAN devices to allow Eucalyptus to manage the SAN. Eucalyptus automatically creates and tears down volumes, snapshots, and data connections from guest instances. The administrator does not need to pre-allocate volumes or LUNs for Eucalyptus.

Eucalyptus offers SAN support for EBS. The SANManager module directs the Storage Controller (SC) to manage supported SAN devices. Enabling the SANManager requires that you configure the `CLOUD_OPTS` variable in eucalyptus.conf so that Eucalyptus loads the SANManager module and selects your SAN device at start time.

To configure SAN support:

1. *Enable SAN access on Eucalyptus*
2. Enable the specific SAN device (*Dell*, *Netapp*, or *JBOD*)

Enable SANManager

To enable the Eucalyptus SANManager:

Enter the following commands:

```
euca-modify-property -p
<partition_name>.storage.sanhost=<SAN_hostname_or_IP_address>
euca-modify-property -p <partition_name>.storage.sanuser=<SAN_admin_user_name>
euca-modify-property -p
<partition_name>.storage.sanpassword=<SAN_admin_password>
```

Enable Dell Equallogic SANs

1. On the SC, open the `/etc/eucalyptus/eucalyptus.conf` file and make the following configuration:

```
CLOUD_OPTS="-Debs.storage.manager=SANManager
 -Debs.san.provider=EquallogicProvider"
```

2. Restart the SC. If the SC is not installed separately, restart the CLC.

```
service eucalyptus-cloud restart
```

Enable NetApp SANs

NetApp Filer devices (FAS 2000 and FAS 6000 series) are managed by Eucalyptus using NetApp ONTAPI (version 7.3.3 or above). Enable ONTAPI and provide administrative Eucalyptus with access to OTAPI. To configure NetApp Filer and enable NetApp SANs:

1. Enable and verify API access to NetApp Filer.
2. Verify that an aggregate with sufficient spare capacity exists.
   - If you have SSH access to the NetApp Filer, enter `aggr show_space`.
   - If an aggregate with spare capacity does not exist, create one using the `aggr create` command.
3. Verify that you have a license for FlexClone installed. At the shell prompt, enter `license` to see the list of all installed licenses.
4. Write down the administrator credentials (or create an administrator account for Eucalyptus). You will need to configure Eucalyptus with these credentials later.
5. On the SC, open the `/etc/eucalyptus/eucalyptus.conf` file and make the following configuration:

```
CLOUD_OPTS="-Debs.storage.manager=SANManager
 -Debs.san.provider=NetappProvider"
```

6. Restart the SC. If the SC is not installed separately, restart the CLC.

```
service eucalyptus-cloud restart
```

Enable Direct Attached Storage (JBOD) SANs

1. On the SC, open the `/etc/eucalyptus/eucalyptus.conf` file and make the following configuration:

```
CLOUD_OPTS="-Debs.storage.manager=DASManager"
```

2. Restart the SC. If the SC is not installed separately, restart the CLC.

```
service eucalyptus-cloud restart
```

Configure DNS

Eucalyptus provides a DNS service that you can configure to:

- Map instance IPs and Walrus bucket names to DNS host names
- Enable DNS delegation to support transparent failover in HA mode

The DNS service will automatically try to bind to port 53. If port 53 cannot be used, DNS will be disabled. Typically, other system services like dnsmasq are configured to run on port 53. To use the Eucalyptus DNS service, you will need to disable these services.

Configure the Subdomain

Before using the DNS service, configure the DNS sub domain name that you want Eucalyptus to handle as follows after the Eucalyptus Cloud Controller (CLC) has been started.

Log in to the CLC (the primary CLC in an HA setup) and enter the following:

```
euca-modify-property -p
system.dns.dnsdomain=<eucadomain.yourdomain>
```

Turn on IP Mapping

To turn on mapping of instance IPs to DNS host names:

Enter the following command:

```
euca-modify-property -p bootstrap.webservices.use_instance_dns=true
```

When this option is enabled, public and private DNS entries are set up for each instance that is launched in Eucalyptus. This also enables virtual hosting for Walrus. Buckets created in Walrus can be accessed as hosts. For example, the bucket `mybucket` is accessible as `mybucket.walrus.eucadomain.yourdomain`.

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

Enable DNS Delegation

For HA: If you are not using HA, you can skip this task.

DNS delegation allows you to forward DNS traffic for the Eucalyptus subdomain to the Eucalyptus CLC hosts. These hosts act as name servers. This allows interruption-free access to Eucalyptus cloud services in the event of a failure. Both primary and secondary CLC hosts are capable of mapping cloud host names to IP addresses of the primary CLC and Walrus hosts.

For example, if the IP address of the primary and secondary CLC are `192.168.5.1` and `192.168.5.2`, and the IP addresses of primary and secondary Walruses are `192.168.6.1` and `192.168.6.2`, the host `eucalyptus.eucadomain.yourdomain` will resolve to `192.168.6.1` and `walrus.eucadomain.yourdomain` will resolve to `192.168.6.1`.

If the primary CLC fails, the secondary CLC will become the primary and `eucalyptus.eucadomain.yourdomain` will resolve to `192.168.5.2`. If the primary Walrus fails, the secondary Walrus will be promoted and `walrus.eucadomain.yourdomain` will resolve to `192.168.6.2`.

To enable DNS delegation:

On the primary CLC, enter the following command:

```
euca-modify-property -p bootstrap.webservices.use_dns_delegation=true
```

Configure the Master DNS Server

Set up your master DNS server to forward the Eucalyptus subdomain to the primary and secondary CLC servers, which act as name servers.

The following example shows how the Linux name server `bind` is set up to forward the Eucalyptus subdomain.

1. Open `/etc/named.conf` and set up the `eucadomain.yourdomain zone`. For example, your `/etc/named.conf` may look like the following:

```
zone "yourdomain" {
type master;
file "/etc/bind/db.yourdomain";
};

#Forward eucadomain.yourdomain
zone "eucadomain.yourdomain" {
type forward;
forward only;
forwarders { <CLC_0_IP>; <CLC_1_IP>; };
};
```

   where `<CLC_0_IP>` is the IP address of your primary CLC and `<CLC_1_IP>` is the IP address of your secondary CLC.

2. Create `/etc/bind/db.yourdomain` if it does not exist. If your master DNS is already set up for `yourdomain`, you will need to add name server entries for `<CLC_0_IP>` and `<CLC_1_IP>`. For example:

```
$TTL 604800
@ IN SOA yourdomain. root.yourdomain. (
2 ; Serial
604800 ; Refresh
86400 ; Retry
2419200 ; Expire
604800 ) ; Negative Cache TTL
;
@ IN NS ns.yourdomain.
@ IN A <master_nameserver_IP>

ns.yourdomain. IN A <master_nameserver_IP>

;Add entries for primary and secondary CLCs
eucadomain.yourdomain. IN NS clc0.eucadomain.yourdomain.
eucadomain.yourdomain. IN NS clc1.eucadomain.yourdomain.

clc0.eucadomain.yourdomain. IN A <CLC_0_IP>
clc1.eucadomain.yourdomain. IN A <CLC_1_IP>
```

   where `clc0.eucadomain.yourdomain` and `clc1.eucadomain.yourdomain` are the host names of your primary and secondary CLC servers.

3. Restart the bind nameserver (`/etc/init.d/bind9 restart` or `/etc/init.d/named restart`, depending on your Linux distribution).
4. Test your setup by pointing `/etc/resolv.conf` on your client to your primary DNS server and attempt to resolve `eucalyptus.eucadomain.yourdomain` using ping or nslookup. It should return the IP address of the primary CLC server.

Set NC Concurrency Level

On some Linux installations, a sufficiently large amount of local disk activity can slow down process scheduling. This can cause other operations (e.g., network communication and instance provisioning) appear to stall. To alleviate this potential problem, we recommend performing the following steps on each NC:

1. Log in to an NC server and open the `/etc/eucalyptus/eucalyptus.conf` file.

2. Change the `CONCURRENT_DISK_OPS` parameter to the number of disk-intensive operations you want the NC to perform at once. Examples of disk-intensive operations include preparing disk images for launch and creating ephemeral storage. Set this value to `1` to serialize all disk-intensive operations. Set to a higher number to increase the amount of disk-intensive operations the NC will perform in parallel.

Increase Walrus Disk Space

The size of Walrus storage must be larger than the sum of all the uploaded images. Each uploaded image requires additional space to accommodate image decryption and the creation of temporary working files. **We recommend that the Walrus storage size be three times the size of all uploaded images.**

For example, you might 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 ready to run in Eucalyptus, you will need to set the "Space reserved for unbundling images" in Walrus to 50 GB or larger. To increase the image cache size in Walrus:

1. Log in to the Dashboard (https://<CLC_IP_address>:8443).
2. Click **Service Components** in the **Quick Links** section.
   The **Service Components** page displays.
3. Click **walrus**.
   The **Properties** section displays.
4. Enter the new size (in MB) in the **Space reserved for unbundling images** field.
5. Click **Save**.

Configure DRBD

For HA: This section is for Eucalyptus HA. If you are not using HA, skip this section.

Before you begin, ensure that you have the following information:

- The IP address and hostname of each Walrus
- The DRBD block device name of each Walrus. In the following examples, we assume that DRBD block device name is `/dev/drbd1`.
- The DRBD backing disk partition names on each Walrus. A partition (either on a new disk or an existing disk) should be dedicated to Walrus. The partition sizes should be identical.

Configuring DRBD requires that you complete edit the Eucalyptus DRBD file to include your Walrus information, and edit the master DRBD file to tell it to look for the Eucalyptus DRBD file.

To configure DRBD:

1. Log in to the primary Walrus.
2. Load the DRBD module

```
modprobe drbd
```

   There is no output from this command.

3. Copy the example Eucalyptus DRBD file (`/etc/eucalyptus/drbd.conf.example`) to `/etc/eucalyptus/drbd.conf`.
4. Open the `/etc/eucalyptus/drbd.conf` file and make the following edits:
   - Change the value of `<walrus-host-1>` to the hostname of the primary Walrus.
   - Change the value of `<drbd-block-dev, e.g., /dev/drbd1>` to `/dev/drbd1`
   - Change the value of `<drbd-backing-disk-dev, e.g. /dev/sdb1>` to `/dev/sdb1`
   - Change the value of `<walrus-host-1-ip>` to the IP address of the primary Walrus.

- Change the value of `<walrus-host-2>` to the hostname of the secondary Walrus.
- Change the value of `<drbd-block-dev, e.g., /dev/drbd1>` to `/dev/drbd1`
- Change the value of `<drbd-backing-disk-dev, e.g. /dev/sdb1>` to `/dev/sdb1`
- Change the value of `<walrus-host-2-ip>` to the IP address of the secondary Walrus.

The file should look like the following example:

```
common {
  protocol C;
}

resource r0 {

  on walrus00.eucalyptus.com {
    device    /dev/drbd1;
    disk      /dev/sdb1;
    address   192.168.58.1:7789;
    meta-disk internal;
  }

  on walrus01.eucalyptus.com {
    device    /dev/drbd1;
    disk      /dev/sdb1;
    address   192.168.58.2:7789;
    meta-disk internal;
  }

  syncer {
    rate 40M;
  }

  net {
    after-sb-0pri discard-zero-changes;
    after-sb-1pri discard-secondary;
  }
}
```

Important: On RHEL 6, remove the `common` section (`common { protocol C; }`). The default configuration in RHEL 6 already includes a common section.

5. Save and close the file.
6. Open the master DRBD file (`/etc/drbd.conf`) and append the following line:

```
include "/etc/eucalyptus/drbd.conf";
```

7. Save and close the file.
8. Open the `/etc/eucalyptus/eucalyptus.conf` file and make the following configuration:

```
CLOUD_OPTS="-Dwalrus.storage.manager=DRBDStorageManager"
```

9. Restart Walrus.

```
service eucalyptus-cloud restart
```

10. Copy the `/etc/drbd.conf`, the `/etc/eucalyptus/drbd.conf`, and the `/etc/eucalyptus/eucalyptus.conf` files to the secondary Walrus server.

11. On the primary Walrus, associate the DRBD block device (`/dev/drbd1`) with the disk partition allocated for Walrus (`/dev/sdb1`).

```
drbdmeta --force /dev/drbd1 v08 /dev/sdb1 internal create-md
drbdadm attach r0
drbdadm connect r0
```

    Repeat on the secondary Walrus.

12. Set up the DRBD block device on the primary Walrus:

```
drbdsetup /dev/drbd1 syncer -r 110M
drbdadm --overwrite-data-of-peer primary r0
```

13. On the primary Walrus only, run the following command to indicate whether the data on the DRBD primary and secondary is consistent:

```
drbdadm dstate r0
```

    Wait for the output to display `UpToDate/UpToDate`, then continue to the next step.

    Tip: To view the synchronization process in near-realtime, run `watch -n 2 cat /proc/drbd`.

14. On the primary Walrus, create a filesystem on `/dev/drbd1`. Eucalyptus supports ext3 or ext4. For example:

```
mkfs.ext3 /dev/drbd1
```

15. On the primary CLC, tell Eucalyptus to use DRBD parameters configured in the DRBD config file so Walrus can write to the correct device:

```
euca-modify-property -p walrus.blockdevice=/dev/drbd1
euca-modify-property -p walrus.resource=r0
```

Configure VMware Support

Configuring Eucalyptus to support VMware 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 Cluster Controller (CC).

To configure VMware support for a new install:

1. On the CC, enter the following command:

```
euca-configure-vmware
```

    The output of the above command prompts you for the same parameters that the vSphere client application requests at startup.

2. Enter the requested parameters, making sure to specify the full URL and not just the hostname. If you want to use vCenter, then enter the vCenter host's information as the only vSphere endpoint. If you do not want to use vCenter, then you must enter each ESX/ESXi host as an individual vSphere endpoint. We recommend using vCenter because it is easier to configure and can be more efficient.

```
Please, supply vSphere endpoint IP: 192.168.7.88
Please, supply vSphere username: Administrator
Please, supply vSphere password:
```

```
Do you want to enter another endpoint? [N]: Y
Please, supply vSphere endpoint IP: 192.168.7.89
Please, supply vSphere username: Administrator
Please, supply vSphere password:
Do you want to enter another endpoint? [N]: N
```

After entering all vSphere endpoint information you should see output similar to the following:

```
querying VMware endpoint at https://192.168.51.70/sdk...
detected apiType=HOSTAGENT apiVersion=4
reponse time=391ms
datacenter=ha-datacenter
detected 4 cores
detected 6552MB available for VMs and 6552MB currently unreserved for VMs
detected 151296MB of disk capacity
host=192.168.51.70 datastore=datastore1
Summary of hosts: 1 nodes with 4 total cores
discovered 1 host(s)
192.168.51.70 login=root datastoreName=datastore1 uploadViaHost=true"
```

3. Open the `/etc/eucalyptus/eucalyptus.conf` file and modify the following settings, making sure they are not commented out.
   a) In the `CLUSTER CONTROLLER (CC) / NODE CONTROLLER (NC) SHARED CONFIGURATION` section, set `NC_PORT="8773"`.
   b) In the `CLUSTER CONTROLLER (CC) CONFIGURATION` section, set `NC_SERVICE="/services/VMwareBroker"`.
4. Save the file.
5. Restart the CLC, the VMware Broker, and the CC.
   - For the CLC and the VMware Broker:

     ```
     service eucalyptus-cloud restart
     ```

   - For the CC:

     ```
     service eucalyptus-cc restart
     ```

Set Up Security Groups

In Managed and Managed (No VLAN) networking modes, you must configure the system with parameters that define how Eucalyptus will allocate and manage virtual machine networks. These virtual machine networks are known as security groups. The relevant parameters are set in the `eucalyptus.conf` on all machines running a CC. These parameters are:

- `VNET_SUBNET`
- `VNET_NETMASK`
- `VNET_ADDRSPERNET`

The CC will read `VNET_SUBNET` and `VNET_NETMASK` to construct a range of IP addresses that are available to all security groups. This range will then be further divided into smaller networks of the size specified in `VNET_ADDRSPERNET`.

The first time an instance runs in a given security group, Eucalyptus chooses an unused range of IPs of size specified in `VNET_ADDRSPERNET`. Eucalyptus then implements this network across all CCs. All instances that run within this given security group obtain a specific IP from this range.

Tip: Ten of the IP addresses within each security group network are reserved for Eucalyptus to use as gateway addresses, broadcast address, etc. For example, if you set `VNET_ADDRSPERNET` to `32`, there will be 22 free IPs that are available for instances running in that security group.

In Managed mode, each security group network is assigned an additional parameter that is used as the VLAN tag. This parameter is added to all virtual machine traffic running within the security group. By default, Eucalyptus uses VLAN tags starting at 2, going to a maximum of 4094. The maximum is dependent on how many security group networks of the size specified in `VNET_ADDRSPERNET` fit in the network defined by `VNET_SUBNET` and `VNET_NETMASK`.

If your networking environment is already using VLANs for other reasons, Eucalyptus supports the definition of a smaller range of VLANs that are available to Eucalyptus. To set this range with a running and configured Eucalyptus installation:

1. Determine the range that your cluster controllers are configured to support.

```
euca-describe-properties | grep cluster.maxnetworktag
euca-describe-properties | grep cluster.minnetworktag
```

2. Define a range that is a proper subset of the above bounds.

```
euca-modify-property -p cloud.network.global_max_network_tag=<max_vlan_tag>
euca-modify-property -p cloud.network.global_min_network_tag=<min_vlan_tag>
```

Finding More Information

Eucalyptus has the following guides to help you with more information:

- The *Administration Guide* details ways to manage your Eucalyptus deployment. Refer to this guide to learn more about managing your Eucalyptus components, managing access to Eucalyptus, and managing Eucalyptus resources, like instances and images.
- The *Usage Guide* details ways to use Eucalyptus for your computing and storage needs. Refer to this guide to learn more about getting and using euca2ools, creating images, running instances, and using dynamic block storage devices.

Appendix: Upgrading Eucalyptus

If you are upgrading from a previous version of Eucalyptus, follow the directions detailed in this section.

Migrating Users

There are two considerations for migrating users during the upgrade process:

- In Eucalyptus 3.0.2 the concept of **user** has changed. What used to be a user is now an **account**, and the user is an identity within the account. In the upgrade process to Eucalyptus 3.0.2, users are converted to accounts.
- Eucalyptus 3.0.2 enforces case insensitivity. So a user labeled **john** and another user labeled **JOHN** will collide during the upgrade process. You must either relabel conflicting accounts or be aware that Eucalyptus will relabel these accounts. For example, the upgrade process will maintain the **john** label but will relabel the other one as **john-**.

Migrating VLAN Range

In Eucalyptus 2.x, there were per-cluster settings in the web user interface for **Min VLANs** and **Max VLANs**. These values defined the range of numeric VLAN tags allowed for that cluster. In some cases, this was necessary due to limitations of the network switches being used. Failing to preserve these values might break your cloud's network configuration.

In Eucalyptus 3.0.2, the VLAN tag range is a global setting. The safest thing is to choose a range that is the intersection of the user's per-cluster settings from 2.x. However, this will not always be the preferred configuration for each user.

Use the `euca-describe-properties` command to display the new settings:

```
PROPERTY    cloud.network.global_max_network_tag    1000
PROPERTY    cloud.network.global_min_network_tag    2
```

Finding Backup Files

The upgrade process creates a backup to `/var/lib/eucalyptus/upgrade/eucalyptus.backup.<timestamp>`. For example:

```
/var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905212
```

If the upgrade fails and needs to be reverted to 2.x, you can find your preserved data in this directory.

CentOS 5

This section explains tasks to perform in order to upgrade to Eucalyptus 3.0.2 on CentOS 5 machines.

Prepare the Configuration File

To upgrade to Eucalyptus 3.0.2 on CentOS:

1. Move your credentials so the package managers can locate them.
   a) Place the entitlement certificate in `/etc/pki/tls/certs`.

   ```
   mv <license_name>-1.3.0.crt /etc/pki/tls/certs/<license_name>-1.3.0.crt
   ```

   b) Place the private key in `/etc/pki/tls/private`.

   ```
   mv <license_name>.key /etc/pki/tls/private/<license_name>.key
   ```

c) Place the Eucalyptus GPG key in `/etc/pki/rpm-gpg`.

```
mv c1240596-eucalyptus-release-key.pub
/etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
```

2. On all hosts, create a file in `/etc/yum.repos.d` called eucalyptus-enterprise.repo with the following content:

```
[eucalyptus-enterprise]
name=Eucalyptus Enterprise 3.0
baseurl=https://downloads.eucalyptus.com/software/enterprise/3.0/centos/$releasever/$basearch
gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
gpgcheck=1
sslclientcert=/etc/pki/tls/certs/<license_name>-1.3.0.crt
sslclientkey=/etc/pki/tls/private/<license_name>.key
```

3. On all systems that will run either Eucalyptus or Euca2ools, create a file in `/etc/yum.repos.d` called euca2ools.repo with the following content:

```
[euca2ools]
name=Euca2ools 2.0
baseurl=http://downloads.eucalyptus.com/software/euca2ools/2.0/rhel/$releasever/$basearch
gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
gpgcheck=1
```

4. Enable the EPEL repository with the following two commands:

```
wget
http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm

rpm -Uvh epel-release-5-4.noarch.rpm
```

Shutdown Components

To shut down Eucalyptus components:

Tip:

The following is an example that scripts the individual steps noted in this section.

```
for x in $( euca_conf --list-nodes | tail -n +2 | awk '{ print $1
}');
do ssh root@$x "service eucalyptus-nc stop"; done
for x in $( euca_conf --list-clusters | tail -n +2 | awk '{ print $2
}'); do ssh root@$x "service eucalyptus-cc stop"; done
for x in $( euca_conf --list-scs | tail -n +2 | awk '{ print $2 }');
do ssh root@$x "service eucalyptus-cloud stop"; done
for x in $( euca_conf --list-walruses | tail -n +2 | awk '{ print $2
}'); do ssh root@$x "service eucalyptus-cloud stop"; done
service eucalyptus-cloud stop
```

If you don't have ssh keys set up, you'll have to type a lot of passwords during this.

1. Terminate any running instances, as in the following example:

```
euca-terminate-instances <instance01_id> <instance02_id>
```

2. Log in to an NC host and shut down the NC service.

```
service eucalyptus-nc stop
```

   Repeat for each machine hosting an NC.

3. Log in to a CC host and shut down the CC service.

```
service eucalyptus-cc cleanstop
```

   Repeat for each machine hosting a CC.

4. Shut down the VMware Broker service on the CC host.

```
service eucalyptus-cloud stop
```

   Tip: This command also shuts down a CLC, Walrus, and SC components co-located with the CC and VMware Broker to stop at the same time, in the correct order.

   Repeat for each machine hosting the VMware Broker.

5. Log in to an SC host and shut down the SC service.

```
service eucalyptus-cloud stop
```

   Repeat for any other machine hosting an SC.

6. Log in to the Walrus host and shut down the Walrus service.

```
service eucalyptus-cloud stop
```

7. Log in to the CLC host and shut down the CLC service.

```
service eucalyptus-cloud stop
```

Upgrade Eucalyptus Packages

Use

```
yum update
```

to upgrade the Eucalyptus packages.

If you have previously customized your configuration files,

```
yum
```

will emit a warning, and install the new configuration files with a different name, to preserve your customizations. You should customize the new configuration files as necessary, and then rename these files, before proceeding.

Tip: For larger deployments, use a script to upgrade the component host machines. For example:

```
for host in 28 29 32 33 35 39 40; do echo 192.168.51.$host; ssh
192.168.51.$host 'yum update -y $( rpm -qa | grep euca )' ; done
```

Start Eucalyptus

1. In the CLC, enter the following command.

```
service eucalyptus-cloud start
```

The process starts the database upgrade. Eucalyptus returns output similar to the following example.

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326904600...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false             using:
# Upgrade configuration:    false             using:
# Upgrade database:         true              using: upgrade_db
# Same version:             false             using:
# Start upgrading: db
Upgrading your database...
.
.
.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

Note: You might see some warnings in the output. These are a known issue.

2. Log in to the Walrus server and enter the following command:

```
service eucalyptus-cloud start
```

Eucalyptus returns output similar to the following example.

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905005...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false             using:
# Upgrade configuration:    false             using:
# Upgrade database:         true              using: upgrade_db
# Same version:             false             using:
# Start upgrading: db
CLC is disabled.  Skipping DB upgrade.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

3. Log in to the CC server and enter the following:

```
service eucalyptus-cc start
```

4. If you have a multi-cluster setup, repeat the previous step for each cluster.
5. If you are using Eucalyptus with VMware support, start the VMware Broker. On the CC server, and enter the following:

```
service eucalyptus-cloud start
```

6. Repeat for each CC server.
7. Log in to the SC server and enter the following command:

```
service eucalyptus-cloud start
```

Eucalyptus returns output similar to the following example>

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905005...
#                           UPGRADE INFORMATION
#===============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false              using:
# Upgrade configuration:    false              using:
# Upgrade database:         true               using: upgrade_db
# Same version:             false              using:
# Start upgrading: db
CLC is disabled.  Skipping DB upgrade.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

8. Log in to an NC server and enter the following command:

```
service eucalyptus-nc start
```

9. Repeat for each NC server.

Verify the Components

Verify that all Eucalyptus components are running and properly connected to one another. Check to make sure that the status of each component is enabled.

To verify that all services are enabled:

1. Verify your Walruses:

```
euca_conf --list-walruses
```

Eucalyptus returns a list, as in the following example.

```
WALRUS  walrus     walrus     192.168.51.28     ENABLED {}
```

2. Verify your CCs:

```
euca_conf --list-clusters
```

Eucalyptus returns a list, as in the following example.

```
CLUSTER test00     test00_cc     192.168.51.29     ENABLED {}
CLUSTER test01     test01_cc     192.168.51.35     ENABLED {}
```

3. Verify your SCs:

```
euca_conf --list-scs
```

Eucalyptus returns a list, as in the following example.

```
STORAGECONTROLLER     test01     test01_sc     192.168.51.39      ENABLED {}
STORAGECONTROLLER     test00     test00_sc     192.168.51.32      ENABLED {}
```

4. Make sure that NCs are presenting available resources to the CC.

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

The returned output should a non-zero number in the `free` and `max` columns, as in the following example.

```
AVAILABILITYZONE        test00  192.168.51.29
arn:euca:eucalyptus:test00:cluster:test00_cc/
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small     0004 / 0004   1    128     2
AVAILABILITYZONE        |- c1.medium    0004 / 0004   1    256     5
AVAILABILITYZONE        |- m1.large     0002 / 0002   2    512    10
AVAILABILITYZONE        |- m1.xlarge    0002 / 0002   2   1024    20
AVAILABILITYZONE        |- c1.xlarge    0001 / 0001   4   2048    20
AVAILABILITYZONE        test01  192.168.51.35
arn:euca:eucalyptus:test01:cluster:test01_cc/
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small     0004 / 0004   1    128     2
AVAILABILITYZONE        |- c1.medium    0004 / 0004   1    256     5
AVAILABILITYZONE        |- m1.large     0002 / 0002   2    512    10
AVAILABILITYZONE        |- m1.xlarge    0002 / 0002   2   1024    20
AVAILABILITYZONE        |- c1.xlarge    0001 / 0001   4   2048    20
```

Upgrade Credentials

All users' credentials will still work after the upgrade. However the new Eucalyptus access control commands will not work until you upgrade your credentials. Other users must updates theirs as well.

To update your credentials:

1. Enter the following command:

```
euca_conf --get-credentials <filename>
```

2. Unzip the file

```
unzip -o <filename>
```

RHEL 5

This section explains tasks to perform in order to upgrade to Eucalyptus 3.0.2 on RHEL 5 machines.

Prepare the Configuration File

To update the distro in RHEL 5:

1. Move your credentials so the package managers can locate them.
   a) Place the entitlement certificate in `/etc/pki/tls/certs`.

```
mv <license_name>-1.3.0.crt /etc/pki/tls/certs/<license_name>-1.3.0.crt
```

   b) Place the private key in `/etc/pki/tls/private`.

```
mv <license_name>.key /etc/pki/tls/private/<license_name>.key
```

   c) Place the Eucalyptus GPG key in `/etc/pki/rpm-gpg`.

```
mv c1240596-eucalyptus-release-key.pub
/etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
```

2. On all servers, create a file in `/etc/yum.repos.d` called `eucalyptus-enterprise.repo` with the following content:

```
[eucalyptus-enterprise]
name=Eucalyptus Enterprise 3.0
baseurl=https://downloads.eucalyptus.com/software/enterprise/3.0/rhel/$releasever/$basearch
gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
gpgcheck=1
sslclientcert=/etc/pki/tls/certs/<license_name>-1.3.0.crt
sslclientkey=/etc/pki/tls/private/<license_name>.key
```

3. On all systems that will run either Eucalyptus or Euca2ools, create a file in `/etc/yum.repos.d` called `euca2ools.repo` with the following content:

```
[euca2ools]
name=Euca2ools 2.0
baseurl=http://downloads.eucalyptus.com/software/euca2ools/2.0/rhel/$releasever/$basearch
gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
gpgcheck=1
```

Shutdown Components

To shut down Eucalyptus components:

Tip:

The following is an example that scripts the individual steps noted in this section.

```
for x in $( euca_conf --list-nodes | tail -n +2 | awk '{ print $1
}');
do ssh root@$x "service eucalyptus-nc stop"; done
for x in $( euca_conf --list-clusters | tail -n +2 | awk '{ print $2
}'); do ssh root@$x "service eucalyptus-cc stop"; done
for x in $( euca_conf --list-scs | tail -n +2 | awk '{ print $2 }');
do ssh root@$x "service eucalyptus-cloud stop"; done
for x in $( euca_conf --list-walruses | tail -n +2 | awk '{ print $2
}'); do ssh root@$x "service eucalyptus-cloud stop"; done
service eucalyptus-cloud stop
```

If you don't have ssh keys set up, you'll have to type a lot of passwords during this.

1. Terminate any running instances, as in the following example:

```
euca-terminate-instances <instance01_id> <instance02_id>
```

2. Log in to an NC host and shut down the NC service.

```
service eucalyptus-nc stop
```

   Repeat for each machine hosting an NC.

3. Log in to a CC host and shut down the CC service.

```
service eucalyptus-cc cleanstop
```

   Repeat for each machine hosting a CC.

4. Shut down the VMware Broker service on the CC host.

```
service eucalyptus-cloud stop
```

   Tip: This command also shuts down a CLC, Walrus, and SC components co-located with the CC and VMware Broker to stop at the same time, in the correct order.

   Repeat for each machine hosting the VMware Broker.

5. Log in to an SC host and shut down the SC service.

```
service eucalyptus-cloud stop
```

   Repeat for any other machine hosting an SC.

6. Log in to the Walrus host and shut down the Walrus service.

```
service eucalyptus-cloud stop
```

7. Log in to the CLC host and shut down the CLC service.

```
service eucalyptus-cloud stop
```

Upgrade Eucalyptus Packages

Use

```
yum update
```

to upgrade the Eucalyptus packages.

If you have previously customized your configuration files,

```
yum
```

will emit a warning, and install the new configuration files with a different name, to preserve your customizations. You should customize the new configuration files as necessary, and then rename these files, before proceeding.

Tip: For larger deployments, use a script to upgrade the component host machines. For example:

```
for host in 28 29 32 33 35 39 40; do echo 192.168.51.$host; ssh
192.168.51.$host 'yum update -y $( rpm -qa | grep euca )' ; done
```

Start Eucalyptus

1. In the CLC, enter the following command.

```
service eucalyptus-cloud start
```

The process starts the database upgrade. Eucalyptus returns output similar to the following example.

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326904600...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false             using:
# Upgrade configuration:    false             using:
# Upgrade database:         true              using: upgrade_db
# Same version:             false             using:
# Start upgrading: db
Upgrading your database...
.
.
.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

Note: You might see some warnings in the output. These are a known issue.

2. Log in to the Walrus server and enter the following command:

```
service eucalyptus-cloud start
```

Eucalyptus returns output similar to the following example.

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905005...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false             using:
# Upgrade configuration:    false             using:
# Upgrade database:         true              using: upgrade_db
# Same version:             false             using:
# Start upgrading: db
CLC is disabled.  Skipping DB upgrade.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
```

```
and stderr to //var/log/eucalyptus/startup.log
done.
```

3. Log in to the CC server and enter the following:

```
service eucalyptus-cc start
```

4. If you have a multi-cluster setup, repeat the previous step for each cluster.
5. If you are using Eucalyptus with VMware support, start the VMware Broker. On the CC server, and enter the following:

```
service eucalyptus-cloud start
```

6. Repeat for each CC server.
7. Log in to the SC server and enter the following command:

```
service eucalyptus-cloud start
```

   Eucalyptus returns output similar to the following example>

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905005...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false              using:
# Upgrade configuration:    false              using:
# Upgrade database:         true               using: upgrade_db
# Same version:             false              using:
# Start upgrading: db
CLC is disabled.  Skipping DB upgrade.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

8. Log in to an NC server and enter the following command:

```
service eucalyptus-nc start
```

9. Repeat for each NC server.

Verify the Components

Verify that all Eucalyptus components are running and properly connected to one another. Check to make sure that the status of each component is enabled.

To verify that all services are enabled:

1. Verify your Walruses:

```
euca_conf --list-walruses
```

Eucalyptus returns a list, as in the following example.

```
WALRUS  walrus     walrus     192.168.51.28     ENABLED {}
```

2. Verify your CCs:

```
euca_conf --list-clusters
```

Eucalyptus returns a list, as in the following example.

```
CLUSTER test00     test00_cc     192.168.51.29     ENABLED {}
CLUSTER test01     test01_cc     192.168.51.35     ENABLED {}
```

3. Verify your SCs:

```
euca_conf --list-scs
```

Eucalyptus returns a list, as in the following example.

```
STORAGECONTROLLER     test01     test01_sc     192.168.51.39      ENABLED {}
STORAGECONTROLLER     test00     test00_sc     192.168.51.32      ENABLED {}
```

4. Make sure that NCs are presenting available resources to the CC.

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

The returned output should a non-zero number in the `free` and `max` columns, as in the following example.

```
AVAILABILITYZONE        test00  192.168.51.29
arn:euca:eucalyptus:test00:cluster:test00_cc/
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small     0004 / 0004   1    128     2
AVAILABILITYZONE        |- c1.medium    0004 / 0004   1    256     5
AVAILABILITYZONE        |- m1.large     0002 / 0002   2    512    10
AVAILABILITYZONE        |- m1.xlarge    0002 / 0002   2   1024    20
AVAILABILITYZONE        |- c1.xlarge    0001 / 0001   4   2048    20
AVAILABILITYZONE        test01  192.168.51.35
arn:euca:eucalyptus:test01:cluster:test01_cc/
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small     0004 / 0004   1    128     2
AVAILABILITYZONE        |- c1.medium    0004 / 0004   1    256     5
AVAILABILITYZONE        |- m1.large     0002 / 0002   2    512    10
AVAILABILITYZONE        |- m1.xlarge    0002 / 0002   2   1024    20
AVAILABILITYZONE        |- c1.xlarge    0001 / 0001   4   2048    20
```

Upgrade Credentials

All users' credentials will still work after the upgrade. However the new Eucalyptus access control commands will not work until you upgrade your credentials. Other users must updates theirs as well.

To update your credentials:

1. Enter the following command:

```
euca_conf --get-credentials <filename>
```

2. Unzip the file

```
unzip -o <filename>
```

RHEL 6

This section explains tasks to perform in order to upgrade to Eucalyptus 3.0.2 on RHEL 6 machines.

Prepare the Configuration File

To update the distro in RHEL 6:

1. Move your credentials so the package managers can locate them.
   a) Place the entitlement certificate in `/etc/pki/tls/certs`.

```
mv <license_name>-1.3.0.crt /etc/pki/tls/certs/<license_name>-1.3.0.crt
```

   b) Place the private key in `/etc/pki/tls/private`.

```
mv <license_name>.key /etc/pki/tls/private/<license_name>.key
```

   c) Place the Eucalyptus GPG key in `/etc/pki/rpm-gpg`.

```
mv c1240596-eucalyptus-release-key.pub
/etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
```

2. On all servers, create a file in `/etc/yum.repos.d` called `eucalyptus-enterprise.repo` with the following content:

```
[eucalyptus-enterprise]
name=Eucalyptus Enterprise 3.0
baseurl=https://downloads.eucalyptus.com/software/enterprise/3.0/rhel/$releasever/$basearch
gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
gpgcheck=1
sslclientcert=/etc/pki/tls/certs/<license_name>-1.3.0.crt
sslclientkey=/etc/pki/tls/private/<license_name>.key
```

3. On all servers that will run either Eucalyptus or Euca2ools, create a file in `/etc/yum.repos.d` called `euca2ools.repo` with the following content:

```
[euca2ools]
name=Euca2ools 2.0
baseurl=http://downloads.eucalyptus.com/software/euca2ools/2.0/rhel/$releasever/$basearch
gpgkey=file:///etc/pki/rpm-gpg/c1240596-eucalyptus-release-key.pub
gpgcheck=1
```

Shutdown Components

To shut down Eucalyptus components:

Tip:

The following is an example that scripts the individual steps noted in this section.

```
for x in $( euca_conf --list-nodes | tail -n +2 | awk '{ print $1
}');
do ssh root@$x "service eucalyptus-nc stop"; done
for x in $( euca_conf --list-clusters | tail -n +2 | awk '{ print $2
}'); do ssh root@$x "service eucalyptus-cc stop"; done
for x in $( euca_conf --list-scs | tail -n +2 | awk '{ print $2 }');
do ssh root@$x "service eucalyptus-cloud stop"; done
for x in $( euca_conf --list-walruses | tail -n +2 | awk '{ print $2
}'); do ssh root@$x "service eucalyptus-cloud stop"; done
service eucalyptus-cloud stop
```

If you don't have ssh keys set up, you'll have to type a lot of passwords during this.

1. Terminate any running instances, as in the following example:

```
euca-terminate-instances <instance01_id> <instance02_id>
```

2. Log in to an NC host and shut down the NC service.

```
service eucalyptus-nc stop
```

   Repeat for each machine hosting an NC.

3. Log in to a CC host and shut down the CC service.

```
service eucalyptus-cc cleanstop
```

   Repeat for each machine hosting a CC.

4. Shut down the VMware Broker service on the CC host.

```
service eucalyptus-cloud stop
```

   Tip: This command also shuts down a CLC, Walrus, and SC components co-located with the CC and VMware Broker to stop at the same time, in the correct order.

   Repeat for each machine hosting the VMware Broker.

5. Log in to an SC host and shut down the SC service.

```
service eucalyptus-cloud stop
```

   Repeat for any other machine hosting an SC.

6. Log in to the Walrus host and shut down the Walrus service.

```
service eucalyptus-cloud stop
```

7. Log in to the CLC host and shut down the CLC service.

```
service eucalyptus-cloud stop
```

Upgrade Eucalyptus Packages

Use

```
yum update
```

to upgrade the Eucalyptus packages.

If you have previously customized your configuration files,

```
yum
```

will emit a warning, and install the new configuration files with a different name, to preserve your customizations. You should customize the new configuration files as necessary, and then rename these files, before proceeding.

Tip: For larger deployments, use a script to upgrade the component host machines. For example:

```
for host in 28 29 32 33 35 39 40; do echo 192.168.51.$host; ssh
192.168.51.$host 'yum update -y $( rpm -qa | grep euca )' ; done
```

Start Eucalyptus

1. In the CLC, enter the following command.

```
service eucalyptus-cloud start
```

The process starts the database upgrade. Eucalyptus returns output similar to the following example.

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326904600...
#                          UPGRADE INFORMATION
#=============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false             using:
# Upgrade configuration:    false             using:
# Upgrade database:         true              using: upgrade_db
# Same version:             false             using:
# Start upgrading: db
Upgrading your database...
.
.
.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

Note: You might see some warnings in the output. These are a known issue.

2. Log in to the Walrus server and enter the following command:

```
service eucalyptus-cloud start
```

Eucalyptus returns output similar to the following example.

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905005...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false             using:
# Upgrade configuration:    false             using:
# Upgrade database:         true              using: upgrade_db
# Same version:             false             using:
# Start upgrading: db
CLC is disabled.  Skipping DB upgrade.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

3. Log in to the CC server and enter the following:

```
service eucalyptus-cc start
```

4. If you have a multi-cluster setup, repeat the previous step for each cluster.
5. If you are using Eucalyptus with VMware support, start the VMware Broker. On the CC server, and enter the following:

```
service eucalyptus-cloud start
```

6. Repeat for each CC server.
7. Log in to the SC server and enter the following command:

```
service eucalyptus-cloud start
```

Eucalyptus returns output similar to the following example>

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905005...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false             using:
# Upgrade configuration:    false             using:
# Upgrade database:         true              using: upgrade_db
# Same version:             false             using:
# Start upgrading: db
CLC is disabled.  Skipping DB upgrade.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

8. Log in to an NC server and enter the following command:

```
service eucalyptus-nc start
```

9. Repeat for each NC server.

Verify the Components

Verify that all Eucalyptus components are running and properly connected to one another. Check to make sure that the status of each component is enabled.

To verify that all services are enabled:

1. Verify your Walruses:

```
euca_conf --list-walruses
```

Eucalyptus returns a list, as in the following example.

```
WALRUS  walrus     walrus     192.168.51.28     ENABLED {}
```

2. Verify your CCs:

```
euca_conf --list-clusters
```

Eucalyptus returns a list, as in the following example.

```
CLUSTER test00     test00_cc     192.168.51.29     ENABLED {}
CLUSTER test01     test01_cc     192.168.51.35     ENABLED {}
```

3. Verify your SCs:

```
euca_conf --list-scs
```

Eucalyptus returns a list, as in the following example.

```
STORAGECONTROLLER     test01     test01_sc     192.168.51.39      ENABLED {}
STORAGECONTROLLER     test00     test00_sc     192.168.51.32      ENABLED {}
```

4. Make sure that NCs are presenting available resources to the CC.

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

The returned output should a non-zero number in the `free` and `max` columns, as in the following example.

```
AVAILABILITYZONE        test00  192.168.51.29
arn:euca:eucalyptus:test00:cluster:test00_cc/
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small     0004 / 0004   1    128     2
AVAILABILITYZONE        |- c1.medium    0004 / 0004   1    256     5
AVAILABILITYZONE        |- m1.large     0002 / 0002   2    512    10
AVAILABILITYZONE        |- m1.xlarge    0002 / 0002   2   1024    20
AVAILABILITYZONE        |- c1.xlarge    0001 / 0001   4   2048    20
AVAILABILITYZONE        test01  192.168.51.35
arn:euca:eucalyptus:test01:cluster:test01_cc/
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small     0004 / 0004   1    128     2
```

```
AVAILABILITYZONE        |- c1.medium    0004 / 0004   1    256     5
AVAILABILITYZONE        |- m1.large     0002 / 0002   2    512    10
AVAILABILITYZONE        |- m1.xlarge    0002 / 0002   2   1024    20
AVAILABILITYZONE        |- c1.xlarge    0001 / 0001   4   2048    20
```

Upgrade Credentials

All users' credentials will still work after the upgrade. However the new Eucalyptus access control commands will not work until you upgrade your credentials. Other users must updates theirs as well.

To update your credentials:

1. Enter the following command:

```
euca_conf --get-credentials <filename>
```

2. Unzip the file

```
unzip -o <filename>
```

Ubuntu 10.04 LTS

This section explains tasks to perform in order to upgrade to Eucalyptus 3.0.2 on Ubuntu 10.04 LTS machines.

Prepare the Configuration File

To update the distro in Ubuntu 10.04 LTS:

1. Copy the entitlement certificate to the `/etc/ssl/certs` directory on each server that you want to install Eucalyptus on.

```
mv <license_name>-1.3.0.crt /etc/ssl/certs/<license_name>-1.3.0.crt
```

2. Copy the private key file to the /etc/ssl/private directory on each server that you want to install Eucalyptus on.

```
mv <license_name>.key /etc/ssl/private/<license_name>.key
```

Important: Make sure that the private key's file permissions are restricted to only the root user and ssl-certs group.

3. Add the public key to the list of trusted keys:

```
apt-key add c1240596-eucalyptus-release-key.pub
```

4. On each server that you want to install Eucalyptus on, go to `/etc/apt/apt.conf.d` and create a new file (for example, `eucarepo`) with the following content:

```
Acquire {
https {
VerifyPeer "true";
SslCert "/etc/ssl/certs/<license_name>-1.3.0.crt";
SslKey "/etc/ssl/private/<license_name>.key";
};
};
```

5. Create a file in `/etc/apt/sources.list.d` called `eucalyptus-enterprise.list` with the following content:

```
deb https://downloads.eucalyptus.com/software/enterprise/3.0/ubuntu lucid
universe
```

6. On all machines that will run either Eucalyptus or Euca2ools, create a file in /etc/apt/sources.list.d called euca2ools.list with the following content:

```
deb http://downloads.eucalyptus.com/software/euca2ools/2.0/ubuntu lucid
universe
```

Shutdown Components

To shut down Eucalyptus components:

Tip:

The following is an example that scripts the individual steps noted in this section.

```
for x in $( euca_conf --list-nodes | tail -n +2 | awk '{ print $1
}');
do ssh root@$x "service eucalyptus-nc stop"; done
for x in $( euca_conf --list-clusters | tail -n +2 | awk '{ print $2
}'); do ssh root@$x "service eucalyptus-cc stop"; done
for x in $( euca_conf --list-scs | tail -n +2 | awk '{ print $2 }');
do ssh root@$x "service eucalyptus-cloud stop"; done
for x in $( euca_conf --list-walruses | tail -n +2 | awk '{ print $2
}'); do ssh root@$x "service eucalyptus-cloud stop"; done
service eucalyptus-cloud stop
```

If you don't have ssh keys set up, you'll have to type a lot of passwords during this.

1. Terminate any running instances, as in the following example:

```
euca-terminate-instances <instance01_id> <instance02_id>
```

2. Log in to an NC host and shut down the NC service.

```
service eucalyptus-nc stop
```

Repeat for each machine hosting an NC.

3. Log in to a CC host and shut down the CC service.

```
service eucalyptus-cc cleanstop
```

Repeat for each machine hosting a CC.

4. Shut down the VMware Broker service on the CC host.

```
service eucalyptus-cloud stop
```

Tip: This command also shuts down a CLC, Walrus, and SC components co-located with the CC and VMware Broker to stop at the same time, in the correct order.

Repeat for each machine hosting the VMware Broker.

5. Log in to an SC host and shut down the SC service.

```
service eucalyptus-cloud stop
```

   Repeat for any other machine hosting an SC.

6. Log in to the Walrus host and shut down the Walrus service.

```
service eucalyptus-cloud stop
```

7. Log in to the CLC host and shut down the CLC service.

```
service eucalyptus-cloud stop
```

Upgrade Eucalyptus Packages

Use

```
yum update
```

to upgrade the Eucalyptus packages.

If you have previously customized your configuration files,

```
yum
```

will emit a warning, and install the new configuration files with a different name, to preserve your customizations. You should customize the new configuration files as necessary, and then rename these files, before proceeding.

Tip: For larger deployments, use a script to upgrade the component host machines. For example:

```
for host in 28 29 32 33 35 39 40; do echo 192.168.51.$host; ssh
192.168.51.$host 'yum update -y $( rpm -qa | grep euca )' ; done
```

Start Eucalyptus

1. In the CLC, enter the following command.

```
service eucalyptus-cloud start
```

   The process starts the database upgrade. Eucalyptus returns output similar to the following example.

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326904600...
#                          UPGRADE INFORMATION
#==============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false             using:
# Upgrade configuration:    false             using:
# Upgrade database:         true              using: upgrade_db
# Same version:             false             using:
# Start upgrading: db
Upgrading your database...
.
.
.
```

```
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

Note: You might see some warnings in the output. These are a known issue.

2. Log in to the Walrus server and enter the following command:

```
service eucalyptus-cloud start
```

Eucalyptus returns output similar to the following example.

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905005...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false              using:
# Upgrade configuration:    false              using:
# Upgrade database:         true               using: upgrade_db
# Same version:             false              using:
# Start upgrading: db
CLC is disabled.  Skipping DB upgrade.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

3. Log in to the CC server and enter the following:

```
service eucalyptus-cc start
```

4. If you have a multi-cluster setup, repeat the previous step for each cluster.
5. If you are using Eucalyptus with VMware support, start the VMware Broker. On the CC server, and enter the following:

```
service eucalyptus-cloud start
```

6. Repeat for each CC server.
7. Log in to the SC server and enter the following command:

```
service eucalyptus-cloud start
```

Eucalyptus returns output similar to the following example>

```
Starting Eucalyptus services: Attempting database upgrade from 2.0.3
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905005...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              2.0.3
# New Version:              3.0.0
# Upgrade keys:             false              using:
```

```
# Upgrade configuration:    false             using:
# Upgrade database:         true              using: upgrade_db
# Same version:             false             using:
# Start upgrading: db
CLC is disabled.  Skipping DB upgrade.
# Done upgrading:  db
done.
[debug:0387] redirecting stdout to //var/log/eucalyptus/startup.log
and stderr to //var/log/eucalyptus/startup.log
done.
```

8. Log in to an NC server and enter the following command:

```
service eucalyptus-nc start
```

9. Repeat for each NC server.

Verify the Components

Verify that all Eucalyptus components are running and properly connected to one another. Check to make sure that the status of each component is enabled.

To verify that all services are enabled:

1. Verify your Walruses:

```
euca_conf --list-walruses
```

Eucalyptus returns a list, as in the following example.

```
WALRUS  walrus     walrus     192.168.51.28     ENABLED {}
```

2. Verify your CCs:

```
euca_conf --list-clusters
```

Eucalyptus returns a list, as in the following example.

```
CLUSTER test00     test00_cc     192.168.51.29     ENABLED {}
CLUSTER test01     test01_cc     192.168.51.35     ENABLED {}
```

3. Verify your SCs:

```
euca_conf --list-scs
```

Eucalyptus returns a list, as in the following example.

```
STORAGECONTROLLER     test01     test01_sc     192.168.51.39      ENABLED {}
STORAGECONTROLLER     test00     test00_sc     192.168.51.32      ENABLED {}
```

4. Make sure that NCs are presenting available resources to the CC.

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

The returned output should a non-zero number in the `free` and `max` columns, as in the following example.

```
AVAILABILITYZONE        test00  192.168.51.29
arn:euca:eucalyptus:test00:cluster:test00_cc/
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small     0004 / 0004   1    128     2
AVAILABILITYZONE        |- c1.medium    0004 / 0004   1    256     5
AVAILABILITYZONE        |- m1.large     0002 / 0002   2    512    10
AVAILABILITYZONE        |- m1.xlarge    0002 / 0002   2   1024    20
AVAILABILITYZONE        |- c1.xlarge    0001 / 0001   4   2048    20
AVAILABILITYZONE        test01  192.168.51.35
arn:euca:eucalyptus:test01:cluster:test01_cc/
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small     0004 / 0004   1    128     2
AVAILABILITYZONE        |- c1.medium    0004 / 0004   1    256     5
AVAILABILITYZONE        |- m1.large     0002 / 0002   2    512    10
AVAILABILITYZONE        |- m1.xlarge    0002 / 0002   2   1024    20
AVAILABILITYZONE        |- c1.xlarge    0001 / 0001   4   2048    20
```

Upgrade Credentials

All users' credentials will still work after the upgrade. However the new Eucalyptus access control commands will not work until you upgrade your credentials. Other users must updates theirs as well.

To update your credentials:

1. Enter the following command:

```
euca_conf --get-credentials <filename>
```

2. Unzip the file

```
unzip -o <filename>
```