Eucalyptus 3.3.2 Installation Guide

2013-09-27 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.

Guide version: Build 1873 (2013-09-27 09:59:39)

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.

Important: If you are upgrading from a previous version of Eucalyptus, see *Appendix A: 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?	Relevant 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 6
- Red Hat Enterprise Linux 6

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, IAM, ELB, Auto Scaling, and CloudWatch services. This offers you the capability of a hybrid cloud.

Eucalyptus Components

Eucalyptus is comprised of six components: Cloud Controller (CLC), Walrus, Cluster Controller (CC), Storage Controller (SC), Node Controller (NC) and an optional VMware Broker (Broker or VB). 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 Eucalyptus Administrator Console.

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 NCs and schedules virtual machine (VM) execution on specific NCs. The CC also manages the virtual machine networks. All NCs 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. 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 or VB) is an optional Eucalyptus component, which is available if you are a Eucalyptus subscriber. VMware Broker enables Eucalyptus to deploy virtual machines (VMs) on VMware infrastructure elements. VMware 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 6 and RHEL 6. Eucalyptus only supports 64-bit architecture.
- 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 6 and RHEL 6 installations must have KVM installed and configured on NC host machines. When you install Eucalyptus from packages, KVM will be installed on all NCs.
    - For information about using KVM on CentOS 6, go to the *Virtualization* page.
    - For more information about using KVM on RHEL 6, go to the *Virtualization* page in the Red Hat documentation.
    - VMware-based installations do not include NCs, but must have a VMware hypervisor pool installed and configured (VMware versions 4.0, 4.1 and 5.0).
- 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.

- Depending on some configurations, 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 (except the CC and NC) 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 A: Upgrading Eucalyptus*.

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

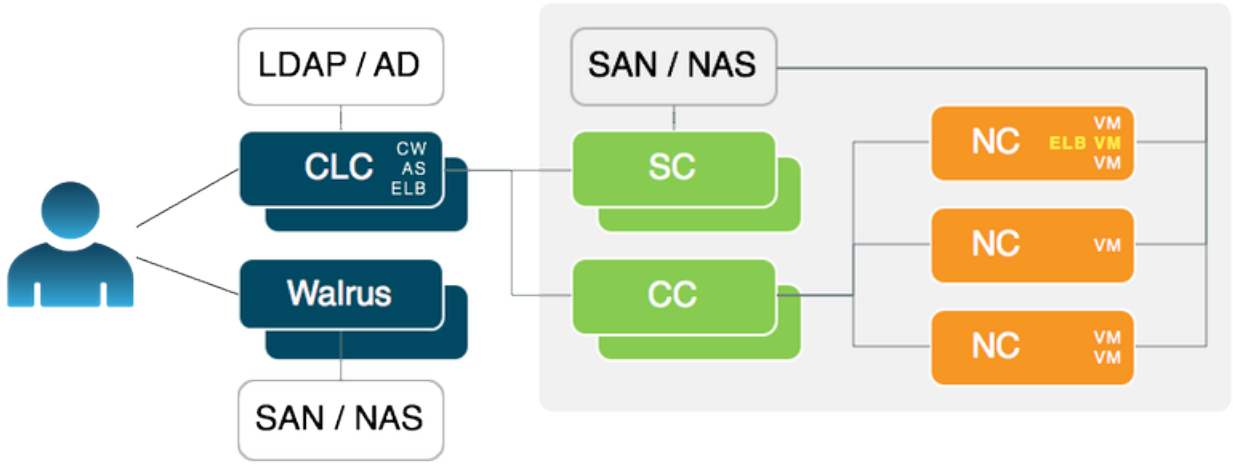
- **The infrastructure you plan to install Eucalyptus on:** Think about the application workload performance and resource utilization tuning. Think about how many machines you want on your system.
- **The amount of control you plan to give Eucalyptus on your network:** Use your existing architecture 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.

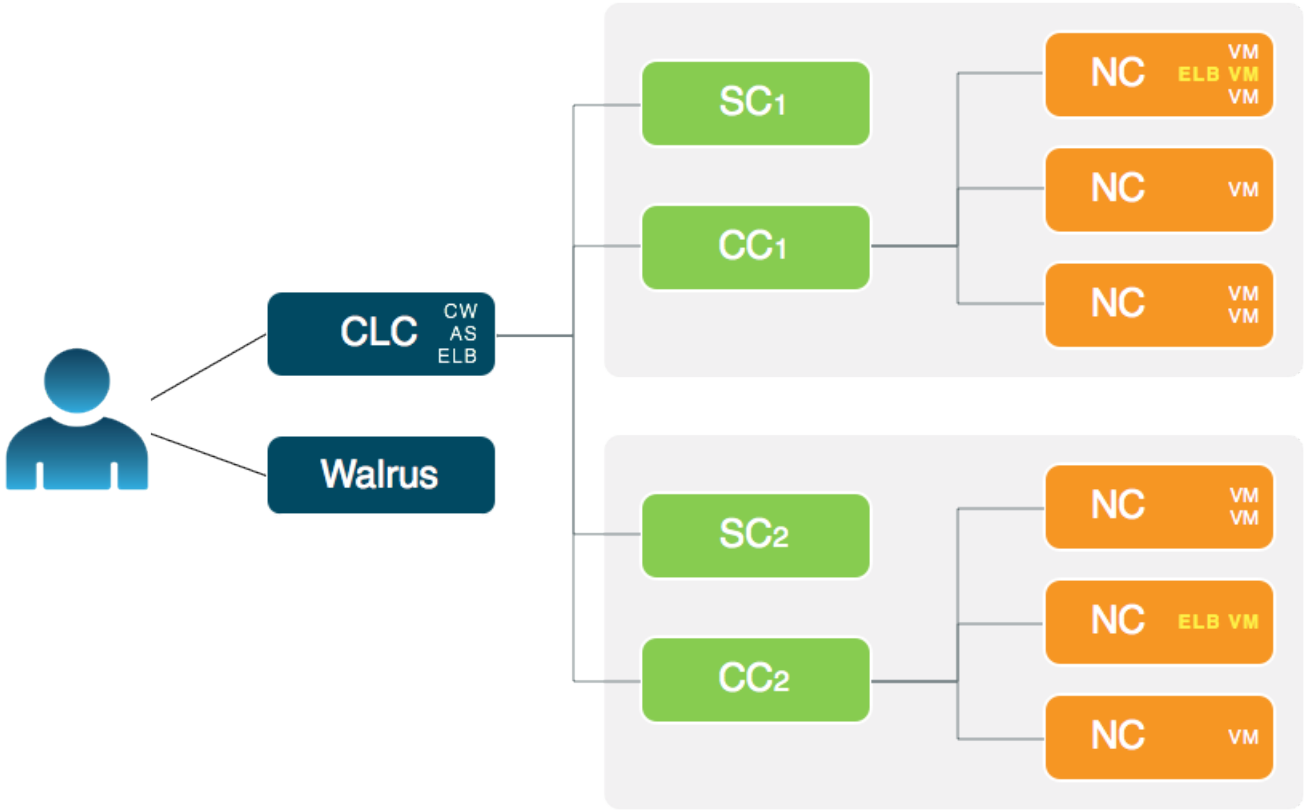
However, if the CCs are on separate network interfaces (one for the network on which the cloud components are hosted and another for the network that NCs use) the CCs will act as software routers between these networks in some networking configurations. So each cluster can use an internal private network 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), you must have primary and secondary cloud and cluster components. In the event of a failure, the secondary component becomes the primary component.



Eucalyptus HA uses a service called an 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 a 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.

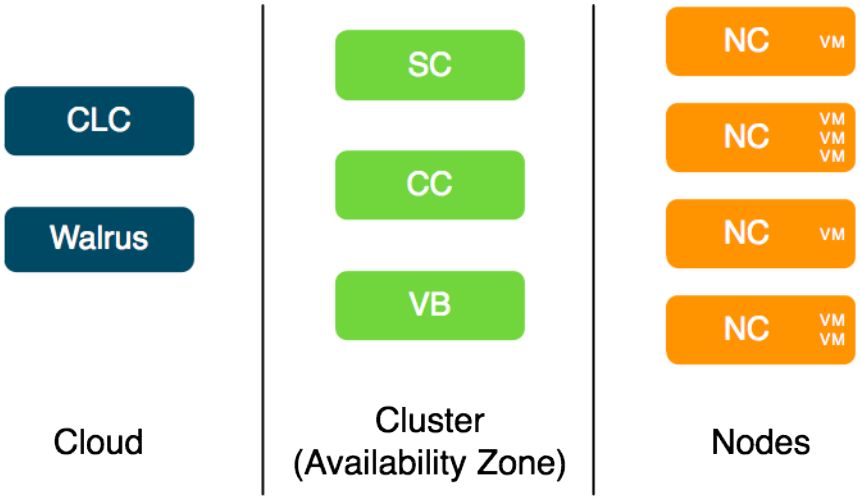
Often your decision about how to distribute Eucalyptus components across an installation must trade deployment simplicity for performance or high-availability. For example, placing all cloud and cluster components on a single machine can simplify administration because there is only one machine to monitor 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 physical 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 front end servers and the NC servers in its cluster.

SC Placement

The machine on which the SC is deployed must always have TCP/IP connectivity to the CLC. If you are a subscriber and 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. For instance, one should be able to connect from the CC/VB host to the vSphere endpoint on ports 443, 902, and 903.

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 CLC logging	`/var/lib/eucalyptus/db` `/var/log/eucalyptus`	20GB 2GB
Walrus Walrus logging	`/var/lib/eucalyptus/bukkits` `/var/log/eucalyptus`	250GB 2GB
SC	`/var/lib/eucalyptus/volumes` (EBS storage) This disk space on the SC is only required if you are not using a SAN driver.	250GB
CC CC logging	`/var/lib/eucalyptus/CC` `/var/log/eucalyptus`	5GB 2GB
NC NC logging	`/var/lib/eucalyptus/instances` `/var/log/eucalyptus`	250GB 2GB

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:

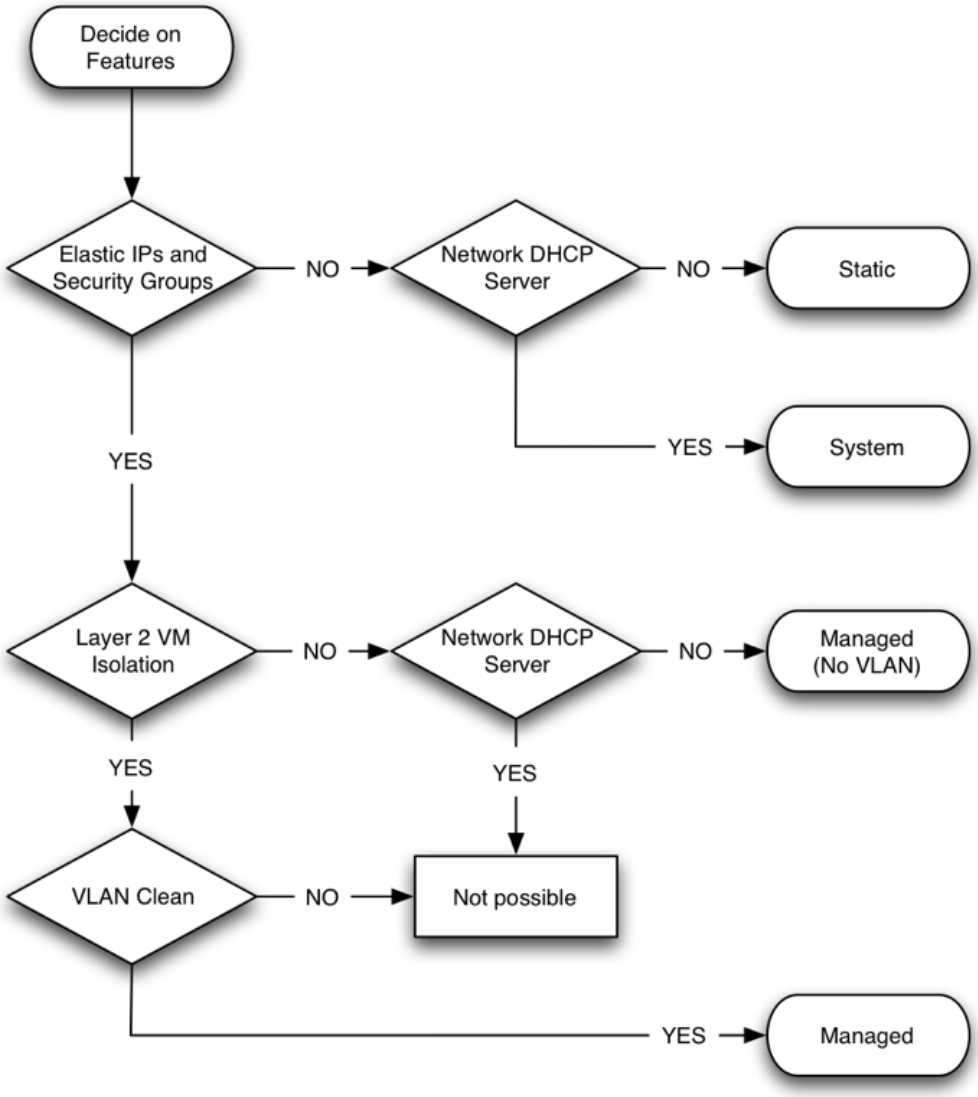
- Do you plan to support elastic IPs and security groups?
- Do you plan to provide your own network DHCP server?
- Do you plan to support Layer 2 VM isolation?

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 login and usage of a new VM instance you must authorize network access to the default security group with the `euca-authorize` command.	Managed Managed (No VLAN)
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 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 private 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 security 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 security 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 in different security groups. 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 between the CC and NCs 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 between the CC and NCs 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 provided 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

- Limited (Layer-3) 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 NC'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

Important: If you plan to use Elastic Load Balancing (ELB), note that ELB only works with Managed and Managed (No VLAN) networking modes. This is because ELB relies on security groups.

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

Important: If you plan to use Elastic Load Balancing (ELB), note that ELB only works with Managed and Managed (No VLAN) networking modes. This is because ELB relies on security groups.

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 Guest OS Support

Eucalyptus requires the following to use Windows as a guest operating system:

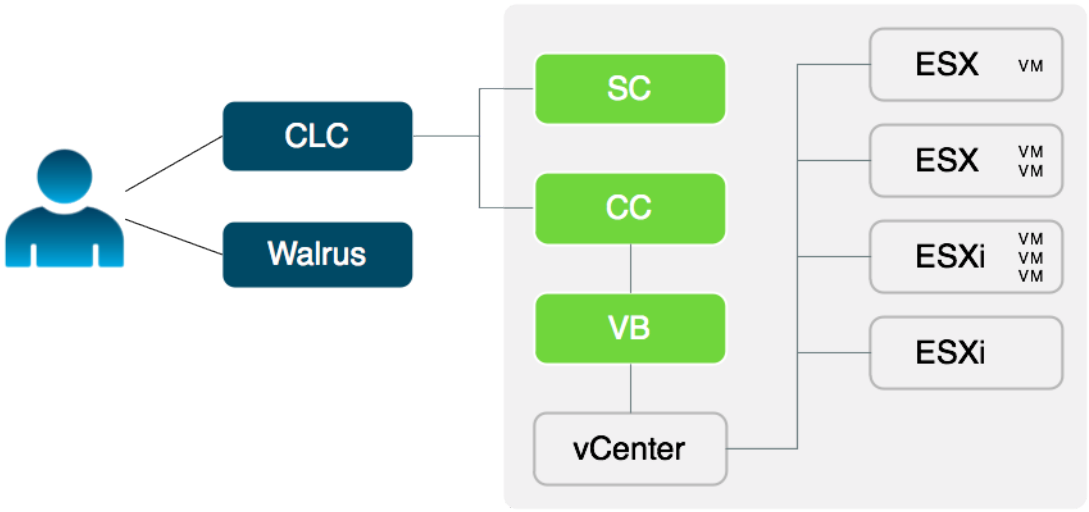
- A licensed installation copy (.iso image or CD/DVD disk) of a compatible Windows OS. Eucalyptus currently supports Windows virtual machines created from Windows Server 2003 R2 Enterprise (32/64 bit); Windows Server 2008 SP2, Datacenter (32/64 bit); Windows Server 2008 R2, Datacenter; and Windows 7 Professional.
- A VNC client such as RealVNC or Virtual Manager/Virtual Viewer 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). In the following diagram VB is controlling VMware infrastructure through a vCenter server, but it can also connect to ESX/ESXi hosts directly, without vCenter server present.



Eucalyptus provides:

- Support for VMware vSphere infrastructure as the platform for deploying virtual machines
- The ability to extend cloud-based features (for example, elastic IPs, security groups, Amazon S3, etc.) to a VMware infrastructure
- Compatibility with VMware vSphere client, which can be used alongside Eucalyptus

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

VMware Support Prerequisites

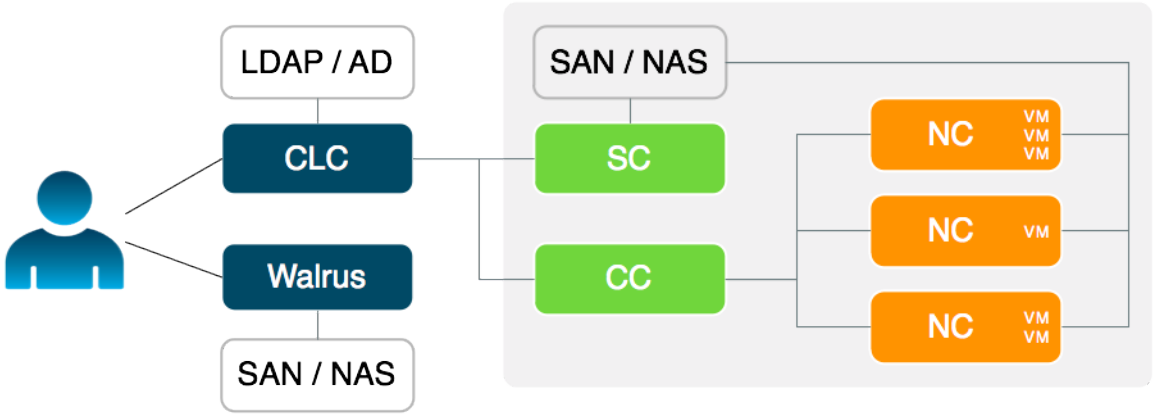
If you plan to use Eucalyptus with VMware, there are some additional prerequisites:

- 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. See "Configuring VMware" section for more details.

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), NetApp Filer FAS 2000 and FAS 6000 series and EMC VNX 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). For EMC, Eucalyptus expects that the EMC NaviSecCLI software will be installed on the Storage Controller host.

SAN Support Prerequisites

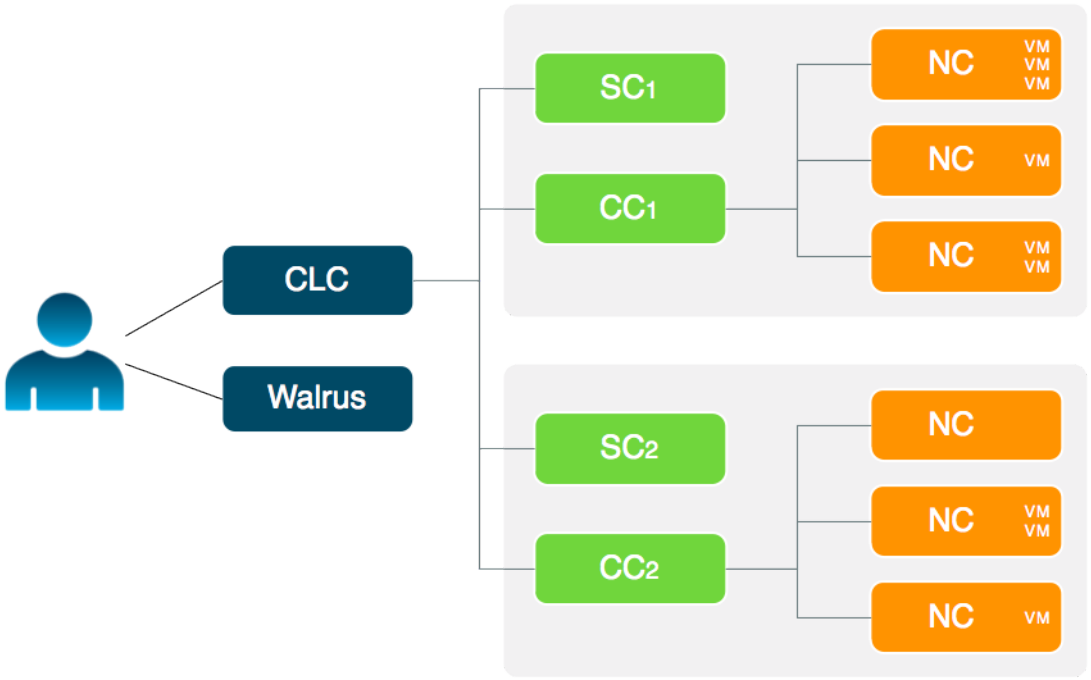
Eucalyptus supports the following SAN devices:

- 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*
- EMC VNX Series (For more information about EMC VNX, go to *VNX Family*

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.



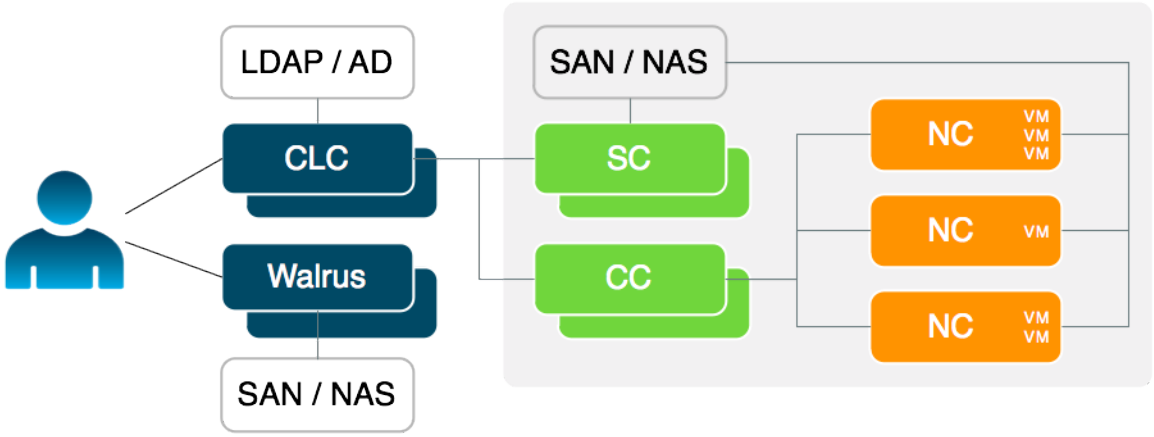
High Availability Support

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 machine running an NC fails, Eucalyptus will continue to be available for user requests. However, instances running on that specific NC 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, you can configure redundant component services. With redundant component services, Eucalyptus can isolate and mask the a component's failure.
- **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 (for example, 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 (for example, 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 (for example, 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.

SAN and Multipathing

Multipathing is a way to make the data path from the NC or SC to your SAN device highly available. Mulipathing does this by giving the host two network paths that both lead to the same data volume. This allows the host to switch from one network path to the other, in the event that one path becomes unavailable. Essentially, multipathing decreases the likelihood that a volume will become unreachable from a host (NC). For information about configuring your SAN for multipathing, see *Configure the Storage Controller*.

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.

Eucalyptus Port Usage

Eucalyptus components use a variety of ports to communicate. The following table lists the all of the important ports used by Eucalyptus.

Port	Description
TCP 5005	DEBUG ONLY: This port is used for debugging Eucalyptus (using the `--debug` flag).
TCP 8080	Port for the administrative web user interface. Forwards to 8443. Configurable with `euca-modify-property`.
TCP 8443	SSL port for the administrative web user interface. Configurable with `euca-modify-property`.
TCP 8772	DEBUG ONLY: JMX port. This is disabled by default, and can be enabled with the `--debug` or `--jmx` options for `CLOUD_OPTS`.
TCP 8773	Web services port for the CLC, Walrus, SC, and VB; also used for external and internal communications by the CLC and Walrus. Configurable with `euca-modify-property`.
TCP 8774	Web services port on the CC. Configured in the `eucalyptus.conf` configuration file.
TCP 8775	Web services port on the NC. Configured in the `eucalyptus.conf` configuration file.
TCP 8776	Used by the image cacher on the CC. Configured in the `eucalyptus.conf` configuration file.
TCP 8777	Database port on the CLC.
TCP 8779 (or next available port, up to TCP 8849)	jGroups failure detection port on CLC, Walrus, VB and SC. If port 8779 is available, it will be used, otherwise, the next port in the range will be attempted until an unused port is found.
TCP 16514	TLS port on Node Controller, required for node migrations.
UDP 7500	Port for diagnostic probing on CLC, Walrus, SC, and VB.
UDP 8773	HA membership port.
TCP/UDP 53	DNS port on the CLC.

Verify Component 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.

Note: Any firewall running on the CC must be compatible with the dynamic changes performed by Eucalyptus when working with security groups. Eucalyptus will flush the 'filter' and 'nat' tables upon boot.

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

1. Verify connection from an end-user to the CLC on TCP ports 8443 and 8773
2. Verify connection from an end-user to Walrus on TCP port 8773
3. Verify connection from the CLC, SC, and NC (or VB) to SC on TCP port 8773
4. Verify connection from the CLC, SC, and NC (or VB) to Walrus on TCP port 8773
5. Verify connection from Walrus, SC, and VB to CLC on TCP port 8777
6. Verify connection from CLC to CC on TCP port 8774
7. Verify connection from CC to VB on TCP port 8773
8. Verify connection from CC to NC on TCP port 8775
9. Verify connection from NC (or VB) to Walrus on TCP port 8773. Or, you can verify the connection from the CC to Walrus on port TCP 8773, and from an NC to the CC on TCP port 8776
10. Verify connection from public IP addresses of Eucalyptus instances (metadata) and CC to CLC on TCP port 8773
11. Verify TCP connectivity between CLC, Walrus, SC and VB on TCP port 8779 (or the first available port in range 8779-8849)
12. Verify connection between CLC, Walrus, SC, and VB on UDP port 7500
13. Verify multicast connectivity for IP address 228.7.7.3 between CLC, Walrus, SC, and VB on UDP port 8773
14. If DNS is enabled, verify connection from an end-user and instance IPs to DNS ports
15. If you use tgt (iSCSI open source target) for EBS storage, verify connection from NC to SC on TCP port 3260
16. 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.

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.

To configure a bridge in CentOS 6 or RHEL6, 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). The following steps describe how to set up a bridge on both CentOS 6 and RHEL 6. We show examples for configuring bridge devices that either obtain IP addresses using DHCP or statically.

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:

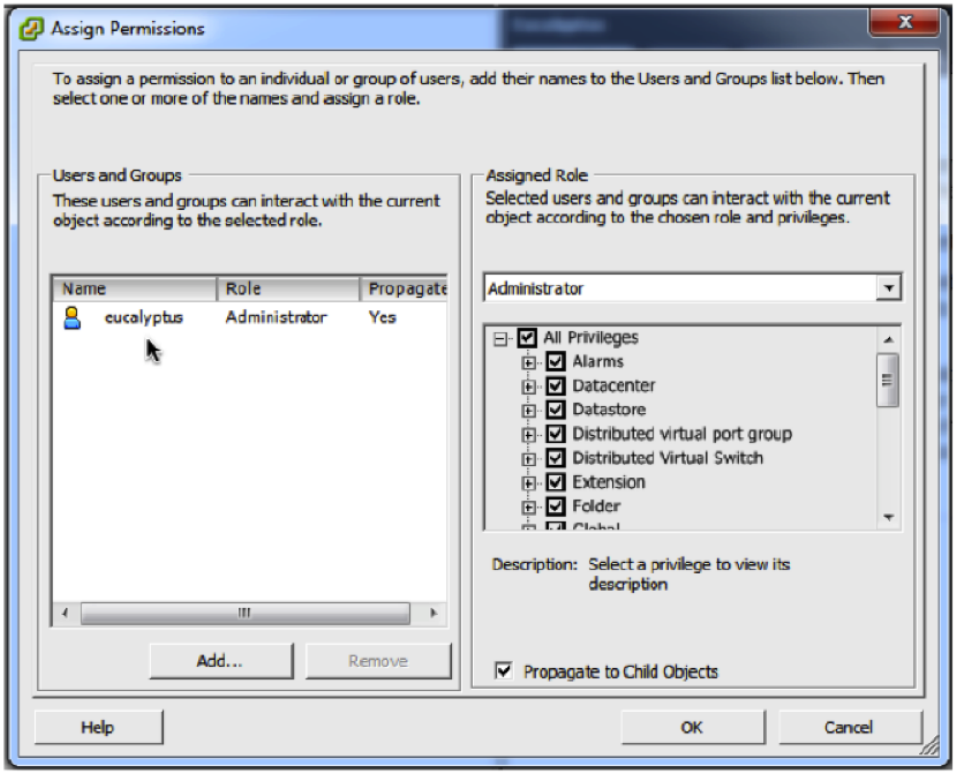
```
service network restart
```

You are now ready to *Configure VMware*.

Configure 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 managed by a vCenter server, create one new user and two new roles as described next.

Create New User

To give the minimal required amount of control to Eucalyptus over your vSphere infrastructure managed on vCenter, create one new user and two new roles. The new user and its password will be used for granting Eucalyptus access to the infrastructure.

1. Create a user (e.g., named `eucalyptus`) on the system where vCenter server is running.

2. Create a role (e.g., named `Eucalyptus vSphere`), for use at the top level of the vSphere hierarchy, with the following privileges:
   - Global
     - Licenses
3. Create a role (e.g., named `Eucalyptus`), for use with vSphere resources to be used by Eucalyptus, with the following privileges:
   - Datastore
     - Allocate Space
     - Browser Datastore
     - Low level file operations
   - Folder
     - Create folder
   - Host
     - Configuration
       - Network Configuration
       - Storage partition configuration
   - Network
     - Assign network
     - Remove
   - Resource
     - Assign Virtual Machine to Resource Pool
   - Virtual Machine
     - (all Virtual Machine permissions)
4. Associate the user with the top-level role
   a) Right-click on the top-level resource, named after vCenter, and select **Add Permission...**
   b) In **Users and groups** section click **Add...**
   c) Add user `eucalyptus` with assigned role `Eucalyptus vSphere` and **Propagate to Child Objects** set to **No**
5. Associate the user with the resource-level role

   For each resource or collection of resources that you want Eucalyptus to use, the `eucalyptus` user must be given sufficient privileges by using the `Eucalyptus` role. 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 just for that datacenter.
   a) Right-click on each of the resources to be used by Eucalyptus and select **Add Permission...**
   b) In **Users and groups** section click **Add...**
   c) Add user `eucalyptus` with assigned role `Eucalyptus` and **Propagate to Child Objects** set to **Yes**

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".

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, since VMware Tools enable clean shutdown of VMs from outside the instance. For information about installing VMware Tools, go to the VMware documentation at *http://www.vmware.com*.

Disable the Firewall

If you have existing firewall rules on your hosts, you should disable the firewall in order to install Eucalyptus. You should re-enable it after installation. If you do not have a firewall enabled, skip this step.

1. To disable your firewall:
   a) Run the command `system-config-firewall-tui`
   b) Turn off the **Enabled** check box.
2. 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.

To configure SELinux to allow Eucalyptus access:

1. Open `/etc/selinux/config` and edit the line `SELINUX=enforcing` 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.

```
yum install ntp
```

2. Open the `/etc/ntp.conf` file 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.

```
chkconfig ntpd on
```

5. Start NTP.

```
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 6 and RHEL 6. The following sections show installation steps on each supported Linux distribution.

Eucalyptus Subscription allows you access to additional software modules. If you are a subscriber, you will receive an entitlement certificate and a private key that allow you to download Eucalyptus subscription modules. You will also receive a GPG public key to be used to verify the Eucalyptus software's integrity. The files will come in the form of a platform specific package.

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

Install Eucalyptus from Release Packages

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 6 or RHEL 6:

1. Configure the Eucalyptus package repository on each host that will run a Eucalyptus component:

```
yum install
http://downloads.eucalyptus.com/software/eucalyptus/3.3/centos/6/x86_64/eucalyptus-release-3.3.noarch.rpm
```

   Enter `y` when prompted to install this package.

2. Configure the Euca2ools package repository on each host that will run a Eucalyptus component or Euca2ools:

```
yum install
http://downloads.eucalyptus.com/software/euca2ools/3.0/centos/6/x86_64/euca2ools-release-3.0.noarch.rpm
```

   Enter `y` when prompted to install this package.

3. Configure the EPEL package repository on each host that will run a Eucalyptus component or Euca2ools:

```
yum install
http://downloads.eucalyptus.com/software/eucalyptus/3.3/centos/6/x86_64/epel-release-6.noarch.rpm
```

   Enter `y` when prompted to install this package.

4. Configure the ELRepo repository on each host that will run Walrus:

```
yum install
http://downloads.eucalyptus.com/software/eucalyptus/3.3/centos/6/x86_64/elrepo-release-6.noarch.rpm
```

   Enter `y` when prompted to install this package.

5. For RHEL 6 systems only, it is necessary to enable the Optional repository in Red Hat Network for each NC, as follows:

   a) Go to *http://rhn.redhat.com* and navigate to the system that will run the NC.

b) Click **Alter Channel Subscriptions**.
c) Make sure the **RHEL Server Optional** checkbox is checked.
d) Click **Change Subscriptions**.

6. If you are not a Eucalyptus subscriber, skip this step. If you are a Eucalyptus subscriber, you should have received an rpm package file containing subscription-only components. Install the Eucalyptus subscription package on each host that will run a Eucalyptus component, as follows:

```
yum install eucalyptus-enterprise-release-3.3*.noarch.rpm
```

Enter `y` when prompted to install this package.

7. If you are a Eucalyptus subscriber and use VMware Broker, install the VMware Broker packages on the host that will run your Cluster Controller (CC), as follows:

```
yum install eucalyptus-enterprise-vmware-broker
eucalyptus-enterprise-vmware-broker-libs
```

Enter `y` when prompted to install this package.

8.  **Note:** Clouds that use the VMware hypervisor do not have NCs; if you plan to use VMware then skip this step.

a) Install the Eucalyptus node controller software on each planned NC host:

```
yum install eucalyptus-nc
```

b) Check that the KVM device node has proper permissions.

Run the following command:

```
ls -l /dev/kvm
```

Verify the output shows that the device node is owned by user root and group kvm.

```
crw-rw-rw- 1 root kvm 10, 232 Nov 30 10:27 /dev/kvm
```

If your kvm device node does not have proper permissions, you need to reboot your NC host.

9. Install the Eucalyptus cloud controller software on each planned CLC host:

```
yum install eucalyptus-cloud
```

10. Install the software for the remaining Eucalyptus components. The following example shows most components being installed on the same host. We recommend that you use different hosts for each component:

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

11. If you would like Load Balancer support enabled in your Cloud, you will need to install the Load Balancer image package on the machine hosting the primary CLC:

```
yum install eucalyptus-load-balancer-image
```

12. If you are a subscriber and use SAN, run the appropriate command for your device on each machine hosting a CLC:

For EMC SAN:

```
yum install eucalyptus-enterprise-storage-san-emc-libs
```

For EqualLogic SAN:

```
yum install eucalyptus-enterprise-storage-san-equallogic-libs
```

For NetApp SAN:

```
yum install eucalyptus-enterprise-storage-san-netapp-libs
```

13. If you are a subscriber and use SAN, run the appropriate command for your device on each machine hosting a SC:

For EMC SAN:

```
yum install eucalyptus-enterprise-storage-san-emc
```

Important: To use Eucalyptus with EMC SAN support, you must have the `NaviCLI-Linux-64-latest.rpm` package installed on each SC. This package is not supplied with Eucalyptus, please see your SAN vendor if it is not already installed.

For EqualLogic SAN:

```
yum install eucalyptus-enterprise-storage-san-equallogic
```

For NetApp SAN:

```
yum install eucalyptus-enterprise-storage-san-netapp
```

14. After you have installed Eucalyptus, test multicast connectivity between each CLC and Walrus, SC, and VMware broker host.

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) Once the receiver command blocks, simultaneously run the following sender command on each Walrus host:

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

The two applications should be able to connect and arbitrary lines entered on the sender should appear on the receiver.

c) Repeat the previous step on each SC host and VMware broker host

d) If you are installing an HA environment, repeat these tasks with the secondary controllers.

Your installation is complete.

You are now ready to *Configure Eucalyptus*.

Installing Eucalyptus from Nightly Packages

Important: Eucalyptus nightly packages are latest Eucalyptus builds. They should be considered unstable/"bleeding edge" software and should not be installed in production. In addition, upgrades from nightlies to released software are not supported.

To install Eucalyptus nightly builds on servers running CentOS 6 or RHEL 6:

1. On all servers, run the following commands:

```
yum install http://downloads.eucalyptus.com/
software/eucalyptus/nightly/3.3/centos/6/x86_64/eucalyptus-release-3.3.2.noarch.rpm
```

   Enter `y` when prompted to install this package.

2. On all systems that will run either Eucalyptus or Euca2ools, run the following commands:

```
yum install http://downloads.eucalyptus.com/software/
euca2ools/nightly/3.0/centos/6/x86_64/euca2ools-release-3.0.noarch.rpm
```

   Enter `y` when prompted to install this package.

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

```
yum install http://downloads.eucalyptus.com/
software/eucalyptus/nightly/3.3/centos/6/x86_64/elrepo-release-6.noarch.rpm
```

   Enter `y` when prompted to install this package.

4. Configure the EPEL package repository:

```
yum install
http://downloads.eucalyptus.com/software/eucalyptus/nightly/3.3/centos/6/x86_64/epel-release-6.noarch.rpm
```

   Enter `y` when prompted to install this package.

5. On all servers, enter:

```
yum update
```

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
yum install 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. If you would like Load Balancer support in your cloud, you will need to install the Load Balancer image package on the machine hosting the primary CLC:

```
yum install eucalyptus-load-balancer-image
```

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 these machines. They are running VMware.

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 most commonly used VNET options are described in the following table.

Option	Description	Modes

`VNET_ADDRESSPERNET`	This option controls how many VM instances can simultaneously be part of an individual user's security group. This option is set to a power of 2 (16, 24, 32, 64, etc,) but it should never be less than 8 and it cannot be larger than: (the total number of available IP addresses - 2).
	
	This option is used with VNET_NETMASK to determine how the IP addresses that are available to VMs are distributed among security groups. VMs within a single security group can communicate directly. Communication between VMs within a security group and clients or VMs in other security groups is controlled by a set of firewall rules. For example, setting
	
	`VNET_NETMASK="255.255.0.0"`
	`VNET_ADDRESSPERNET="32"`
	
	defines a netmask of 255.255.0.0 that uses 16 bits of the IP address to specify a network number. The remaining 16 bits specify valid IP addresses for that network meaning that 2^16 = 65536 IP addresses are assignable on the network. Setting `VNET_ADDRESSPERNET="32"` tells Eucalyptus that each security group can have at most 32 VMs in it (each VM getting its own IP address). Further, it stipulates that at most 2046 security groups can be active at the same time since 65536 / 32 = 2048. Eucalyptus reserves two security groups for its own use.
	
	In addition to subnets at Layer 3, Eucalyptus uses VLANs at Layer 2 in the networking stack to ensure isolation (Managed mode only).
		Managed, Managed (No VLAN)

`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 setting for KVM is `br0`.	Static System Managed (No VLAN)

Option	Description	Modes
`VNET_BROADCAST`, `VNET_ROUTER`	The network broadcast and default gateway to supply to instances in DHCP responses.	Static
`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 6 and RHEL 6, this is typically `root`. Default: `dhcpd`	Static Managed Managed (No VLAN)
`VNET_DNS`	The address of the DNS server to supply to instances in DHCP responses. Example: `VNET_DNS="173.205.188.129"`	Static Managed Managed (No VLAN)
`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_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_MACPREFIX`	This option is used to specify a prefix for MAC addresses generated by Eucalyptus for VM instances. The prefix has to be in the form `HH:HH` where H is a hexadecimal digit. Example: `VNET_MACPREFIX="D0:D0"`	System, Managed, Managed (No VLAN)
`VNET_MODE`	The networking mode in which to run. The same mode must be specified on all CCs and NCs in your cloud. Valid values: `STATIC`, `SYSTEM`, `MANAGED`, `MANAGED-NOVLAN`, Default: `SYSTEM`	All

Option	Description	Modes
`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
`VNET_PUBLICIPS`	A space-separated list of individual and/or hyphenated ranges of public IP addresses to assign to instances. If you do not set a value for this option, all instances will receive only private IP addresses. Example: `VNET_PUBLICIPS="173.25.18.10-173.25.18.254"`	Managed Managed (No-VLAN)
`VNET_SUBNET, VNET_NETMASK`	These options control the internal private network used by instances within Eucalyptus. 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 can reside in each security group. Eucalyptus reserves eleven addresses per security group.	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:

We recommend allowing the CC to act as the gateway for NCs, in Managed mode. To do so, ensure that traffic from all NCs (on private network) is allowed to be masqueraded on the CC, and set the output interface to the the public interface of the CC. You can do this using the following `iptables` command:

```
iptables -t nat -A POSTROUTING -s 10.101.104.0/16 -o em1 -j MASQUERADE
```

Where `10.101.104.0/16` is the private network containing all NCs, and `em1` is the public interface set on the CC.

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. In a multi-cluster configuration, you must set `VNET_PUBLICIPS` identically on all CCs.

NC Configuration

Important:

We recommend allowing the CC to act as the gateway for NCs, in Managed mode. To do so, ensure that traffic from all NCs (on private network) is allowed to be masqueraded on the CC, and set the output interface to the the public interface of the CC. You can do this using the following `iptables` command:

```
iptables -t nat -A POSTROUTING -s 10.101.104.0/16 -o em1 -j MASQUERADE
```

Where `10.101.104.0/16` is the private network containing all NCs, and `em1` is the public interface set on the CC.

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="[Subnet for VMs private IPs. Example: 192.168.0.0]"
VNET_NETMASK="[Netmask for the vnet_subnet. Example: 255.255.0.0]"
VNET_DNS="[DNS server IP]"
VNET_ADDRSPERNET="[Number of simultaneous instances per security group]"
VNET_PUBLICIPS="[Free public IP 1] [Free public IP 2] ..."
VNET_LOCALIP="[IP address that other CCs can use to reach this CC]"
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: br0]"
```

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 cloud.

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="<public subnet to be used by instances>"

VNET_NETMASK="<netmask used in your network layout>"

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 cloud.

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 Loop Devices

In order to start new instances, Eucalyptus needs a sufficient number of loop devices 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.

Eucalyptus installs with a default loop device amount of 256. If you want to change this number, perform the following steps. Otherwise, skip this section and continue to *Configure Multi-Cluster Networking*.

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

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.

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

Important: When L2TP tunneling is enabled in a multi-cluster setup, make sure that you are using different IP ranges for the nodes in each cluster.

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 its list of local IP addresses upon startup, but if the IP that was used to register the CC is not locally available, you can override the CC's notion of 'self' by setting the `VNET_LOCALIP` variable in the `eucalyptus.conf` file.

Important: 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. Please disable tunneling by setting `DISABLE_TUNNELING=Y` in `eucalyptus.conf` on both CC hosts.

Configure the Firewall

Restricting network access

This section provides basic guidance on setting up a firewall around your Eucalyptus components. It is not intended to be exhaustive.

On CLC, Walrus, SC, and VB, you should allow for the following jGroups traffic:

- TCP connections between CLC, Walrus, SC, and VB on port 8779 (or the first available port in range 8779-8849)
- UDP connections between CLC, Walrus, SC, and VB on port 7500
- Multicast connections between CLC, Walrus, SC, and VB to IP 228.7.7.3 on UDP port 8773

On the CLC, you should additionally allow the following connections:

- TCP connections from end-users on ports 8773 and 8443
- TCP connections from CC and Eucalyptus instances (public IPs) on port 8773 (for metadata service)
- TCP connections from Walrus, SC, and VB on port 8777
- End-user and instance connections to DNS ports

On the CC, you should ensure that all firewall rules are compatible with the dynamic changes performed by Eucalyptus, described in the section below. You should also allow the following connections:

- TCP connections from CLC on port 8774
- TCP connections from NC on port 8776, if CC image proxying is enabled

On Walrus, you should also allow the following connections:

- TCP connections from end-users on port 8773
- TCP connections from SC, NC, and VB on port 8773
- TCP connections from CC on port 8773, if CC image proxying is enabled

On the SC, you should also allow the following connections:

- TCP connections from CLC, NC, and VB on TCP port 8773
- TCP connections from NC on TCP port 3260, if tgt (iSCSI open source target) is used for EBS storage

On the VMware Broker, you should also allow the following connections:

- TCP connections from CC on port 8773

On the NC, you should allow the following connections:

- TCP connections from CC on port 8775
- TCP connections from other NCs on port 16514
- DHCP traffic forwarding to VMs
- Traffic forwarding to and from instances' private IP addresses

Managing iptables rules for the CC

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 > /etc/eucalyptus/iptables-preload
```

Caution: Performing this operation to define special iptables rules that are loaded when Eucalyptus starts 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.

Note: Eucalyptus 3.3 requires version 7 of the Java Virtual Machine. Make sure that your CLOUD_OPTS settings in the /etc/eucalyptus/eucalyptus.conf file either do not set `--java-home`, or that `--java-home` points to a version 7 JVM. This needs to happen before services are started but after the upgraded packages are installed.

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

Start the Cloud Controller (CLC)

1. Log in to the Cloud Controller (CLC).
2. Enter the following command to initialize the CLC:

Note: Please ensure that the "eucalyptus-cloud" process is not running prior to executing this command.

```
/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

Tip: If you aren't using the subscription-only VMware Broker module, skip this section.

If you are using Eucalyptus with VMware support, perform the following tasks.

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
```

Important: If you are re-installing the SC, please restart the tgt (iSCSI open source target) daemon.

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

Note: You must register the node controllers before starting the NC service. See *Register the NCs* for information on how to register 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
- If you are using the subscription only VMware Broker, it is listening on port 8773
- The CC is listening on port 8774
- 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 NC on every CC for the cluster on which the NC 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** (`--register-XYZ`) you are registering, because this affects where the commands must be executed.
- The **partition** (`--partition`) the component will belong to. The partition is the same thing as availability zone in AWS.
- The **name** (`--component`) ascribed to the component. This is the name used to identify the component in a human-friendly way. This name is also used when reporting system state changes which require administrator attention. This name must be globally-unique with respect to other component registrations. To ensure this uniqueness, we recommend using a combination of the component type (CLC, SC, CC, etc) and system hostname or IP address when you choose your component names. For example: `clc-eucahost15` or `clc-192.168.0.15`.
- The **IP address** (`--host`) of the service being registered. The host must be specified by IP address to function correctly.

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

Note: We recommend that you use IP addresses rather than host names when registering Eucalyptus components. If you do use hostnames, the underlying IP address may not be a site-local, any-cast, loopback, link-local, or multicast address.

Note: Once you've registered a Eucalyptus component with a host name, to avoid connectivity issues, do not change the host name's underlying 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 [Secondary_CLC_IP] --component [CLC_Name]
```

The `partition` name for the CLC has to be `eucalyptus`. The `component` name is a unique name for this particular component: we recommend a format such as `CLC-[hostname]`.

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]
```

The `partition` name for Walrus has to be `walrus`. Like the CLC, the component name is a unique name for this particular component: we recommend a format such as `walrus-[hostname]`.

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]
```

We recommend that you set the `partition` name to a descriptive name for the availability zone controlled by the CC. For example: `cluster01`.

The `component` must be a unique name. We recommend that you use a short-hand name of the hostname or IP address of the machine, like `cc-[hostname]` or `cc-[IP address]`.

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 the subscription-only VMware Broker module, 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 [broker_name]
```

The VMware Broker must have the same `partition` name as the CC in the same cluster. Like the other components, the `component` is a unique name for this particular component: we recommend a format such as `broker-[hostname]`.

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:

   Note: We recommend that you use IP addresses instead of DNS names when registering Eucalyptus components.

```
/usr/sbin/euca_conf --register-sc --partition [partition_name] --host
[SC_IP_address]
 --component [SC_name]
```

   An SC must have the same `partition` name as the CC in the same cluster. Like the other components, the `component` is a unique name for this particular component: we recommend a format such as `sc-[hostname]`.

   Warning: Newly registered SCs will be in the BROKEN state until they are explicitly configured to use a backend storage provider. The output of the registration for the first SC registered in a partition will look like:

```
SERVICE storage         PARTI00         SC71            BROKEN     37
   http://192.168.51.71:8773/services/Storage
arn:euca:eucalyptus:PARTI00:storage:SC71/
Registered the first storage controller in partition 'PARTI00'.  You
must choose a storage back end with ``euca-modify-property -p
PARTI00.storage.blockstoragemanager=$BACKEND''
```

   This is completely normal and simply indicates that further action must be taken to configure the SC before it will become fully functional. For information about configuring the SC, see *Configure the Runtime Environment->Configure the Storage Controller*

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 the subscription only VMware Broker module, you can skip this task. Eucalyptus software is not installed on machines that are running VMware. You do not have to register the NCs. Instead, you have to configure the VMware Broker, as described in the *Configure VMware Support* section.

Important: If you are using host names rather than IP addresses when registering your NCs, ensure that DNS is working properly, or populate /etc/hosts for all nodes in a cluster.

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

Warning: This task is only for high availability (HA) installations. Do not register an arbitrator if you do not have an HA environment. If you do, you will not be able to access your cloud.

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_IP]>
```

   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_IP]` is the IP of the machine 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 the Storage Controller

Eucalyptus offers SAN support for Eucalyptus block storage (EBS). Eucalyptus directs the Storage Controller (SC) to manage any supported SAN devices. 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 currently offers several backend providers for the Storage Controller: Overlay, DAS, Equallogic, Netapp, and EMC-VNX.

The Eucalyptus Storage Controller must be configured explicitly upon registration. This is a change from previous versions (pre-3.2) of Eucalyptus, which would configure themselves to a default configuration using a tgtd-based filesystem-backed storage controller to provide volumes and snapshots directly from the Storage Controller. As of version 3.2, Eucalyptus Storage Controllers automatically go to the BROKEN state after being registered with the CLC and will remain in that state until the administrator explicitly configures the SC by telling it which backend storage provider to use.

You can check the state of a storage controller by running

```
euca-describe-services -E
```

and note the state and status message of the Storage Controller(s). The output for an unconfigured SC will look like:

```
SERVICE storage         PARTI00         SC71            BROKEN     37
http://192.168.51.71:8773/services/Storage
arn:euca:eucalyptus:PARTI00:storage:SC71/
SERVICEEVENT 6c1f7a0a-21c9-496c-bb79-23ddd5749222
```

```
arn:euca:eucalyptus:PARTI00:storage:SC71/
SERVICEEVENT 6c1f7a0a-21c9-496c-bb79-23ddd5749222 ERROR
SERVICEEVENT 6c1f7a0a-21c9-496c-bb79-23ddd5749222 Sun Nov 18 22:11:13 PST 2012
SERVICEEVENT 6c1f7a0a-21c9-496c-bb79-23ddd5749222 SC blockstorageamanger not
configured. Found empty or unset manager(unset). Legal values are: das,overlay
```

Note the error above: `SC blockstoragemanager not configured. Found empty or unset manager(unset). Legal values are: das,overlay`.

This indicates that the SC is not yet configured. It can be configured by setting the `[partition].storage.blockstoragemanager` property to either 'das' or 'overlay'.

If you have installed the Eucalyptus Enterprise packages for your SAN, you will also see additional options in the output line above, and can set the block storage manager to 'netapp','emc-vnx-flare31','emc-vnx', or 'equallogic' as appropriate.

You can verify that the SC blockstoragemanager is unset using:

```
euca-describe-properties | grep blockstorage
```

To configure SAN support, follow the steps for your desired backend storage device: *Open-Source ISCSI Filesystem-backed*, *Dell Equallogic*, *JBOD*, *Netapp*, or *EMC VNX*.

Configuring the SC to use the local filesystem (Overlay)

This was the default configuration option for the SC in pre-3.2 Eucalyptus. In this configuration the SC itself hosts the volume and snapshots for EBS and stores them as files on the local filesystem. It uses standard linux iSCSI tools to serve the volumes to instances running on NCs.

1. Configure the SC to use the OverlayManager for storage.

```
euca-modify-property -p <partition>.storage.blockstoragemanager=overlay
```

   The output of the command should be similar to:

```
PROPERTY PARTI00.storage.blockstoragemanager overlay was <unset>
```

2. Verify that the property value is now: 'overlay'

```
euca-describe-properties | grep blockstorage
```

Enable Dell Equallogic SANs

1. Configure the SC to use the EquallogicManager for storage.

```
euca-modify-property -p <partition>.storage.blockstoragemanager=equallogic
```

   The output of the command should be similar to:

```
PROPERTY PARTI00.storage.blockstoragemanager equallogic was <unset>
```

2. Verify that the propery value is now: 'equallogic'

```
euca-describe-properties | grep blockstorage
```

3. On the CLC, run the following command to verify that the SC is listed; note that it may be in the `BROKEN` state:

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

4. On the CLC (the primary CLC in an HA setup), enable SAN support in Eucalyptus by entering your SAN's hostname or IP address, the username, and password:

```
euca-modify-property -p [partition_name].storage.sanhost=[SAN_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]
```

   If you have multiple management IP addresses for the SAN adapter, provide a comma-delimited list of IP addresses to the `[partition_name].storage.sanhost` property.

Your Equallogic SAN is now ready to use with Eucalyptus.

Enable Direct Attached Storage (JBOD) SANs

1. Configure the SC to use the (Direct Attached Storage) DASManager for storage.

```
euca-modify-property -p <partition>.storage.blockstoragemanager=das
```

   The output of the command should be similar to:

```
PROPERTY PARTI00.storage.blockstoragemanager das was <unset>
```

2. Verify that the propery value is now: 'das'

```
euca-describe-properties | grep blockstorage
```

3. On the CLC, run the following command to verify that the SC is listed; note that it may be in the `BROKEN` state:

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

4. On the CLC (the primary CLC in an HA setup), set the DAS device name property. The device name can be either a raw device (/dev/sdX, for example), or the name of an existing Linux LVM volume group.

```
euca-modify-property -p <cluster name>.storage.dasdevice=<device name>
```

   For example:

```
euca-modify-property -p cluster0.storage.dasdevice=/dev/sdb
```

Your SAN is now ready to use with Eucalyptus.

Enable NetApp SANs

Eucalyptus supports both NetApp Clustered ONTAP and traditional 7-mode SANs. NetApp Vservers and 7-mode Filers (FAS 2000 and FAS 600 series) are managed by Eucalyptus using NetApp Manageability Software Development Kit (NMSDK) and Data ONTAP APIs. This section covers enabling both NetApp Clustered ONTAP and traditional 7-mode SANs.

Enable NetApp 7-mode SANs

To configure NetApp 7-mode Filer and enable the SAN in Eucalyptus:

1. Verify Data ONTAP version for the 7-mode Filer is 7.3.3 or later.
2. Verify SSL access by typing `secureadmin status`
3. If SSL is marked inactive, enable with `secureadmin setup ssl` and generate a new certificate.
4. Turn on SSL access with `options httpd.admin.ssl.enable on`
5. Enable the iSCSI service on the NetApp device with `option iscsi.enable on` or `option licensed_feature.iscsi.enable on` if you have an embedded license on your array.
6. Turn on the iSCSI service with `iscsi start`
7. Enable the iSCSI service on the NetApp device with `enable iscsi service`
8. 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.
9. Verify that you have a license for FlexClone installed. At the shell prompt, enter `license` to see the list of all installed licenses.
10. Verify that administrator account credentials for NetApp Filer are available to be configured in Eucalyptus. If not, create a new administrator account for use by Eucalyptus
11. Configure the SC to use the NetappManager for storage.

```
euca-modify-property -p <partition>.storage.blockstoragemanager=netapp
```

The output of the command should be similar to:

```
PROPERTY <partition>.storage.blockstoragemanager netapp was <unset>
```

12. Verify that the propery value is now: 'netapp'

```
euca-describe-properties | grep blockstorage
```

13. On the CLC, run the following command to verify that the SC is listed; note that it may be in the `BROKEN` state:

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

14. Wait for the SC to transition to the NOTREADY or DISABLED state.
15. On the CLC (the primary CLC in an HA setup), enable NetApp SAN support in Eucalyptus by entering the Filer's hostname or IP address, the username and password of the administrator account, and CHAP username.

Note: Eucalyptus uses Challenge Handshake Authentication Protocol (CHAP) for disk operations. The CHAP username can be any value, however it should be unique when sharing a NetApp Filer across multiple Eucalyptus clusters.

Note: CHAP support for NetApp has been added in Eucalyptus 3.3. An SC will not transition to ENABLED state until the CHAP username is configured.

```
euca-modify-property -p <partition>.storage.sanhost=<Filer_IP_address>
euca-modify-property -p <partition>.storage.sanuser=<Filer_admin_username>
euca-modify-property -p <partition>.storage.sanpassword=<Filer_admin_password>
euca-modify-property -p <partition>.storage.chapuser=<Chap_username>
```

16. Wait for the SC to transition to the ENABLED state.

Note: The SC must be in the ENABLED state before configuring the following properties.

17. If no aggregate is set, Eucalyptus will query the NetApp Filer for all available aggregates and use the one that has the highest capacity (free space) by default. To make Eucalyptus use specific aggregate(s) configure the following property:

```
euca-modify-property -p
<partition>.storage.aggregate=<aggregate_1_name,aggregate_2_name,...>
```

If you want Eucalyptus to use the smallest aggregate first configure the following property:

```
euca-modify-property -p <partition>.storage.uselargestaggregate=false
```

18. Set the iSCSI data IP on the ENABLED CLC. This IP is used by NCs to perform disk operations on the Filer.

Note: Filer IP address can be used as the data port IP. If this is not set, Eucalyptus will automatically use the Filer IP address/hostname.

Note: Eucalyptus does not support Multipath I/O for NetApp 7-mode Filers.

```
euca-modify-property -p <partition>.storage.ncpaths=<ip>
```

19. Set the iSCSI data IP on the ENABLED CLC. This IP is used by the SC to perform disk operations on the Filer. The SC connects to the Filer in order to transfer snapshots to Walrus during snapshot operations.

Note: The Filer IP address can be used as the data port IP. If this is not set, Eucalyptus will automatically use the Filer IP address/hostname.

Note: Eucalyptus does not support Multipath I/O for NetApp 7-mode Filers.

```
euca-modify-property -p <partition>.storage.scpaths=<ip>
```

Your Netapp 7-mode SAN is now ready to use with Eucalyptus.

Enable NetApp Clustered Data ONTAP SAN

Eucalyptus integrates with NetApp Clustered ONTAP SAN by operating against a Vserver. SC must be configured to operate against Vserver contained in the NetApp Clustered ONTAP environment.

For more information on NetApp Clustered Data ONTAP, see *Clustered Data ONTAP 8.1 and 8.1.1: An Introduction*.

To configure NetApp Vserver and enable the SAN in Eucalyptus:

1. Verify Clustered Data ONTAP version for the SAN is 8.1.1 or later.
2. Verify that FlexClone and iSCSI licenses are installed on the SAN.
3. Verify that a Vserver with iSCSI data protocol is available for use by Eucalyptus.
4. Verify that Vserver administration is delegated to a user with administrative privileges for that Vserver. If not, create a new new Vserver administrator account for use by Eucalyptus.
5. Verify that a management (only) Logical Interface (LIF) is configured for the Vserver and an IP address or hostname is assigned to it.
6. Verify that data LIFs are configured on the Vserver.
7. Verify that one or more aggregates with sufficient spare capacity exists.
8. Verify the network connectivity between Eucalyptus components and the Vserver. The SC must be able communicate with the Vserver over both management and data LIFs. The NC must be able to communicate with the Vserver using the data LIFs.

9. Configure the SC to use the NetApp SAN for storage:

```
euca-modify-property -p <partition>.storage.blockstoragemanager=netapp
```

The output of the command should be similar to:

```
PROPERTY <partition>.storage.blockstoragemanager netapp was <unset>
```

10. Verify that the propery value is now: 'netapp'

```
euca-describe-properties | grep blockstorage
```

11. On the CLC, run the following command to verify that the SC is listed; note that it may be in the BROKEN state:

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

12. Wait for the SC to transition to NOTREADY or DISABLED states.

13. On the CLC (the primary CLC in an HA setup), enable NetApp SAN support in Eucalyptus by entering the Vserver's hostname or IP address, the username and password of the administrator account, CHAP username and Vserver name.

Note: Eucalyptus uses Challenge Handshake Authentication Protocol (CHAP) for disk operations. The CHAP username can be any value, however it should be unique when sharing a NetApp Vserver across multiple Eucalyptus clusters.

Note: CHAP support for NetApp has been added in Eucalyptus 3.3. The SC will not transition to ENABLED state until the CHAP username is configured.

```
euca-modify-property -p <partition>.storage.sanhost=<Vserver_IP_address>
euca-modify-property -p <partition>.storage.sanuser=<Vserver_admin_username>
euca-modify-property -p
<partition>.storage.sanpassword=<Vserver_admin_password>
euca-modify-property -p <partition>.storage.chapuser=<Chap_username>
```

Note: The following command may fail if tried immediately after configuring the block storage manager. Retry the command a few times, pausing for a few seconds after each retry:

```
euca-modify-property -p <partition>.storage.vservername=<Vserver_name>
```

14. Wait for the SC to transition to ENABLED state.

Note: The SC must be in the ENABLED state before configuring the following properties.

15. If no aggregate is set, Eucalyptus will query the NetApp Vserver for all available aggregates and use the one that has the highest capacity (free space) by default. To make Eucalyptus use specific aggregate(s) configure the following property:

```
euca-modify-property -p <partition>.storage.aggregate=<aggregate_1_name,
        aggregate_2_name,...>
```

If you want Eucalyptus to use the smallest aggregate first configure the following property:

```
euca-modify-property -p <partition>.storage.uselargestaggregate=false
```

16. Set an IP address for the iSCSI data LIF on the ENABLED CLC. This is used for NCs performing disk operations on the Vserver. If you want to configure multiple IPs, see see *Configure NetApp Multipathing*.

```
euca-modify-property -p <partition>.storage.ncpaths=<ip>
```

17. Set an IP address for the iSCSI data LIF on the ENABLED CLC. This is used by the SC for performing disk operations on the Vserver. The SC connects to the data LIFs on the Vserver in order to transfer snapshots to Walrus during snapshot operations. If you want to configure multiple IPs, see *Configure NetApp Multipathing*.

```
euca-modify-property -p <partition>.storage.scpaths=<ip>
```

Your NetApp Clustered Data ONTAP SAN is now ready to use with Eucalyptus.

Enable EMC VNX SANs

This adapter uses the newer VNX-Snapshot feature available on VNX devices running FLARE v5.32 or later that have a VNX-Snapshot license. This adapter also requires the Navisphere Secure CLI to be installed on the SCs. The Navisphere CLI must be version 7.32.0.5.54 or later.

Important: You must create a Clone Private LUN (CPL) of at least 1GB on each SP. For more information on creating private LUNs, go to *Allocating clone private LUNs*. Please note that to view this documentation you will need to register for an EMC account.

1. We assume that the Navisphere CLI is installed in `/opt/Navisphere` on the SC.

   Important: Eucalyptus currently supports version 7.32.0.5.54 or later of the Navisphere CLI.

2. Verify that the CLI is installed and can communicate with the VNX from the SCs.

   On each SC that you are configuring, test the naviseccli command as follows:

```
/opt/Navisphere/bin/naviseccli -User <your SAN username> -Password <your SAN
 password> -Scope 0 -Address <management port IP> connection -pingnode -address
 <a data port IP on your VNX>
```

   Verify that the command runs successfully and the ping gets replies from the SAN.

3. On the CLC, run the following command to verify that the SC is listed; note that it may be in the `BROKEN` state:

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

4. Configure the SC to use the EMC VNX VNX-Snapshot-based manager for storage.

```
euca-modify-property -p <partition>.storage.blockstoragemanager=emc-vnx
```

   The output of the command should be similar to:

```
PROPERTY PARTI00.storage.blockstoragemanager emc-vnx was <unset>
```

5. Check the SC to be sure that it has transitioned out of the BROKEN state and is in either NOTREADY or DISABLED before configuring the rest of the properties for the SC. The following commands should be run on the ENABLED CLC to configure the SC.

   On the ENABLED CLC, run:

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

6. On the CLC (the primary CLC in an HA setup), enable SAN support in Eucalyptus by entering your SAN's hostname or IP address, the username, and password:

```
euca-modify-property -p [partition_name].storage.sanhost=[SAN_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]
```

   If you have multiple management IP addresses for the SAN adapter, provide a comma-delimited list of IP addresses to the `[partition_name].storage.sanhost` property.

7. On the ENABLED CLC, set the login scope for the command line access. For most installs, the login scope will be `0`, which indicates a global login scope for the device. `1` indicates a local scope. `2` indicates LDAP authentication for the SAN device. Use login scope value of `2` only if your SAN is configured to use LDAP authentication and you have an admin user configured to use LDAP.

```
euca-modify-property -p <partition_name>.storage.loginscope=<login_scope>
```

8. On the ENABLED CLC, set the username for the Challenge Handshake Authentication Protocol (CHAP). This can be any value, however it should be unique when sharing VNX on multiple Eucalyptus clusters.

```
euca-modify-property -p <partition_name>.storage.chapuser=<chap_username>
```

9. On the ENABLED CLC, set the value for the unique storage pool that you have configured to use with the SC.

```
euca-modify-property -p <partition_name>.storage.storagepool=0
```

10. On the ENABLED CLC, set the iSCSI data port IP for NCs to use to perform disk operations on the SAN. If you want to configure multiple IPs, see *Configure EMC VNX Multipathing*.

```
euca-modify-property -p <partition_name>.storage.ncpaths=<ip>
```

11. On the ENABLED CLC, set the iSCSI data port IP for SCs to use to perform disk operations on the SAN. The SCs connect to the data ports on the SAN in order to transfer snapshots to Walrus during snapshot operations. If you want to configure multiple IPs, see the section on 'multipathing.

```
euca-modify-property -p <partition_name>.storage.scpaths=<ip>
```

12. On the ENABLED CLC, set the path to Navisphere CLI that you downloaded earlier to the SC. The following example shows the default path. This is that path on the SC, not on the CLC.

```
euca-modify-property -p
<partition_name>.storage.clipath=/opt/Navisphere/bin/naviseccli
```

Your EMC VNX SAN is now ready to use with Eucalyptus.

Tip: Note: The time it takes for a LUN migration to complete will depend on the exact VNX model, workload, and volume size, and the amount of data actually stored in the volume. The default timeout for LUN migrations is 12 hours. If your deployment uses volumes >50GB, or if you find that snapshots fail and a "migration timeout" message is seen in the SC logs, then you should increase the timeout to a larger value. It is recommended that if you plan on using volumes in the 100GB range that you set that timeout to 3600 or larger. You can set the timeout using `euca-modify-property` as follows:

```
euca-modify-property -p [partition].storage.lunmigrationtimeout=[time in
  hours]
```

Configure Dell Equallogic Multipathing

Use multipathing to provide network-and-SP-redundancy for the iSCSI data path between the Dell Equallogic SAN and NCs.

Important: It is **strongly** recommended that you get your system up and running and able to create volumes and snapshots using the Dell Equallogic SAN prior to configuring multipathing. Multipathing can be configured after the cloud is fully functional and will apply to any volumes attached/snapshotted after multipathing is configured. Configuring multipathing on a non-multipathed system does not require a restart of the SC, NC, or CLC.

Important: The Dell Equalogic SAN has separate paths for data and management.

The Dell Equallogic management interface is available for executing control operations only. If your Dell Equallogic SAN is configured to use the management port, please note the IP address of the management interface. The SC can be configured to use the management interface by specifying the IP address of the management interface using the `scpaths` property. For example:

```
euca-modify-property -p mypartition.storage.scpaths=192.168.3.1
```

The Dell Equallogic data interface is configured by specifying the IP address of the data interface using the `ncpaths` property. For example:

```
euca-modify-property -p mypartition.storage.ncpaths=192.168.3.1
```

To configure multipathing for a Dell Equallogic SAN:

1. Ensure that the mutipathd service is running on each NC:

   ```
   mpathconf --enable
   ```

2. Configure the /etc/multipath.conf file.

   Note: An example configuration for the Dell Equallogic SAN is installed with Eucalyptus. This file is located in `/usr/share/doc/eucalyptus-3.3.0/multipath.conf.example.equallogic` on each SC. You may need to copy that file to each NC.

3. Start the mutipathd service:

   ```
   service multipathd start
   ```

4. If you modify the /etc/multipath.conf file, be sure to restart and reconfigure multipathd:

   a) Restart the multipathd service:

   ```
   service multipathd restart
   ```

   b) Run multipathd -k:

   ```
   multipathd -k
   ```

   c) Enter the following commands at the multipathd interactive prompt:

   ```
   reconfigure
   quit
   ```

5. Check that the multipath udev rules file is installed by verifying that the file `/etc/udev/rules.d/12-dm-permissions.rules` file exists.

6. Set the ISCSI paths:

   Note: The path specification format is `iface0:ip0,iface1:ip1,...,ifaceN:ipN` where iface may be reused (i.e. multiple iface0 entries). Also, note that 'iface' is optional, you may just specify a comma-delimited list of IPs. Eucalyptus will detect which interfaces on the SC/NC can reach each specified IP and will use the first found. You must only specify the 'iface' value if you want precise control over which interfaces access which IPs. For using a single path only, just specify the IP of the iscsi data port to use on the VNX.

   Note: We recommend initially getting the system working with only one path. The path values can be modified at any time to enable multipathing, so it is possible to get everything working and confirm full VNX functionality before attempting multipathing. To use one path, simply specify a single IP for each the following steps.

   a) Set the NC paths. For example:

   ```
   euca-modify-property -p mypartition.storage.ncpaths=iface0:127.0.0.1
   ```

   b) Set the SC paths. For example:

   ```
   euca-modify-property -p mypartition.storage.scpaths=iface0:127.0.0.1
   ```

   Note: The NC and SC may each have different path lists, or you can optionally only enable multipathing on the NCs or SC if desire.

   c) If you specified an `iface` when setting the SC paths, be sure to include a line in the eucalyptus.conf file of each NC in the cluster that defines each `iface`. For example:

   ```
   STORAGE_INTERFACES="iface0=eth0"
   ```

7. Test and verify the configuration by creating (and attaching to) a volume and creating a snapshot on the partition.
8. If testing is successful, you can now configure multiple paths in your *.storage.ncpaths and *.storage.scpaths configuration. In the following example, the IP addresses for each interface correspond to the paths configured on the Dell Equallogic SAN:

```
euca-modify-property -p
mypartition.storage.ncpaths=iface0:192.168.1.1,iface1:192.168.1.2
```

9. Verify that multipathing is working on an NC by attaching a volume to an instance on that NC and running the following command:

```
 multipath -ll
```

   This command should return output similar to the following:

```
mpathb (36006016098b0300080722f971b2ee211) dm- 0 DGC,VRAID
size=1.0G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|- +-  policy='round- robin 0' prio=50 status=active
| `-  6:0:0:1 sdd 8:48 active ready running
`- +-  policy='round- robin 0' prio=10 status=enabled
`-  7:0:0:1 sdf 8:80 active ready running
```

You have now successfully configured multipathing for your Dell Equallogic SAN installation.

Configure EMC VNX Multipathing

Use multipathing to provide network-and-SP-redundancy for the iSCSI data path between the EMC VNX SAN and NCs.

Important: It is **strongly** recommended that you get your system up and running and able to create volumes and snapshots using the EMC VNX prior to configuring multipathing. Multipathing can be configured after the cloud is fully functional and will apply to any volumes attached/snapshotted after multipathing is configured. Configuring multipathing on a non-multipathed system does not require a restart of the SC, NC, or CLC.

To configure multipathing for a EMC VNX SAN:

1. Ensure that the mutipathd service is running on each NC:
   `mpathconf --enable`
2. Configure the /etc/multipath.conf file.

   Note: An example configuration for EMC VNX is installed with Eucalyptus. This file is located in `/usr/share/doc/eucalyptus-3.3.0/multipath.conf.example.vnx` on each SC. You may need to copy that file to each NC.

3. Start the mutipathd service:

```
service multipathd start
```

4. If you modify the /etc/multipath.conf file, be sure to restart and reconfigure multipathd:
   a) Restart the multipathd service:

```
service multipathd restart
```

   b) Run multipathd -k:

```
multipathd -k
```

   c) Enter the following commands at the multipathd interactive prompt:

```
reconfigure
quit
```

5. Check that the multipath udev rules file is installed by verifying that the file `/etc/udev/rules.d/12-dm-permissions.rules` file exists.
6. Set the ISCSI paths:

   Note: The path specification format is iface0:ip0,iface1:ip1,...,ifaceN:ipN where iface may be re- used (i.e. multiple iface0 entries). Also, note that ‘iface’ is optional, you may just specify a comma- delimited list of IPs. Eucalyptus will detect which interfaces on the SC/NC can reach each specified IP and will use the first found. You must only specify the ‘iface’ value if you want precise control over which interfaces access which IPs. For using a single path only, just specify the IP of the iscsi data port to use on the VNX.

   Note: We recommend initially getting the system working with only one path. The path values can be modified at any time to enable multipathing, so it is possible to get everything working and confirm full VNX functionality before attempting multipathing. To use one path, simply specify a single IP for each the following steps.

   a) Set the NC paths. For example:
      `euca-modify-property -p mypartition.storage.ncpaths=iface0:127.0.0.1`
   b) Set the SC paths. For example:
      `euca-modify-property -p mypartition.storage.scpaths=iface0:127.0.0.1`

      Note: The NC and SC may each have different path lists, or you can optionally only enable multipathing on the NCs or SC if desire.

c) If you specified an `iface` when setting the SC paths, be sure to include a line in the eucalyptus.conf file of each NC in the cluster that defines each `iface`. For example:

```
STORAGE_INTERFACES="iface0=eth0"
```

7. Test and verify the configuration by creating (and attaching to) a volume and creating a snapshot on the partition.
8. If testing is successful, you can now configure multiple paths in your *.storage.ncpaths and *.storage.scpaths configuration.
9. Verify that multipathing is working on an NC by attaching a volume to an instance on that NC and running the following command:

```
multipath -ll
```

This command should return output similar to the following:

```
mpathb (36006016098b0300080722f971b2ee211) dm- 0 DGC,VRAID
size=1.0G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|- +-  policy='round- robin 0' prio=50 status=active
| `-  6:0:0:1 sdd 8:48 active ready running
`- +-  policy='round- robin 0' prio=10 status=enabled
`-  7:0:0:1 sdf 8:80 active ready running
```

You have now successfully configured multipathing for your EMC VNX SAN installation.

Configure NetApp Multipathing

Use multipathing to provide network and controller redundancy for the iSCSI data path between the NetApp Cluster-mode SAN and NCs.

Important: Eucalyptus supports multipathing for NetApp Clustered ONTAP only.

Important: It is **strongly** recommended that you get your system up and running and able to create volumes and snapshots using the NetApp SAN prior to configuring multipathing. Multipathing can be configured after the cloud is fully functional and will apply to any volumes attached/snapshotted after multipathing is configured. Configuring multipathing on a non-multipathed system does not require a restart of the SC, NC, or CLC.

To configure multipathing for a NetApp SAN:

1. Ensure that the mutipathd service is running on each NC:

   ```
   mpathconf --enable
   ```

2. Configure the /etc/multipath.conf file.

   Note: An example configuration for NetApp is installed with Eucalyptus. This file is located in `/usr/share/doc/eucalyptus-3.3.0/multipath.conf.example.netapp` on each SC. You may need to copy that file to each NC.

3. Start the mutipathd service:

   ```
   service multipathd start
   ```

4. If you modify the /etc/multipath.conf file, be sure to restart and reconfigure multipathd:

   a) Restart the multipathd service:

   ```
   service multipathd restart
   ```

b) Run multipathd -k:

```
multipathd -k
```

c) Enter the following commands at the multipathd interactive prompt:

```
reconfigure
quit
```

5. Check that the multipath udev rules file is installed by verifying that the file `/etc/udev/rules.d/12-dm-permissions.rules` file exists.
6. Set the ISCSI paths:

Note: The path specification format is iface0:ip0,iface1:ip1,...,ifaceN:ipN where iface may be re- used (i.e. multiple iface0 entries). Also, note that 'iface' is optional, you may just specify a comma- delimited list of IPs. Eucalyptus will detect which interfaces on the SC/NC can reach each specified IP and will use the first found. You must only specify the 'iface' value if you want precise control over which interfaces access which IPs. For using a single path only, just specify the IP of the iscsi data port to use on the NetApp Clustered ONTAP.

Note: We recommend initially getting the system working with only one path. The path values can be modified at any time to enable multipathing, so it is possible to get everything working and confirm full NetApp functionality before attempting multipathing. To use one path, simply specify a single IP for each the following steps.

a) Set the NC paths. For example:

```
euca-modify-property -p mypartition.storage.ncpaths=iface0:127.0.0.1
```

b) Set the SC paths. For example:

```
euca-modify-property -p mypartition.storage.scpaths=iface0:127.0.0.1
```

Note: The NC and SC may each have different path lists, or you can optionally only enable multipathing on the NCs or SC if desire.

c) If you specified an `iface` when setting the SC paths, be sure to include a line in the eucalyptus.conf file of each NC in the cluster that defines each `iface`. For example:

```
STORAGE_INTERFACES="iface0=eth0"
```

7. Test and verify the configuration by creating (and attaching to) a volume and creating a snapshot on the partition.
8. If testing is successful, you can now configure multiple paths in your *.storage.ncpaths and *.storage.scpaths configuration.
9. Verify that multipathing is working on an NC by attaching a volume to an instance on that NC and running the following command:

```
 multipath -ll
```

This command should return output similar to the following:

```
mpathp (3600a098037542d69535d43514965354e) dm-2 NETAPP,LUN C-Mode
size=2.0G features='3 queue_if_no_path pg_init_retries 50' hwhandler='1 alua'
 wp=rw
|-+- policy='round-robin 0' prio=50 status=active
| |- 18:0:0:0 sdd 8:48 active ready running
| `- 20:0:0:0 sdf 8:80 active ready running
`-+- policy='round-robin 0' prio=10 status=enabled
```

```
  |- 17:0:0:0 sdc 8:32 active ready running
  `- 19:0:0:0 sde 8:64 active ready running
```

You have now successfully configured multipathing for your NetApp Clustered ONTAP system.

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:

1. Enter the following command on the CLC (the primary CLC in an HA setup):

```
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.

2. If you wish to modify the subdomain that is reported as part of the instance DNS name, please enter the following command:

```
euca-modify-property -p
cloud.vmstate.instance_subdomain=.<custom-dns-subdomain>
```

When this value is modified, the public and private DNS names reported for each instance will contain the specified custom DNS subdomain name, instead of the default value, which is `eucalyptus`. For example, if this value is set to `foobar`, the instance DNS names will appear as `euca-A.B.C.D.foobar.<subdomain>`.

Enable DNS Delegation

For HA: If you are not using HA, you can skip this task. If you are using HA and do not enable DNS delegation, you must manually update `EC2_URL`, `S3_URL` and `EUARE_URL` to point to the new primary hosts in case of failover.

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:

1. On the primary CLC, enter the following command:

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

2. Because the credentials are now slightly changed, you must generate the administrative credentials and source the `eucarc` file again. For more information, see *Generate Administrator Credentials*.

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.

Experimental: Advanced DNS options

Recursive lookups and split-horizon DNS are available in Eucalyptus on an experimental basis. No support is provided for these features

1. To enable any of the experimental DNS resolvers, first set `experimental.dns.enabled` to `true`.
2. To enable the experimental recursive DNS resolver, set `experimental.dns.recursive.enabled` to `true`.
3. To enable split-horizon DNS resolution for internal instance public DNS name queries, set `experimental.dns.split_horizon.enabled` to `true`.

Configuring Node Controller

To alleviate potential problems, 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. 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. 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.
3. Set `DISABLE_KEY_INJECTION=1` to disable key injection. By default, the node controller uses the filesystem to perform key injection. This is potentially an unsafe practice.

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 Eucalyptus Administrator Console (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.

Tip: Consider backing the DRBD resource with a logical volume using LVM, this will make growing the backing store easier in the future if you are running low on disk space.

Configuring DRBD requires that you 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 (output of `uname -n`) 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 (output of `uname -n`) 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 and Ubuntu, remove the `common` section (`common { protocol C; }`). The configuration in these distributions already include 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";
```

Important: For Ubuntu, you must also remove the `common` section (`common { protocol C; }`) and the line `include "drbd.d/*.res";`.

7. Save and close the file.
8. On both primary and secondary Walrus, open the file `/etc/fstab` and append the following line for the DRBD resource to allow the `eucalyptus` user to mount and unmount the device:

```
/dev/drbd1 /var/lib/eucalyptus/bukkits ext3 noauto,owner 0 0
```

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

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

11. Copy the `/etc/drbd.conf`, the `/etc/eucalyptus/drbd.conf`, and the `/etc/eucalyptus/eucalyptus.conf` files to the secondary Walrus server.
12. Restart Walrus, first on the primary, and then on the secondary. Restarting the primary Walrus will trigger an HA failover to the secondary, and restarting the secondary will fail back, preparing the entire system for the next steps.

```
service eucalyptus-cloud restart
```

13. Monitor the failover with euca-describe-services. If successful, stop the eucalyptus-cloud service again on both Walruses whilst you configure the DRBD device.

```
service eucalyptus-cloud stop
```

14. 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 up r0
```

Important: Repeat this step on the secondary Walrus.

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

Tip: With a large DRBD device, the initial synchronization can take a considerable amount of time. Consult *Skip Initial Device Synchronization* for instructions on how to skip the synchronization.

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

16. 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`.

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

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

18. With the DRBD device now configured, start the eucalyptus-cloud service on both Walruses.

```
service eucalyptus-cloud start
```

19. 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
```

Skip Initial Device Synchronization

For HA: Only perform these steps if you have no existing data on your DRBD devices. We assume you have completed the steps up to and including step *11* in *Configure DRBD*.

1. Before promoting the DRBD resource to primary, generate a new UUID and clear the bitmap on the primary Walrus ONLY.

```
drbdadm -- --clear-bitmap new-current-uuid r0
```

2. On the same Walrus, promote the resource to primary:

```
drbdadm primary r0
```

3. Resume the configuration at step *15* in *Configure DRBD*.

Synchronize Configuration in Eucalyptus HA

This section provides guidance on how to keep configuration files sychronized between Eucalyptus HA component pairs. The csync2 utility can be used to achieve this, it's designed to keep configuration consistent between systems in large clusters. Two standalone Walrus systems assumed to be in HA are used in the following example.

For HA: Only perform these steps between HA pairs containing the same components to ensure the configuration matches. For assistance in implementing more advanced tasks the csync2 man pages should be consulted directly.

Note: Completion of this section is optional. You may also wish to expand on this example and generate your own more complex configurations for other components, adding more groups and including additional files in the sychronization tasks.

1. Install the csync2 utility via the EPEL repository on both systems:

```
yum -y install csync2
```

2. On the primary system, generate a pre-shared key for the hosts you wish to sync:

```
csync2 -k /etc/csync2/eucalyptus.key
```

3. On the primary system edit the /etc/csync2/csync2.cfg configuration file to create the synchronization template. Use the one below as an example, creating a group for your two Walrus hosts and adding the eucalyptus and drbd configuration files. Replace the host lines below with the full hostnames of your Walrus systems:

```
    group eucalyptus {
    host euca-walrus-ha-1;
    host euca-walrus-ha-2;
    key /etc/csync2/eucalyptus.key;
    include /etc/csync2/csync2.cfg;
    include /etc/eucalyptus/eucalyptus.conf;
    include /etc/eucalyptus/drbd.conf;
    include /etc/drbd.conf;
    }
```

4. Copy the base configuration file and pre-shared key generated earlier to the secondary Walrus server:

```
scp /etc/csync2/csync2.cfg /etc/csync2/eucalyptus.key
euca-walrus-ha-2:/etc/csync2/
```

5. Next, ensure the csync2 service and xinetd are set to start at boot on both Walrus systems:

```
  chkconfig csync2 on
  chkconfig xinetd on
  service xinetd restart
```

6. Next, perform a dry-run sync to check for proposed changes. Run this on your primary Walrus.

```
csync2 -xvd
```

7. If you are happy with the proposed changes, go ahead and perform the sync.

```
csync2 -xv
```

8. With the initial synchronization complete, a periodic compare and sync could be run via a cronjob which runs the csync2 utility every 5 minutes. In the future, changes can be made to the synchronized files on your primary Walrus and these would be propogated to the other system by the cronjob. Add the following to /etc/cron.d/csync2 as root on the primary Walrus:

```
*/5 * * * * root csync2 -x
```

Configure VMware Support

After registering the VMware Broker, as described above, the Broker component will be activated, but will not be usable (i.e., listed as `ENABLED`) until it is configured with information about your VMware infrastructure. An unconfigured Broker is as good as a Cluster Controller with no Node Controllers to deploy virtual machines on. Until the Broker is properly configured, its logs (e.g., `cloud-output.log`) will contain a reminder of the fact:

```
VMware Broker has not been configured (see euca-configure-vmware)
```

Configuration for the VMware Broker is described by an XML document. A minimal configuration, which would supply just enough information for the Broker to become usable, can be generated automatically, by answering a set of questions about your VMware endpoints. All further configuration must be done by editing the XML document manually, though with help from a validation mechanism. We recommend starting with a minimal configuration and editing the generated document to further expand it.

The steps for creating minimal and full-featured configurations, as well as for validating them, are described next. All these steps involve `euca-configure-vmware` command, which must be executed on the CC/Broker host. For authorization, the same type of credentials that other administrative `euca-` commands require must be supplied (e.g., via `eucarc`). If CLC and CC/Broker run on different hosts, the credentials may have to be copied from the CLC host to the CC/Broker host.

Minimal VMware Broker configuration

At the very least, a VMware Broker needs the IP addresses and access credentials of each VMware endpoint (either vCenter or ESX/ESXi host). To create a minimal configuration automatically, this information must be entered, for each endpoint, when prompted by `euca-configure-vmware` command. If the Broker has never been configured, the command will detect that and will ask for information upon invocation without any flags.

1. On the CC/Broker host, enter the following command:

```
euca-configure-vmware
```

The output of the above command prompts for the same parameters that the vSphere Client application, distribued by VMware, requests at startup.

2. Enter the requested parameters, making sure to specify just the IP addresses of VMware endpoints and not URLs. If you want to use vCenter, then enter the IP address of the vCenter server. If you do not want to use vCenter, then enter IP addresses of each ESX/ESXi host. We recommend using vCenter because it is easier to configure and can be more efficient.

```
Please, supply vSphere endpoint IP: 192.168.51.77
Please, supply vSphere username: root
Please, supply vSphere password:
Do you want to enter another endpoint? [N]: y
Please, supply vSphere endpoint IP: 192.168.51.78
Please, supply vSphere username [root]:
Please, supply vSphere password [*****]:
Do you want to enter another endpoint? [N]: N
```

After entering all vSphere endpoint information, if the access credentials are correct, you should see output similar to the following:

```
discovered 2 host(s)
        192.168.51.78 login=root datastoreName=datastore1 (7)
uploadViaHost=true network=VM Network
        192.168.51.77 login=root datastoreName=datastore1 (6)
uploadViaHost=true network=VM Network
```

If vCenter endpoint is entered, the output may list multiple ESX(i) hosts that were discovered by querying vCenter:

```
Please, supply vSphere endpoint IP: 192.168.51.48
Please, supply vSphere username: Administrator
Please, supply vSphere password:
Do you want to enter another endpoint? [N]:
discovered 7 host(s)
        192.168.51.175 login=Administrator datastoreName=datastore1
uploadViaHost=null network=VM Network
        192.168.51.24 login=Administrator datastoreName=datastore1 (3)
uploadViaHost=null network=VM Network
        192.168.51.22 login=Administrator datastoreName=datastore1 (5)
uploadViaHost=null network=VM Network
        192.168.51.78 login=Administrator datastoreName=datastore1 (7)
uploadViaHost=null network=VM Network
        192.168.51.18 login=Administrator datastoreName=datastore1 (4)
uploadViaHost=null network=VM Network
        192.168.51.77 login=Administrator datastoreName=datastore1 (6)
uploadViaHost=null network=VM Network
        192.168.51.116 login=Administrator datastoreName=datastore1 (1)
uploadViaHost=null network=VM Network
```

This process both generates the XML configuration and configures the Broker. From this point onward, invoking `euca-configure-vmware` with no parameters will cause the current configuration of the Broker to be validated. To make the new configuration active, the Broker must be restarted.

3. Restart the VMware Broker.

```
service eucalyptus-cloud restart
```

Re-generating VMware Broker configuration

After the Broker has been configured, to generate a configuration again, one must use a two-step process:

1. On the CC/Broker host, use the `--generate` flag to create another configuration, which is saved in an XML file in the `/tmp` directory.

```
euca-configure-vmware --generate
```

   Note the path to the newly generated XML configuration that is printed by the command.

```
Please, supply vSphere endpoint IP: 192.168.51.116
Please, supply vSphere username: root
Please, supply vSphere password:
Do you want to enter another endpoint? [N]:
discovered 1 host(s)
        192.168.51.116 login=root datastoreName=datastore1 (1)
uploadViaHost=true network=VM Network
New config file was saved to /tmp/euca_vmwarexsiVPj.xml
```

2. Modify the configuration in Broker's database by providing that file to `euca-configure-vmware`:

```
euca-configure-vmware /tmp/euca_vmwarexsiVPj.xml
```

   The XML document is validated by contacting the vSphere endpoints and some diagnostic information is reported.

```
Network mode: MANAGED
discovered 1 host(s)
        192.168.51.116 login=root datastoreName=datastore1 (1)
uploadViaHost=true network=VM Network
```

3. Restart the VMware Broker.

```
service eucalyptus-cloud restart
```

Full-featured VMware Broker configuration

This section may be skipped if the minimal configuration produced automatically was sufficient to access all hypervisor nodes and the default names chosen for networks and datastores were adequate. If that is not the case, the configuration, in the form of an XML document, will have to be edited manually.

1. There are two ways to edit the XML document:

   - By invoking `euca-configure-vmware` with `--edit` flag, which invokes an editor (as specified by the `$EDITOR` environment variable, which must be set for the flag to work), with current configuration loaded in it, and updates the configuration when the editor terminates successfully.

```
euca-configure-vmware --edit
```

   - By editing an XML file out of band and providing `euca-configure-vmware` with the path to the file, which is then used to update the configuration of the Broker.

```
euca-configure-vmware /path/to/file.xml
```

   In both cases, before the configuration is updated, the XML document is validated for correctness, both in terms of XML syntax and in the validity of information provided therein with respect to the VMware infrastructure (i.e.,

endpoints, access credentials, and any named resources, such as networks and datastores, are verified by requests to VMware).

```
Network mode: MANAGED
discovered 1 host(s)
        192.168.51.116 login=root datastoreName=datastore1 (1)
uploadViaHost=true network=VM Network
```

2. Restart the VMware Broker.

```
service eucalyptus-cloud restart
```

XML configuration structure

The part of the document that describes vSphere endpoints can be hierarchical, reflecting the hierarchy of abstractions defined within vSphere: endpoints may contain datacenters, datacenters may contain clusters, and clusters may contain hosts. However, just as parts of the hierarchy are optional in vSphere (e.g., there may be one default datacenter and no clusters) the hierarchy is optional in the VMware Broker configuration.

The only required element is `<endpoint/>`, which must be enclosed by the `<vsphere/>` element, which in turn must be enclosed by the `<configuration/>` element. These requirements are satisifed by any minimal configuration, as generated by the steps described above. Minimal configurations typically look as follows:

```
<configuration>
    <vsphere>
        <endpoint
           url="https://192.168.51.116/sdk"
           login="root"
           password="RSA/ECB/PKCS1PaddingDYGIG..."
           discover="true"/>
    </vsphere>
</configuration>
```

When other elements are present, however, they must be arranged relative to each other in a hierarchy. This hierarchy is shown in the following template, which describes all valid elements in a VMware Broker configuration and their attributes (some attributes are grouped into categories, namely `CREDENTIALS` and `EXTRAS`).

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

For example, if a `<datacenter/>` is specified, it must be contained by the `<endpoint/>` to which it belongs. Likewise, any `<cluster/>` must be contained within an `<endpoint/>`, if any. And so on. All endpoints must be contained by the single `<vsphere/>` element. These elements and attributes will be discussed below.

XML configuration attributes

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

`CREDENTIALS` and `EXTRAS` are categories of attributes. These attributes can be specified for any vSphere-related element with values propagating from higher-level elements to lower-level elements, where the values can be overridden selectively. For example, if one were to specify `maxCores="4"` in the `<endpoint/>` element, then all hosts belonging to that endpoint would advertise 4 cores instead of their actual number of physical cores. However, the lower-level parameter always overrides the higher-level parameter. So, if a `<host/>` specifies `maxCores="8"`, that will override `maxCores="4"` specified in the `<endpoint/>` or `<datacenter/>` that contains it. This kind of inheritance of values with possibility of overriding applies to all attributes in `CREDENTIALS` and `EXTRAS` categories.

- **`CREDENTIALS`** consist of `'login'` and `'password'` attributes, the latter of which can be specified in plaintext or encrypted (as produced by `euca-configure-vmware`). At the very least they must be specified either for each `<endpoint/>` or once in the enclosing `<vsphere/>` element, in which case they will be used for all endpoints without explicitly specified credentials. If credentials are specified for any elements contained by `<endpoint/>`, they will be used for the optional data transfer connections to individual ESX/ESXi hosts (see `uploadViaHost` attribute below). Thus, if login or password on ESX/ESXi hosts are different from login and password on vCenter, the values for ESX/ESXi must be specified separately.
- **`EXTRAS`** attributes allow one to restrict Eucalyptus's behavior in several ways. By default, Eucalyptus will attempt to use all resources that it discovers, such as memory, cores, and storage space on a datastore. Furthermore, when multiple options are available, e.g., for a datastore or a network, it will make an arbitrary choice. With the following attributes, one can make the exact choices when desired:
  - `'datastore'` - name of the vSphere datastore to use (first one found by default).
  - `'network'` - name of the vSphere network to use (first one found by default).
  - `'maxCores'` - number of virtual cores to use on an ESX(i) host for Eucalyptus instances (same as physical cores by default).
  - `'maxMemMB'` - memory, in MB, to use on an ESX(i) host for Eucalyptus instances (same as physical RAM by default).
  - `'maxDiskMB'` - disk size, in MB, to use on a datastore for Eucalyptus instances (free space on the datastore by default).
  - `'uploadViaHost'` - upload VM disk contents directly to the ESX(i) host rather than through vCenter ("false" by default). This option is ignored when the endpoint is an ESX(i) host. The default behavior is to upload VM's disk files through vCenter. To avoid overloading the vCenter with I/O traffic, however, Eucalyptus can perform the upload directly to an individual host. In this case, if the credentials (login or password) for the host are different from vCenter credentials, they must be specified explicitly in one or more elements contained by the `<endpoint/>` (e.g., in each `<datacenter/>` or each `<cluster/>` or each `<host/>` element).

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

```
DISALLOWED BY CONFIGURATION
```

Storage attributes

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

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

- `scratchDirectory` and `scratchDirectoryLimitMb` attributes of the optional element `<paths/>` define where on the file system and how much space the VMware Broker may use for non-cacheable work. Default values are `/var/lib/eucalyptus/vmware/tmp` and 50GB, respectively.
- `cacheDirectory` and `cacheDirectoryLimitMb` attributes of the optional element `<paths/>` define where on the file system and how much space the VMware Broker may use for cacheable work. Default values are `/var/lib/eucalyptus/vmware/cache` and 50GB, respectively.

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 based on the size specified in `VNET_ADDRSPERNET`. Note that Eucalyptus reserves eleven addresses per security group, so these networks will be smaller than the value 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: Eleven 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 21 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 configure Eucalyptus to use VLANs within a specified range:

1. Choose your range (a contiguous range of VLANs between 2 and 4095).
2. Configure your cluster controllers with a VNET_SUBNET/VNET_NETMASK/VNET_ADDRSPERNET that is large enough to encapsulate your desired range. For example, for a VLAN range of 1024-2048, you could set VNET_NETMASK to 255.254.0.0 to get a large enough network (131072 addresses), and VNET_ADDRSPERNET to 64, to give 2048 possible VLANs.
3. Configure your cloud controller to work within that range. Use the following commands to verify that the range is now set to be 2-2048, a superset of the desired range.

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

4. Constrict the range to be within the range that the CC can support as follows:

```
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>
```

This ensures that Eucalyptus will only use tags between 1024 and 2048, giving you a total of 1024 security groups, one VLAN per security group.

Tip: If VMs are already running in the system using a VLAN tag that is outside the range specified by global_min_network_tag-global_max_network_tag, that network will continue to run until all VMs within the network are terminated and the system removes reference to that network. Best practice is to configure these values in advance of running virtual machines.

Configure the Load Balancer

Eucalyptus provides optional support for Load Balancing. In order to use this support, you will need to register the Load Balancer image with the cloud.

Installing and Registering the Load Balancer Image

Eucalyptus provides a tools for installing and registering the Load Balancer image. Once you have run the tool, your Load Balancer will be ready to use.

Run the following command on the machine where you installed the eucalyptus-load-balancer-image package:

```
euca-install-load-balancer --install-default
```

Verify Load Balancer Configuration

If you would like to verify that Load Balancer support is enabled you can list installed Load Balancers. The currently active Load Balancer will be listed as enabled. If no Load Balancers are listed, or none are marked as enabled, then your Load Balancer support has not been configured properly.

1. Run the following command to list installed Load Balancer images:

```
euca-install-load-balancer --list
```

2. You can also check the enabled Load Balancer EMI with:

```
euca-describe-properties loadbalancing.loadbalancer_emi
```

3. If you need to manually set the enabled Load Balancer EMI use:

```
euca-modify-property -p loadbalancing.loadbalancer_emi=emi-12345678
```

Change the Administration Password

Change the default password for the administration user. You can do this using the `euare-usermodloginprofile` or by logging in to the Eucalyptus Administrator Console (https://[CLC_IP_address]:8443). The first time you log in to the console, you are prompted for a new password.

Finding More Information

Read More

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 *User 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.
- The *Euca2ools Reference Guide* describes the Euca2ools commands. Refer to this guide for more information about required and optional parameters for each command.

Get Involved

The following resources can help you to learn more, connect with other Eucalyptus users, or get actively involved with Eucalyptus development.

- The Eucalyptus IRC channel is #eucalyptus on Freenode. This channel is used for real-time communication among users and developers. Information on *how to use the network* is available from Freenode.
- Engage hosts the Eucalyptus knowledge base and discussion forum. This provides user discussions, answers to problem reports, and other communications. Engage is available at *https://engage.eucalyptus.com/*

Appendix A: Upgrading Eucalyptus

This section details the tasks you need to perform in order to upgrade your current version of Eucalyptus.

You can upgrade to Eucalyptus 3.3.2 from either 3.2.2 or 3.3.1. To upgrade from any other version of Eucalyptus, you should first upgrade to either of those versions. Follow the directions in the previous Installation Guide, and then upgrade to 3.3.2 using the directions in this section.

Important: Eucalyptus does not support components that are at different releases, even at the sub-minor level. For example, you cannot have a CLC at 3.3.2 and a Walrus at 3.3.1. Please make sure that you update all Eucalyptus components when you upgrade.

Prepare the Configuration File

Complete the following steps to upgrade to Eucalyptus 3.3.2 on CentOS 6 or RHEL 6.

Note:

You should already have the repositories installed for euca2ools, EPEL, and ELRepo from your 3.2.2 installation. If you do not have these installed, refer to the installation instructions to find out how to add these to your machines.

The steps in this section should be performed on all machines with Eucalyptus installed.

1. Remove any hand-written repository files for earlier versions of Eucalyptus and Euca2ools from `/etc/yum.repos.d`.
2. Install the new Eucalyptus release package on each host that will run a Eucalyptus component:

```
yum install
http://downloads.eucalyptus.com/software/eucalyptus/3.3/centos/6/x86_64/eucalyptus-release-3.3.noarch.rpm
```

3. 

   Tip: It's recommended that you install the new version of Euca2ools, although this is not required. If you don't install the new version of Euca2ools, you will not be able to use new features from the command line.

   Install the new Euca2ools release package on each host that will run a Eucalyptus component:

```
yum --nogpgcheck install
http://downloads.eucalyptus.com/software/euca2ools/3.0/centos/6/x86_64/euca2ools-release-3.0.noarch.rpm
```

4. If you have a Eucalyptus subscription, you will have received a package that grants you access to VMware and SAN components. If you wish to use either of these components, install this package on each host that will run a Eucalyptus component with this command:

```
yum install eucalyptus-enterprise-release-3.3*.noarch.rpm
```

You are now ready to *Shutdown Components*.

Shutdown Components

To shut down Eucalyptus components:

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

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

2. 

Tip:

The following is an example that scripts the individual steps to shut down Eucalyptus components 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.

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

Before upgrading Eucalyptus packages, we suggest fully updating your systems using `yum update` where possible.

Note: When using Walrus in a high availability (HA) configuration, *mount(8)* is now used for mounting and unmounting the DRBD device, instead of *mount(2)*. If Walrus HA is configured, add an entry to the `/etc/fstab` to mount the DRBD device to `/var/lib/eucalyptus/bukkits` on both primary and secondary Walrus. For example, if the DRBD device in the Eucalyptus DRBD resource file is defined as `/dev/drbd1` and the filesystem format is ext3, add the following line to `/etc/fstab`:

```
/dev/drbd1 /var/lib/eucalyptus/bukkits ext3 noauto,owner 0 0
```

.

To upgrade Eucalyptus packages:

1. Enter the following command on each machine running a Eucalyptus component:

```
yum clean expire-cache
```

2. Enter the following command on each machine running a Eucalyptus component:

```
yum update 'eucalyptus*'
```

If you have previously customized your configuration files, `yum` returns a warning, and installs the new configuration files with a different name. This preserves your customizations. Before you continue, customize and rename the new Configuration files.

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 $( rpm -qa | grep euca )' ; done
```

3. If you are a subscriber and use Equallogic SAN, run the following command on each machine hosting a CLC:

Important: Skip this step if you are upgrading from 3.3.1.

```
yum install eucalyptus-enterprise-storage-san-equallogic-libs
```

You are now ready to *Upgrade Euca2ools Packages*.

Upgrade Euca2ools Packages

To use the new features available in Eucalyptus 3.3.2, you must upgrade to the latest version of the Euca2ools packages:

1. Enter the following command on each machine running a Eucalyptus component:

```
yum clean expire-cache
```

2. Enter the following command on each machine running a Eucalyptus component:

```
yum update euca2ools
```

You are now ready to *Start Eucalyptus*.

Start Eucalyptus

Note: Eucalyptus 3.3 requires version 7 of the Java Virtual Machine. Make sure that your CLOUD_OPTS settings in the /etc/eucalyptus/eucalyptus.conf file either do not set `--java-home`, or that `--java-home` points to a version 7 JVM. This needs to happen before services are started but after the upgraded packages are installed.

1. In the CLC, enter the following command.

```
service eucalyptus-cloud start
```

If you are upgrading from 3.2.2 you will see that the process starts the database upgrade. Eucalyptus returns output similar to the following example.

```
Starting Eucalyptus services: Attempting database upgrade from 3.2.2
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326904600...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              3.2.2
# New Version:              3.3.2
# 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
```

If you are upgrading from 3.2.2 you will see that Eucalyptus returns output similar to the following example.

```
Starting Eucalyptus services: Attempting database upgrade from 3.2.2
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905005...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              3.2.2
# New Version:              3.3.2
# 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. If you are using Eucalyptus with VMware support, start the VMware Broker on the CC server by running the following command:

```
service eucalyptus-cloud start
```

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

```
service eucalyptus-cc start
```

5. If you have a multi-cluster setup, repeat the previous step for each cluster.
6. Repeat for each CC server.
7. Log in to the SC server and enter the following command:

```
service eucalyptus-cloud start
```

If you are upgrading from 3.2.2 you will see that Eucalyptus returns output similar to the following example>

```
Starting Eucalyptus services: Attempting database upgrade from 3.2.2
at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1326905005...
#                           UPGRADE INFORMATION
#=============================================================================
# Old Version:              3.2.2
# New Version:              3.3.2
# 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.

You are now ready to *Verify the Components*.

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:

   a)
```
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
 {}
```

   b)  **Note:** If you are a subscriber and use NetApp as the storage SAN backend, the following step is required to enable the SAN support in 3.3.

   On the CLC (the primary CLC in an HA setup), configure the CHAP username to be used by the Eucalyptus. The CHAP username can be any value, however it should be unique when sharing a NetApp Filer across multiple Eucalyptus clusters. The SC will not transition to ENABLED state until the CHAP username is configured. For example:

   `euca-modify-property -p <partition>.storage.chapuser=<Chap_username>`

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
```

You are now ready to *Upgrade Credentials*.

Install the Load Balancer Image

In Eucalyptus 3.3.1 we introduced load balancer support. In order to enable this after upgrade, you must first source your existing credentials, and then follow the instructions for load balancer installation and configuration. For more information, see *Configure the Load Balancer*.

Your upgrade is now complete. If at any point your upgrade failed, see *Dealing with Failed Upgrades*.

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, perform the following steps.

1. Download new credentials.

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

2. Unzip the credentials file.

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

3. Source the `eucarc` file.

```
source eucarc
```

You are now ready to *Install the Load Balancer Image*.

Dealing with Failed Upgrades

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 your earlier version, you can find your preserved data in this directory.

If the upgrade fails, all changes to the database and configuration files will be rolled back. You can retry the upgrade by following the upgrade instructions in the sections, *Shutdown Components* and *Upgrade Eucalyptus Packages*.

If you do not want to continue with the upgrade after a failure, you can downgrade your installation back to the previous version. Please note that downgrade instructions are different, depending on whether your Eucalyptus services are co-located or each run on their own machine. You will need to perform the downgrade for all services running on a single machine at the same time.

The `/var/lib/eucalyptus/db` and `/var/lib/eucalyptus/keys` directories should not be affected by the upgrade. If they have been removed subsequent to the upgrade, you must restore the contents of these directories from your backups before downgrading.

To downgrade from a failed upgrade, perform the tasks listed in the following sections.

Downgrade Eucalyptus

1. Downgrade to the Eucalyptus 3.3.1 release package on each host.

```
yum downgrade
http://downloads.eucalyptus.com/software/eucalyptus/3.3/centos/6/x86_64/eucalyptus-release-3.3.noarch.rpm
```

2. If you have a Eucalyptus subscription, downgrade your subscription release package on each host to the release package you obtained for Eucalyptus 3.3.1.

```
yum downgrade eucalyptus-enterprise-3.3*.rpm
```

3. Expire the cache for the yum repositories on each host.

```
yum clean expire-cache
```

4. Log in to each NC host and downgrade it. To downgrade to a specific version, append the version number to each package name. For example, to downgrade to 3.3.1, add `-3.3.1` to each of the packages in the following list:

```
yum downgrade eucalyptus eucalyptus-gl eucalyptus-admin-tools eucalyptus-nc
python-eucadmin
```

Important:

Use the `yum shell` command for the following instructions. This will allow you to perform more complex transactions that are required for the downgrade.

5. Log in to each machine running a Eucalyptus service and run the following command:

```
yum shell
```

6. Add the transaction commands listed below for each component installed on the machine. If more than one component asks you to use the same transactional command, you only need to specify that command once.

CLC Service Transaction Commands:

```
downgrade eucalyptus-cloud
downgrade eucalyptus
downgrade eucalyptus-common-java
downgrade eucalyptus-common-java-libs
downgrade eucalyptus-admin-tools
downgrade python-eucadmin
```

Additional CLC Service Transaction Commands for Eucalyptus Subscription customers:

```
downgrade eucalyptus-enterprise-vmware-broker-libs
downgrade eucalyptus-enterprise-storage-san-common-libs
downgrade eucalyptus-enterprise-storage-san-libs
```

CC Service Transaction Commands:

```
downgrade eucalyptus-cc
downgrade eucalyptus
downgrade eucalyptus-gl
```

```
downgrade eucalyptus-admin-tools
downgrade python-eucadmin
```

SC Service Transaction Commands:

```
downgrade eucalyptus-sc
downgrade eucalyptus
downgrade eucalyptus-common-java
downgrade eucalyptus-common-java-libs
downgrade eucalyptus-admin-tools
downgrade python-eucadmin
```

Walrus Service Transaction Commands:

```
downgrade eucalyptus-walrus
downgrade eucalyptus
downgrade eucalyptus-common-java
downgrade eucalyptus-common-java-libs
downgrade eucalyptus-admin-tools
downgrade python-eucadmin
```

SAN EMC Transaction Commands:

```
remove eucalyptus-enterprise-storage-san-emc
downgrade eucalyptus-enterprise-storage-san-emc-libs
downgrade eucalyptus-enterprise-storage-san-common
downgrade eucalyptus-enterprise-storage-san-common-libs
```

SAN EqualLogic Transaction Commands:

```
downgrade eucalyptus-enterprise-storage-san-equallogic
remove eucalyptus-enterprise-storage-san-equallogic-libs
downgrade eucalyptus-enterprise-storage-san-common
downgrade remove eucalyptus-enterprise-storage-san-common-libs
```

SAN NetApp Transaction Commands:

```
downgrade eucalyptus-enterprise-storage-san-netapp
downgrade eucalyptus-enterprise-storage-san-netapp-libs
downgrade eucalyptus-enterprise-storage-san-common
downgrade eucalyptus-enterprise-storage-san-common-libs
```

VMWare Broker Transaction Commands:

```
downgrade eucalyptus-enterprise-vmware-broker
downgrade eucalyptus-enterprise-vmware-broker-libs
```

7. When all transaction commands have been entered run the following command to verify that the transaction will be successful:

```
ts solve
```

8. Perform the downgrade by running the following command in the transaction shell:

```
run
```

9. Exit the transaction shell using the following command:

```
exit
```

10. Remove the `/etc/eucalyptus/.upgrade` file from each machine:

```
rm /etc/eucalyptus/.upgrade
```

Enter `y` when prompted, to remove this file. It is important to remove this file from every Eucalyptus host.

Prepare System for Upgrade

1. Clear out the `/var/run/eucalyptus/` directory on all machines used for Eucalyptus.
2. Downgrade Euca2ools to 2.1.0.
3. Perform the upgrade tasks for your Eucalyptus version.
4. Start the cloud back up. Make sure all services show `ENABLED`.

Appendix B: Creating a Local Eucalyptus Package Repository

In certain situations (such as installing Eucalyptus from behind a firewall), you might need to install Eucalyptus from a local repository. This section augments the *standard installation instructions* with additional instructions for downloading and installing Eucalyptus from a local repository.

To install Eucalyptus from behind a firewall on CentOS 6 or RHEL 6:

1. Download the Eucalyptus repository to a local directory. For example:

```
wget -r --no-parent \
http://downloads.eucalyptus.com/software/eucalyptus/3.2/centos/6/x86_64/ \
-P /tmp/eucalyptus
```

2. Download euca2ools:

```
wget -r --no-parent \
http://downloads.eucalyptus.com/software/euca2ools/2.1/centos/6/x86_64/ \
-P /tmp/euca2ools
```

3. In step 1 of the *existing installation instructions*, modify the baseurl to point to your Eucalyptus local repository:

```
baseurl=file:///tmp/eucalyptus/downloads.eucalyptus.com/software/eucalyptus/3.2/centos/6/x86_64
```

4. In step 2 of the *existing installation instructions*, modify the baseurl to point to your local Euca2ools repository:

```
baseurl=file:///tmp/euca2ools/downloads.eucalyptus.com/software/euca2ools/2.1/centos/6/x86_64
```

5. Run `yum update`.

Appendix C: Moving to High Availability

With Eucalyptus 3.2 users may register additional components to bring high availability to an existing cloud. This section provides instructions for registering additional Eucalyptus components (specifically Walrus) in an existing environment for the purpose of achieving high availability.

Important: Adding an additional Walrus will require downtime of your cloud platform.

Registering redundant components to your existing Eucalyptus installation will take your platform into a highly available configuration. Registering the Cloud Controller, Cluster Controller and Storage Controller (with SAN adapter only) is fairly trivial but adding an additional Walrus requires that the user configure DRBD for bukkit storage replication.

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

- Installed the same operating system on any additional server you will add for High Availability of a Eucalyptus component.
- Installed and configured Eucalyptus with matching configuration on any additional system you will be adding.
- If registering an additional Walrus, ensure you are able to move the contents of /var/lib/eucalyptus/bukkits/ to a temporary storage area.
- A block device (disk or partition) is available for use as a DRBD device. Consider using LVM for future growth.

Important: The configuration of Highly Availiable components must match. For more information, see *HA Planning*.

1. To register an additional Eucalyptus component please follow the instructions in *Registering Eucalyptus*.
2. Additional steps are required for adding another Walrus component. You will need to configure a DRBD device for the bukkit store. Start by shutting down the eucalyptus-cloud service on both Walrus servers.

```
service eucalyptus-cloud stop
```

3. On the original primary system, copy the current contents of /var/lib/eucalyptus/bukkits to a temporary location.

```
cp -R --preserve /var/lib/eucalyptus/bukkits /newlocation/
```

4. Configure your new DRBD device. For more information, see *Configure DRBD*. After you are finished configuring your device, move to the next step.
5. On the primary Walrus, ensure the new DRBD resource is in a primary state before proceeding.

```
drbd-overview
```

6. On the primary Walrus, copy the preserved contents of /var/lib/eucalyptus/bukkits from the temporary location to the new DRBD device.

Tip: At this point you should run `service eucalyptus-cloud stop` to ensure cloud services are stopped during migration.

```
cp -R --preserve /newlocation/* /var/lib/eucalyptus/bukkits/
```

Tip: If you are migrating large amounts of data, consider skipping initial device synchronization as explained in *Skip Initial Device Synchronization*.

7. On the primary Walrus, monitor the state of the resource with `drbd-overview` to observe the sync. Data will not be replicated until the resource is marked UpToDate/UpToDate.

```
drbd-overview
```

8. With the synchronization complete, start the eucalyptus-cloud service to bring up Walrus.

Your HA environment is now ready.

Appendix D: Installing Standalone Euca2ools

This topic discusses how to perform a standalone installation of Euca2ools.

If you're running recent versions of Fedora, Debian, or Ubuntu, you can install Euca2ools using `yum` or `apt`.

If you're running RHEL/Centos, you can use the following instructions to install Euca2ools.

To perform a standalone installation of Euca2ools on RHEL/CentOS:

1. Configure the EPEL package repository:

```
yum install
http://downloads.eucalyptus.com/software/eucalyptus/3.3/centos/6/x86_64/epel-release-6.noarch.rpm
```

2. Configure the Euca2ools package repository:

```
yum install
http://downloads.eucalyptus.com/software/euca2ools/3.0/centos/6/x86_64/euca2ools-release-3.0.noarch.rpm
```

3. Install Euca2ools:

```
yum install euca2ools
```

You've now performed a standalone installation of Euca2ools.

Index

S

U

V

W