EUCALYPTUS

Eucalyptus 4.2.2 Installation Guide

2016-07-15

Contents

Installation Overview

This topic helps 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.

This guide walks you through installations for a few different use cases. You can choose from one of the installation types listed in the following table.

What Do You Want to Do?	Installation Type
Quickly deploy Eucalyptus on one machine	If you have a CentOS 6.7 minimal install and a few IP addresses to spare, try the FastStart script. Run the following command as root: `bash <(curl -Ls hphelion.com/eucalyptus-install)`
Create a development or production environment	*Eucalyptus Installation*
Upgrade from a previous version of Eucalyptus	*Eucalyptus Upgrade*

We recommend that you read the section you choose in the order presented. There are no shortcuts for installing Eucalyptus, though Eucalyptus FastStart is fairly easy. However, to customize your installation, 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.

Document version: Build 3221 (2016-07-14 22:01:24)

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 use 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

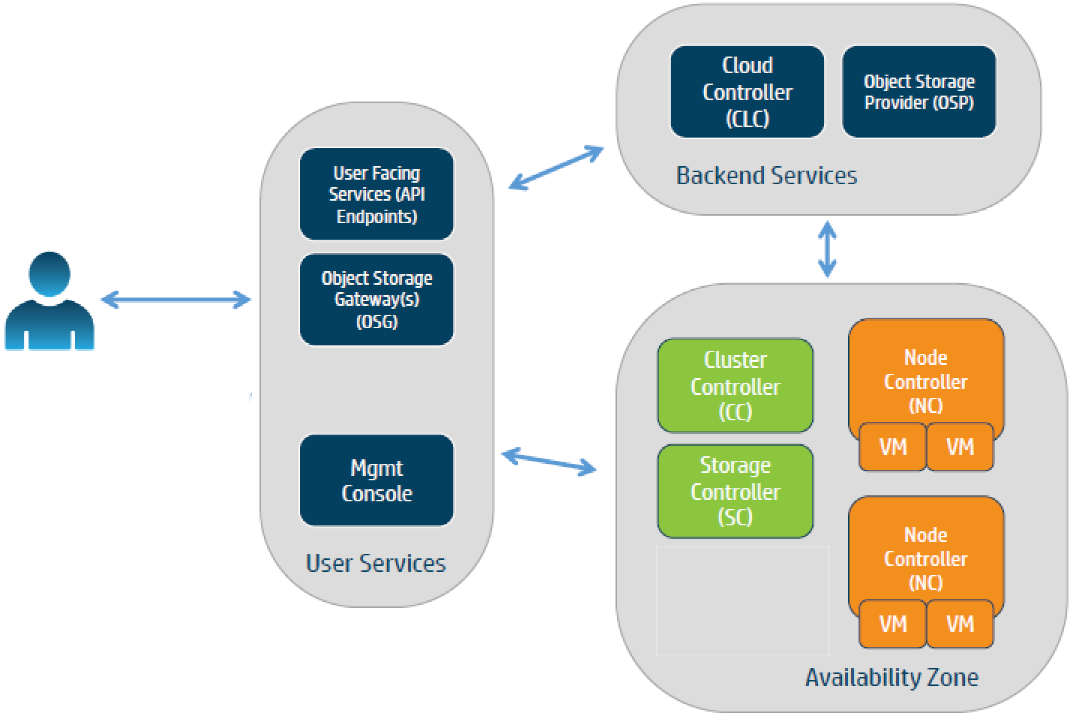
Eucalyptus was designed to be easy to install and as non-intrusive as possible. The software framework is modular, with industry-standard, language-agnostic communication.

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 compatibility with Amazon's EC2, S3, IAM, ELB, Auto Scaling, and CloudWatch services. This offers you the capability of a hybrid cloud.

Eucalyptus Components

This topic describes the various components that comprise a Eucalyptus cloud.

The following image shows an example of Eucalyptus components.



A detailed description of each Eucalyptus component follows.

Cloud Controller

In many deployments, the Cloud Controller (CLC) and the User-Facing Services (UFS) are on the same host machine. This server is the entry-point into the cloud for administrators, developers, project managers, and end-users. The CLC handles persistence and is the backend for the UFS. A Eucalyptus cloud must have exactly one CLC.

User-Facing Services

The User-Facing Services (UFS) serve as endpoints for the AWS-compatible services offered by Eucalyptus: EC2 (compute), AS (AutoScaling), CW (CloudWatch), ELB (LoadBalancing), IAM (Euare), and STS (tokens). A Eucalyptus cloud can have several UFS host machines.

Management Console

The Eucalyptus Management Console is an easy-to-use web-based interface that allows you to manage your Eucalyptus cloud. The Management Console is often deployed on the same host machine as the UFS. A Eucalyptus cloud can have multiple Management Console host machines.

Object Storage Gateway

The Object Storage Gateway (OSG) passes requests to object storage providers and talks to the persistence layer (DB) to authenticate requests. You can use Walrus or Riak CS as the object storage provider. The OSG is part of the UFS group.

Object Storage Provider

The Object Storage Provider (OSP) can be either the Eucalyptus Walrus backend or Riak CS. Walrus is intended for light S3 usage and is a single host. Riak CS is an open source scalable general purpose data platform created by *Basho*; it is intended for deployments with heavy S3 usage.

Cluster Controller

The Cluster Controller (CC) must run on a host 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 in Managed and Managed (No VLAN) *networking modes*. 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 can interface 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 can persist past VM termination and are commonly used to store persistent data. An EBS volume cannot be shared between multiple VMs at once and can be accessed only within the same availability zone in which the VM is running. Users can create snapshots from EBS volumes. Snapshots are stored by the OSG and made available across availability zones. Eucalyptus with SAN support provides the ability to 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 manages the virtual machine networks in Edge *networking mode*. The NC is also responsible for the management of the virtual network endpoint.

System Requirements

To install Eucalyptus, your system must meet the baseline requirements described in this topic.

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. For more information, see the *Eucalyptus Reference Architecture* and look at the physical resources recommended for your deployment type. See the Compatibility Matrix in the *Release Notes* for supported versions.

Compute Requirements

- Physical Machines: All Eucalyptus components must be installed on physical servers, 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 supports only 64-bit architecture.
- Machine Clocks: Each Eucalyptus host machine and any client machine clocks must be synchronized (for example, using NTP). These clocks must be synchronized all the time, not only during the installation process.
- 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 30GB 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 a greater number of VMs.
- Each machine in your network needs a minimum of 4GB 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).
- Some configurations require that machines hosting a CC have two network interfaces, each with a minimum of 1Gb Ethernet.
- For virtual machine traffic isolation, the network ports connecting Ethernet interfaces might need to allow VLAN trunking.
- For *Managed* and *Managed (No VLAN)* modes, Eucalyptus needs two sets of IP addresses.
- For *Edge* mode, Eucalyptus needs at least one existing network.
- For *VPC and MidoNet*, Eucalyptus needs MidoNet to be installed. For more information, see *Install MidoNet*.
- The network connecting machines that host 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 connects the CC to the NCs. For information about testing connectivity, see *Verify Connectivity*.

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

Eucalyptus Installation

This section details steps to install Eucalyptus.

To install Eucalyptus, perform the following tasks in the order presented in this section.

Plan 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 *Eucalyptus Upgrade*.

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.

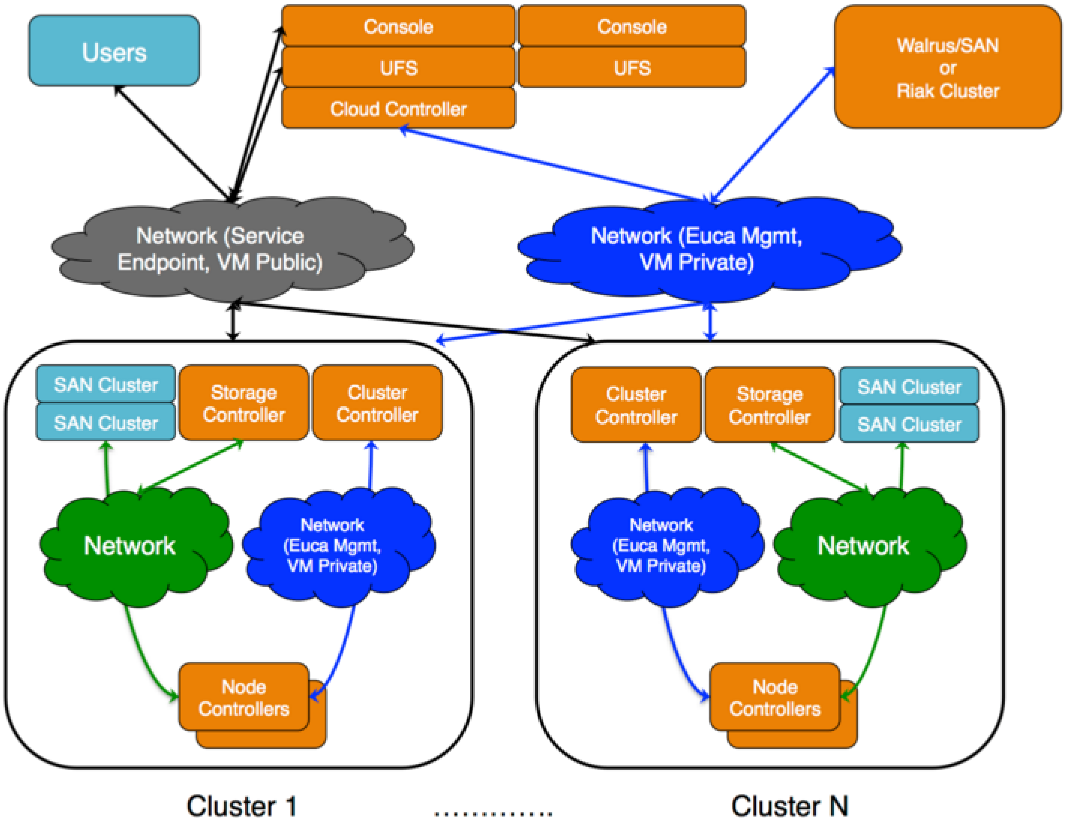
This section describes how to evaluate each tradeoff to determine the best choice to make, and how to verify that the resource environment can support the features that are enabled as a consequence of making a choice.

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.

Tip: For more help in planning your installation, see the *Eucalyptus Reference Architecture*, which includes use cases and reference architectures for various deployments.

Eucalyptus Architecture Overview

This topics describes the relationship of the components in a Eucalyptus installation.



The cloud components: Cloud Controller (CLC) and Walrus, as well as user components: User-Facing Services (UFS) and the Management Console, 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 subnets (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. Each cluster can use an internal private network for its NCs, and the CCs can route traffic from that private 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.

Riak CS clusters provide an alternative to Walrus as an object storage provider. SAN clusters are available to Eucalyptus subscribers.

Plan Your Hardware

This topic describes ways you can install Eucalyptus services on your physical servers.

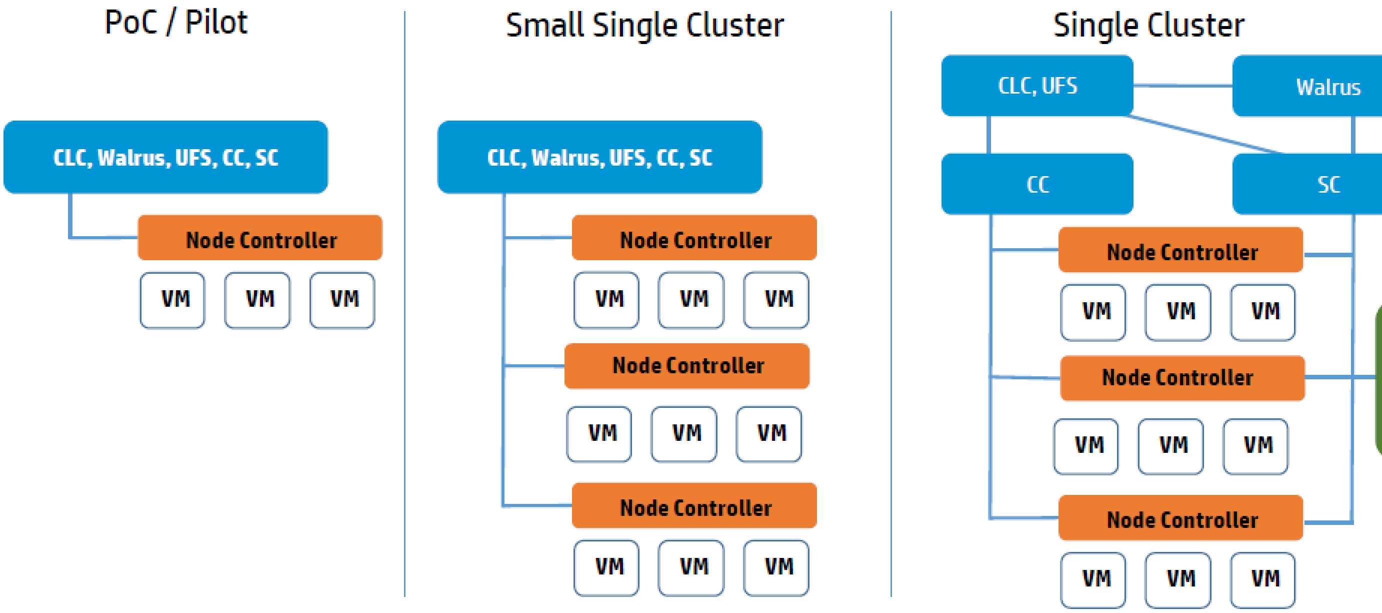
You can run Eucalyptus services in any combination on the various physical servers in a data center. For example, you can install the Cloud Controller, Walrus, CC, and SC on one host machine, and an NC on one or more host machines. Or you can install each service on an independent physical server. This gives each service its own local resources to work with.

Often in installation decisions, you must trade deployment simplicity for performance. For example, if you place all cloud and cluster services on a single machine, it makes for simple administration. This is because there is only one machine to monitor and control for the Eucalyptus control services. But, each service acts as an independent web service.

So if they share a single machine, the reduced physical resources available to each service might become a performance bottleneck.

Plan Services Placement

A Eucalyptus deployment includes user services (UFS and Management Console), as well as cloud services (Cloud Controller and Walrus) and one or more clusters, each of which contains a Cluster Controller, a Storage Controller, and one or more Node Controllers.



Cloud Services

The main decision for cloud services is whether to install the Cloud Controller (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, they use SOAP and REST to work together.

Sometimes the key factor for cloud services 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 services on the same server.

All services should be in the same data center. They use aggressive time-outs to maintain system responsiveness so separating them over a long-latency, lossy network link will not work.

User Services

The User Facing Services (UFS) handle most of the AWS APIs and provide an entry point for clients and users interacting with the Eucalyptus cloud. The UFS and the Management Console are often hosted on the same machine since both must be accessible from the public, client-facing network.

You may optionally choose to have redundant UFS and Management Console host machines behind a load balancer.

Cluster Services

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

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

Things to consider for CC placement:

- For Edge mode, the CC physical server will not act as a software gateway. Network traffic will be limited to small control messages.
- For Managed and Managed (No VLAN) modes, the CC physical server 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.
- In all cases, place the CC on a server that has TCP/IP connectivity to the Eucalyptus front-end servers and the NC servers in its zone.

Things to consider for SC placement:

- The SC host machine must always have TCP/IP connectivity to the CLC and be able use multicast to the CLC.
- The SC must have TCP/IP connectivity to the UFS/OSG hosts for uploading snapshots into the object store. (The SC does not require connectivity directly to users, it is an internal component and does not serve user EBS API requests; that job is done by the UFS.)
- The SC must be reachable via TCP/IP from all NCs in the cluster within which the SC is registered. The SC and NC exchange tokens to authorize volume attachment, so they must be able to directly communicate. The SC provides the NCs with network access to the dynamic block volumes on the SC's storage (if the SC is configured for overlay local filesystem or DAS-JBOD).
- If you are a subscriber and use one of the Eucalyptus-provided SAN integration drivers, the SC must also have TCP/IP connectivity to the SAN device. The SC sends control messages to the SAN and acts as a proxy to upload snapshots from the SAN to the UFS/OSG.
- If you are going to use overlay local filesystem or DAS-JBOD configurations to export local SC storage for EBS, then 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.

Node Services

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

Plan Disk Space

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

We recommend that you choose a disk for the 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 heavy S3 usage, Riak CS is a better choice for object storage.

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

Service	Directory	Minimum Size
Cloud Controller (CLC)	`/var/lib/eucalyptus/db`	20GB
CLC logging	`/var/log/eucalyptus`	2GB
Walrus	`/var/lib/eucalyptus/bukkits`	250GB
Walrus logging	`/var/log/eucalyptus`	2GB

Service	Directory	Minimum Size
Storage Controller (SC) (EBS storage) ★ **Important:** This disk space on the SC is only required if you are not using a SAN driver or if you are using Direct Attached Storage (DAS). For more information, see *Configure the Storage Controller*.	/var/lib/eucalyptus/volumes /var/log/eucalyptus	250GB
User-Facing Services (UFS) UFS logging	/var/lib/eucalyptus /var/log/eucalyptus	5GB 2GB
Management Console Console logging	/var/log/eucalyptus-console	5GB 2GB
Cluster Controller (CC) CC logging	/var/lib/eucalyptus/CC /var/log/eucalyptus	5GB 2GB
Node Controller (NC) NC logging	/var/lib/eucalyptus/instances /var/log/eucalyptus	250GB 2GB

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

Plan 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

This topic details what Eucalyptus needs in order 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 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.

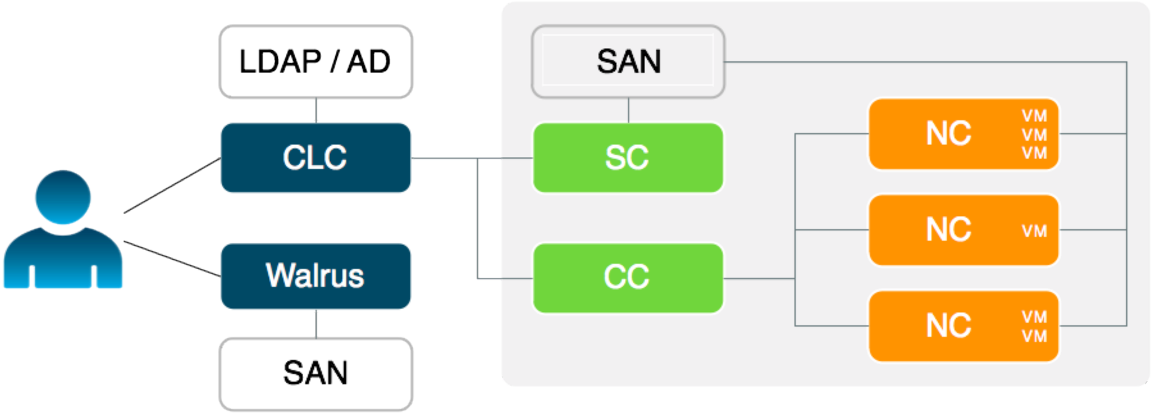
For additional Windows-related licensing information, see the following links:

- *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*

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

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.

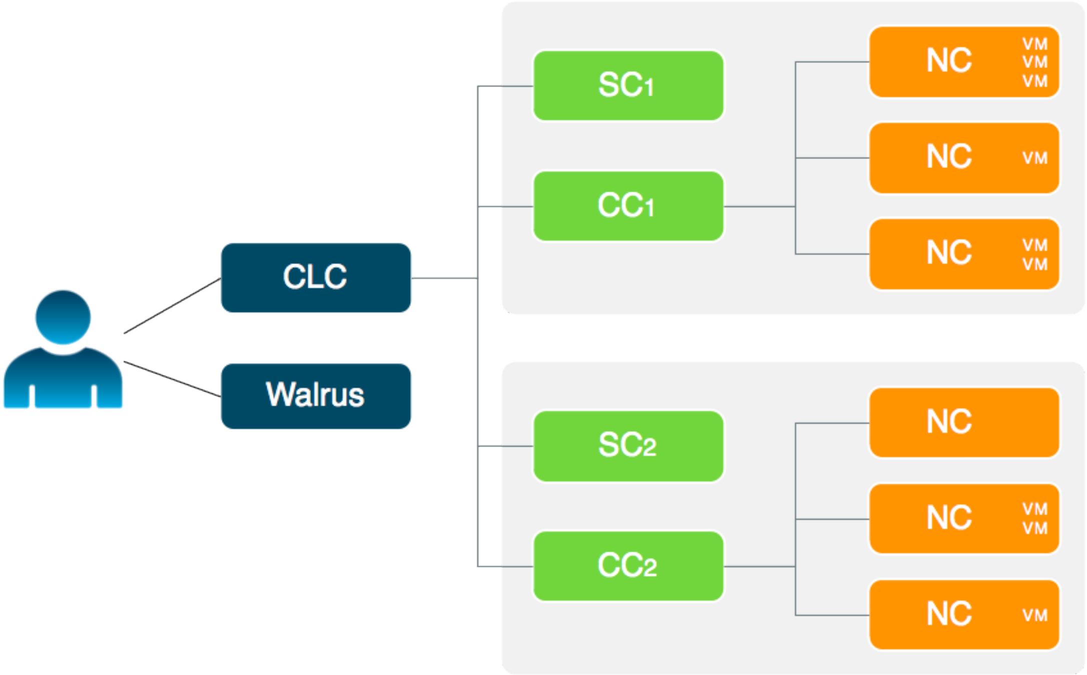
Eucalyptus supports these SAN devices:

- HP 3PAR SAN
- NetApp SAN
- Dell EqualLogic SAN

See the Compatibility Matrix in the *Release Notes* for supported versions.

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.



Object Storage

Eucalyptus supports Walrus and Riak CS as its object storage backend. There is no extra planning if you use Walrus. If you use Riak CS, you can use a single Riak CS cluster for several Eucalyptus clouds. Basho (the vendor of RiakCS) recommends five nodes for each Riak CS cluster. This also means that you have to set up and configure a load balancer between the Riak CS nodes and the object storage gateway (OSG).

Plan Networking Modes

Eucalyptus overlays a virtual network on top of your existing network. In order to do this, Eucalyptus supports four different networking modes: Edge, Managed, Managed (No VLAN), and VPC (MidoNet).

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?

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 Eucalyptus cloud. How these addresses are allocated and assigned to instances is determined by a networking mode. The distinction between public and private addresses becomes important in Edge, 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.	Edge Managed Managed (No VLAN) VPC (MidoNet)
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.	Edge Managed Managed (No VLAN) VPC (MidoNet)
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 ebtables (Edge) or VLAN tags (Managed), thus, protecting VMs from possible eavesdropping by VM instances belonging to other security groups.	Edge Managed VPC (MidoNet)
DHCP server	Eucalyptus assigns IP addresses to VMs in all modes.	Edge Managed Managed (No VLAN) VPC (MidoNet)

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.

Each networking mode is detailed in the following sections.

Edge Mode

Edge mode offers the most features of the EC2 Classic-compatible networking modes. It is designed to integrate into already extant (or straightforward to deploy) underlying network topologies. However, Edge mode can impose constraints in certain environments, in which case you can choose another mode.

In Edge networking mode, the component responsible for implementing Eucalyptus VM networking artifacts is running at the edge of a Eucalyptus deployment: the Node Controller (NC). Eucalyptus provides a stand-alone component called `eucanetd` in each NC. This component dynamically receives changing Eucalyptus networking views and is responsible for configuring the Linux machine on which the NC is running to reflect the latest view.

Edge networking mode integrates with your existing network infrastructure, allowing you to inform Eucalyptus, through configuration parameters for Edge mode, about the existing network, which Eucalyptus then will consume when implementing the networking view.

Edge networking mode integrates with two basic types of pre-existing network setups:

- One flat IP network used to service Eucalyptus component systems, Eucalyptus VM public IPs (elastic IPs), and Eucalyptus VM private IPs.
- Two networks, one for Eucalyptus components and Eucalyptus VM public IPs, and the other for Eucalyptus VM private IPs.

Important: Edge networking mode will not set up the network from scratch as do Managed and Managed (No VLAN) modes. Instead, it integrates with networks that already exist. If the network, netmask, and router don't already exist, you must create them outside of Eucalyptus before configuring Edge mode.

Edge Mode Requirements

- Each NC must have an interface configured with an IP on a VM public and a VM private network (which can be the same network).
  - There must be unused IP addresses on the VM public network for Eucalyptus to assign VM elastic IPs.
  - There must be unused IP addresses on the VM private network for Eucalyptus to assign VM private IPs.
- There must be IP connectivity from each NC machine (where `eucanetd` runs) and the CLC, for metadata re-directs for 169.254.169.254 to the active CLC to function.
- There must be a functioning router in place for the private network. This router will be the default gateway for VM instances.
- The private and public networks can be the same network, but they can also be separate networks.
- The Node Controllers (NCs) need a bridge configured on the private network, with the bridge interface itself having been assigned an IP from the network.
- If you're using a public network, the NCs need an interface on the public network as well (if the public and private networks are the same network, then the bridge needs an IP assigned on the network).
- If you run multiple clusters, each cluster can use the same network as its private network, or they can use separate networks as private networks. If you use separate networks, you need to have a router in place that is configured to route traffic between the networks.
- If you use private addressing only mode, the Cloud Controller machines must have a route back to the VM private network.

Edge Mode Limitations

- All NCs must have interfaces on the VM public (Elastic IP) network.
- Global network updates (such as security group rule updates, security group VM membership updates, and elastic IP updates) are applied through an "eventually consistent" mechanism, as opposed to an "atomic" mechanism. That is, there may be a brief period of time where one NC has the new state implemented but another NC has the previous state implemented.
- Mappings between VM MAC addresses and private IPs are strictly enforced. This means that instances cannot communicate using addresses the cloud has not assigned to them.

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.

In Managed mode, you define a subnet (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 time or dynamically at run-time. This capability is similar to Amazon's 'elastic IPs'.

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 *Prepare VLAN*.

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.
- There must be a separate Layer 2 network for each cluster in a multiple cluster environment.

Managed Mode Limitations

- The number of instances permitted in each security group is limited by the size of the private subnet and the number of security groups you choose to allow.
- The machine that hosts the CC will be a router in the data path for any VM traffic that is not 'VM private IP to VM private IP, where both VMs are in the same security group'.
- Network switch must be properly configured. For more information, see *Prepare VLAN*.
- The machine that hosts the CC is a single point of failure for most VM network communication.
- Instances may belong only to one security group.

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.
- If you plan to set up more than one cluster, you need to have a bridge for security groups to span the clusters.

Managed (No VLAN) Mode Limitations

- Limited (Layer-3) VM isolation.
- The number of instances permitted in each security group is limited by the size of the private subnet and the number of security groups you choose to allow.
- Instances may belong only to one security group.

Understanding VPC and MidoNet

This topic describes MidoNet components and their Eucalyptus deployment options, which provide support for VPC on Eucalyptus.

Eucalyptus Virtual Private Cloud (VPC) support is implemented with a Software-Defined Networking (SDN) technology developed by Midokura, called MidoNet. MidoNet is an open-source network virtualization platform for Infrastructure-as-a-Service (IaaS) clouds that implements and exposes virtual network components as software abstractions, enabling programmatic provisioning of virtual networks. The purpose of this topic is to describe some possible network reference architectures for the deployment, configuration, and operation of MidoNet for Eucalyptus clouds.

MidoNet Components

A MidoNet deployment consists of four types of nodes (according to their logical functions or services offered), connected via four IP networks as depicted in Figure 1. MidoNet does not require any specific hardware, and can be deployed in commodity x86_64 servers. Interactions with MidoNet are accomplished through Application Programming Interface (API) calls, which are translated into (virtual) network topology changes. Network state information is stored in a logically centralized data store, called the Network State Database (NSDB), which is implemented on top of two open-source distributed coordination and data store technologies: Zookeeper and Cassandra. Implementation of (virtual) network topology is realized via cooperation and coordination among MidoNet agents, which are deployed in nodes that participate in MidoNet.

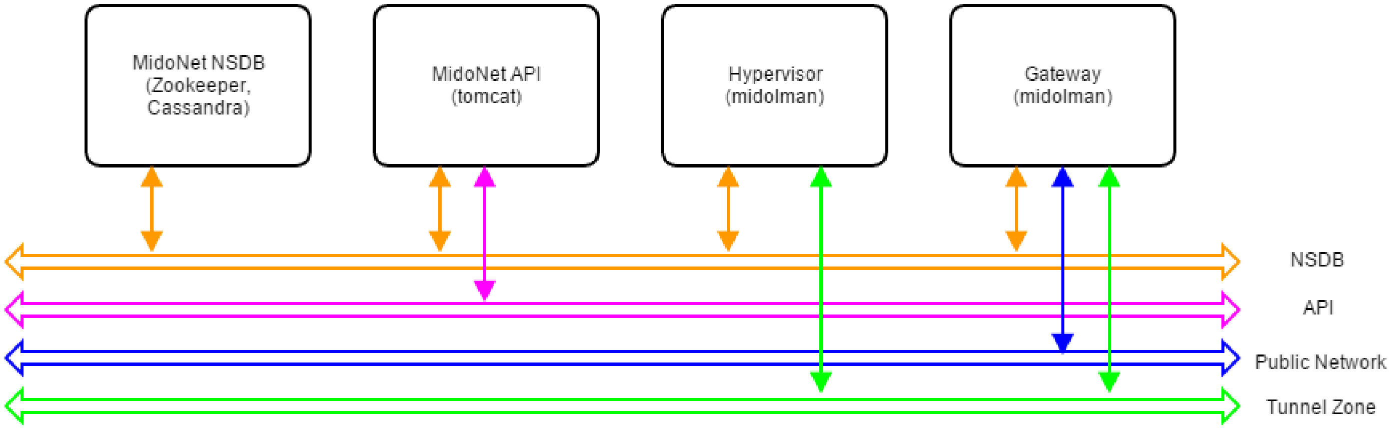


Figure 1: Logical view of a MidoNet deployment. Four components are connected via four networks.

Node types:

- MidoNet Network State Database (NSDB): consists of a cluster of Zookeeper and Cassandra. All MidoNet nodes must have IP connectivity with NSDB.
- MidoNet API: consists of tomcat and MidoNet web app. Exposes MidoNet REST APIs.
- Hypervisor: MidoNet agent (midolman) are required in all Hypervisors to enable VMs to be connected via MidoNet overlay networks/SDN.
- Gateway: Gateway nodes are connected to the public network, and enable the network flow from MidoNet overlays to the public network.

Physical Networks

- NSDB: IP network that connects all nodes that participate in MidoNet. While NSDB and Tunnel Zone networks can be the same, it is recommended to have an isolated (physical or VLAN) segment.
- API: in Eucalyptus deployments only eucanetd/CLC needs access to the API network. Only "special hosts/processes" should have access to this network. The use of "localhost" network on the node running CLC/eucanetd is sufficient and recommended in Eucalyptus deployments.
- Tunnel Zone: IP network that transports the MidoNet overlay traffic (Eucalyptus VM traffic), which is not "visible" on the physical network.
- Public network: network with access to the Internet (or corporate/enterprise) network.

MidoNet Deployment Scale

Three reference architectures are presented in this document, ordered by complexity and size:

- Proof-of-Concept (PoC)
- Production: Small
- Production: Large

Production: Large reference architecture represents the most complete and recommended deployment model of MidoNet for Eucalyptus. Whenever possible (such as when resources are available), deployments should closely match with the Production: Large reference architecture (even on small scale clouds).

All MidoNet components are designed and implemented to horizontally scale. Therefore, it is possible to start small and add resources as they become available.

MidoNet Software Version

There are currently two distributions of MidoNet:

- Midokura Enterprise MidoNet (commercial version with 24/7 support - 30 day evaluation available). Eucalyptus tested and validated using MEM v1.9 series.
- Open Source MidoNet (available at http://www.midonet.org)

MEM version 1.9 is currently the recommended/validated version for Eucalyptus deployments.

Eucalyptus with MidoNet

A Eucalyptus with MidoNet deployment consists of the following components:

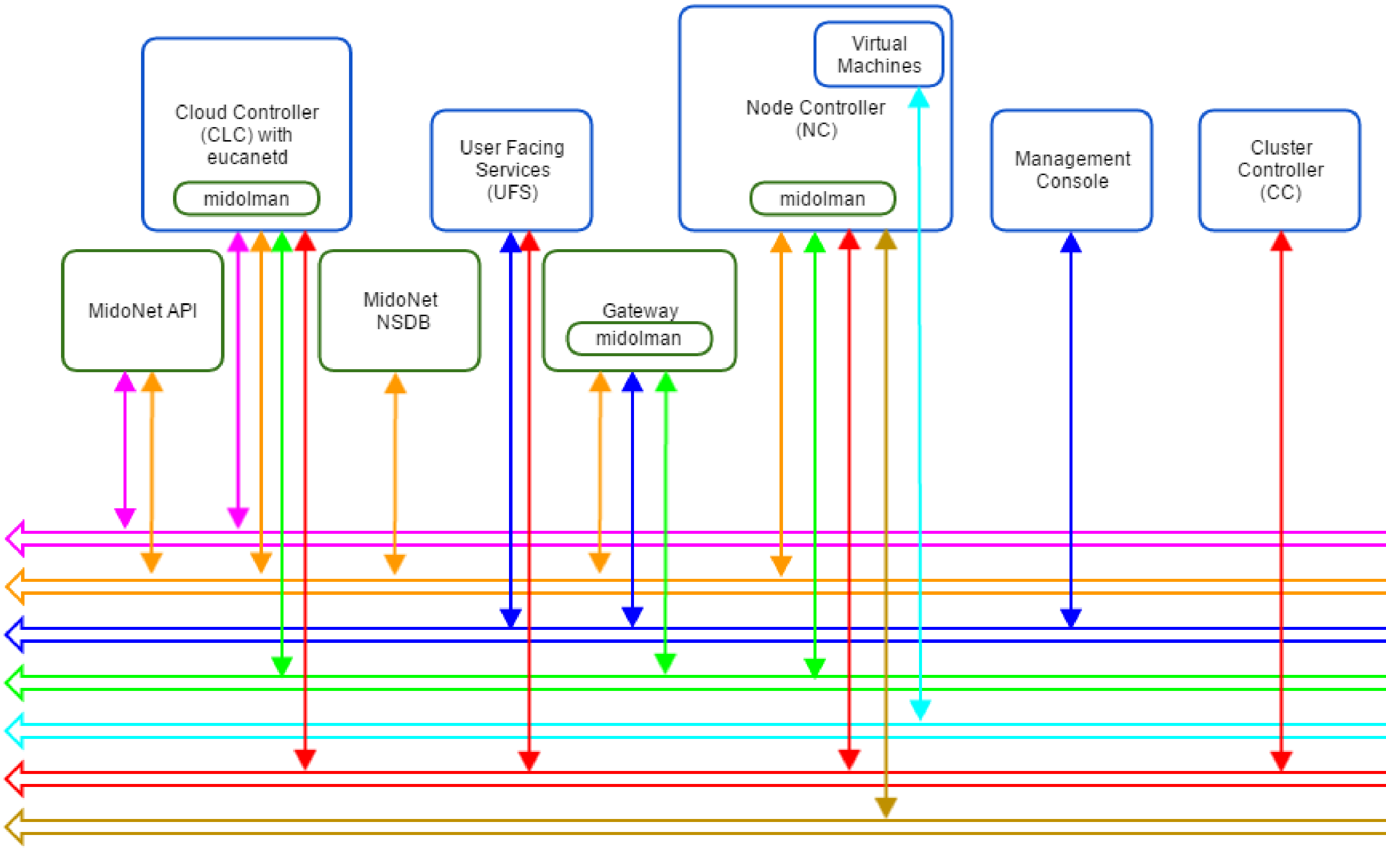


Figure 2: Logical view of a Eucalyptus with MidoNet deployment. VM private network is created/virtualized by MidoNet, and 'software-defined' by eucanetd. Ideally, each component and network should have its own set of independent resources. In practice, components are grouped and consolidated into a set of servers, as detailed in different reference architectures.

MidoNet components, Eucalyptus components, and three extra networks are present.

Proof of Concept (PoC)

The PoC reference architecture is designed for very small and transient workloads, typical in development and testing environments. Quick deployment with minimal external network requirements are the key points of PoC reference architecture.

Requirements

Servers:

- Four (4) or more modern Intel cores or AMD modules - exclude logical cores that share CPU resources from the count (Hyperthreads and AMD cores within a module)
- 2GB of RAM reserved for MidoNet Agent (when applicable)
- 4GB of RAM reserved for MidoNet NSDB (when applicable)
- 4GB of RAM reserved for MidoNet API (when applicable)
- 30GB of free disk space for NSDB (when applicable)

Physical Network:

- One (1) 1Gbps IP Network
- A range or list of public IP addresses (Euca_public_IPs)
- Internet Gateway

Limits:

- Ten (10) MidoNet agents (i.e., 1 Gateway node, 1 CLC, and 8 NCs)
- One (1) MidoNet Gateway
- No fail over, fault tolerance, and/or network load balancing/sharing

Deployment Topology

- Single server with all MidoNet components (NSDB, API, and midolman), and with CLC/eucanetd
- A server acting as MidoNet Gateway - when BGP terminated links are used, this node must not be co-located with CLC/eucanetd (in a proxy_arp setup described below, it is possible to consolidate CLC/eucanetd with MidoNet Gateway). This is due to incompatibilities in CentOS/RHEL6 netns (used by eucanetd), and bgpd (started by midolman when BGP links are configured).
- Hypervisors with midolman
- One IP network handling NSDB, Tunnel Zone, and Public Network traffic
- API communication via loopback/localhost network

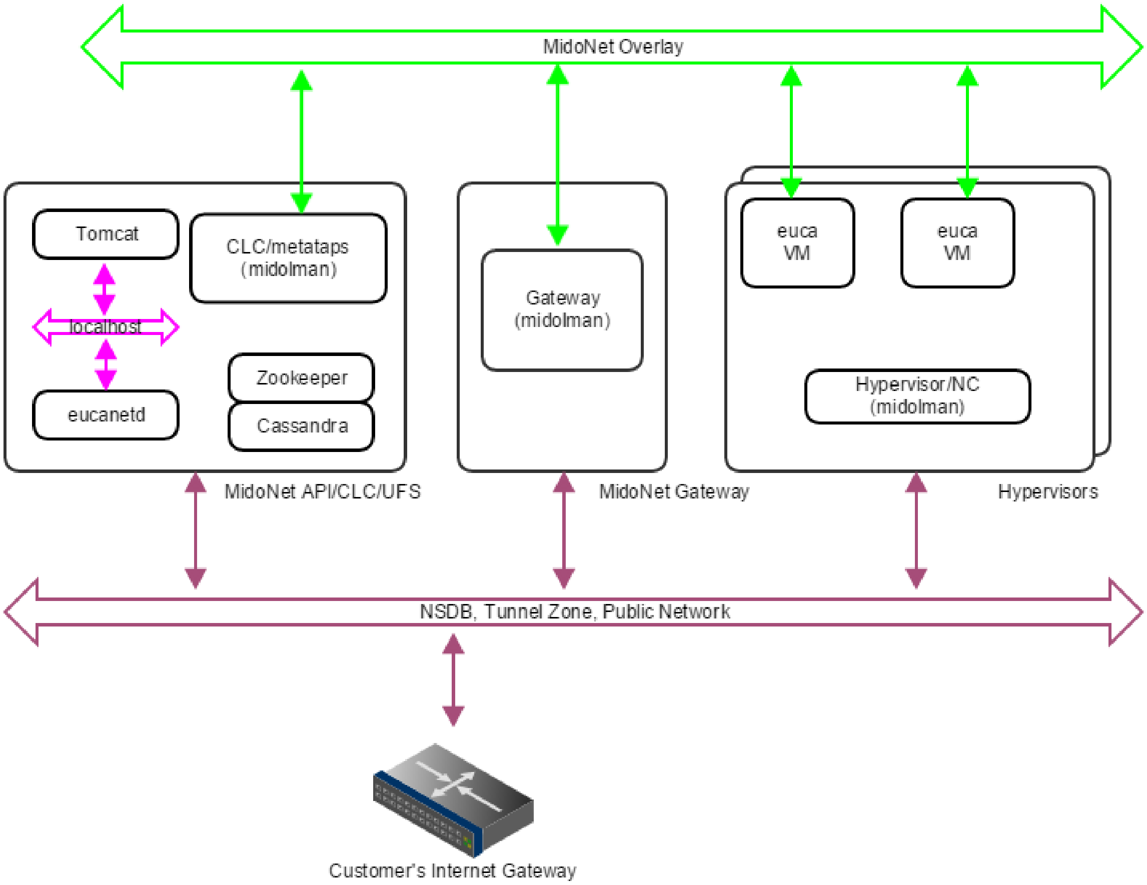


Figure 3: PoC deployment topology. A single IP network carries NSDB, Tunnel Zone, and Public Network traffic. A single server handles MidoNet NSDB, API (and possibly Gateway) functionality.

MidoNet Gateway Bindings

Three ways to realize MidoNet Gateway bindings are discussed below, starting with the most recommended setup.

Public CIDR block(s) allocated for Eucalyptus (Euca_Public_IPs) needs to be routed to MidoNet Gateway by the customer network - this is an environment requirement, outside of control of both MidoNet and Eucalyptus systems. One way to accomplish this is to have a BGP terminated link available. MidoNet Gateway will establish a BGP session with the customer router to: (1) advertise Euca_Public_IPs to the customer router; and (2) get the default route from the customer router.

If a BGP terminated link is not available, but the routing of Euca_Public_IPs is delegated to MidoNet Gateway (configuration of customer routing infrastructure), similar setup can be used. In such scenario, static routes are configured on the customer router (to route Euca_Public_IPs to MidoNet Gateway), and on MidoNet (to use the customer router as the default route).

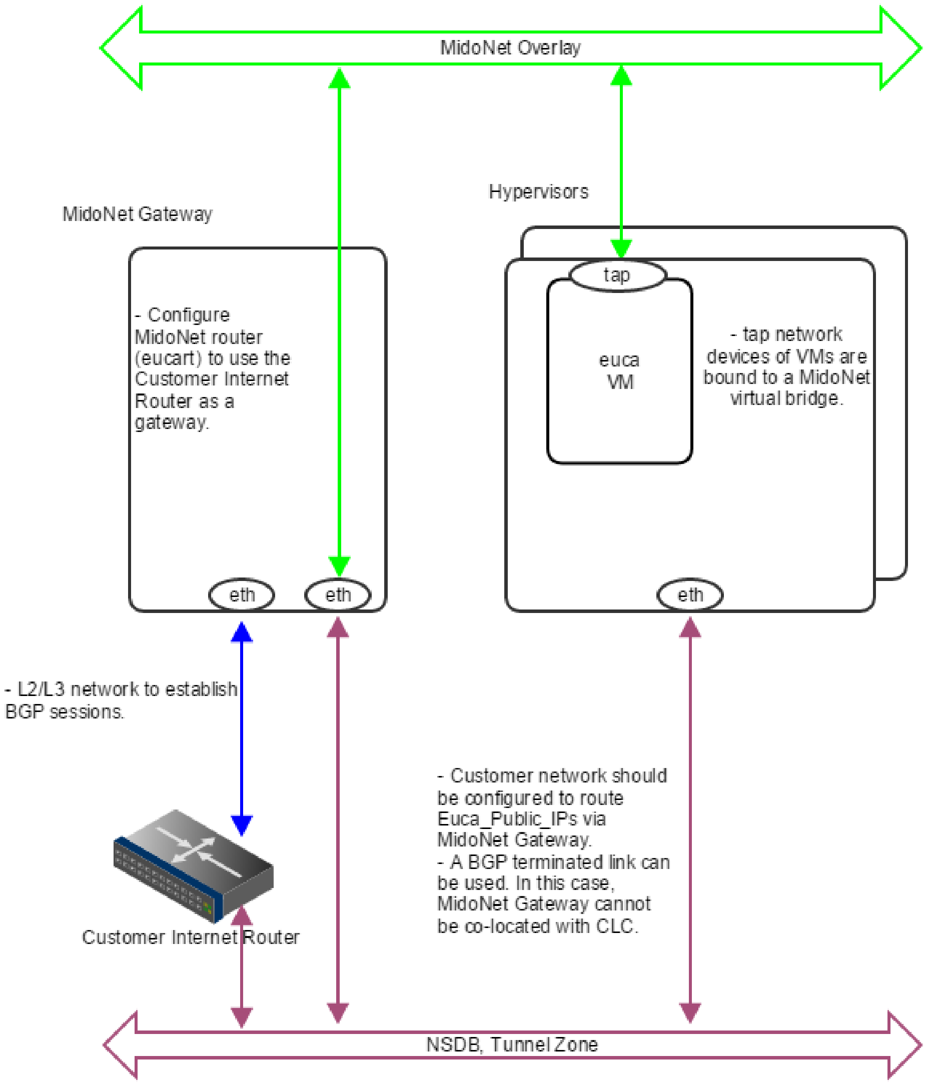

MidoNet Overlay
MidoNet Gateway
Hypervisors
- Configure MidoNet router (eucart) to use the Customer Internet Router as a gateway.
tap
euca VM
- tap network devices of VMs are bound to a MidoNet virtual bridge.
eth
eth
eth
- L2/L3 network to establish BGP sessions.
- Customer network should be configured to route Euca_Public_IPs via MidoNet Gateway.
- A BGP terminated link can be used. In this case, MidoNet Gateway cannot be co-located with CLC.
Customer Internet Router
NSDB, Tunnel Zone

Figure 4: How servers are bound to MidoNet in a PoC deployment with BGP. A BGP terminated link is required: the gateway node eth device is bound to MidoNet virtual router (when BGP is involved, the MidoNet Gateway and Eucalyptus CLC cannot be co-located). Virtual machine tap devices are bound to MidoNet virtual bridges.

If routed Euca_Public_IPs are not available, static routes on all involved nodes (L2 connectivity is required among nodes) can be used as illustrated below.

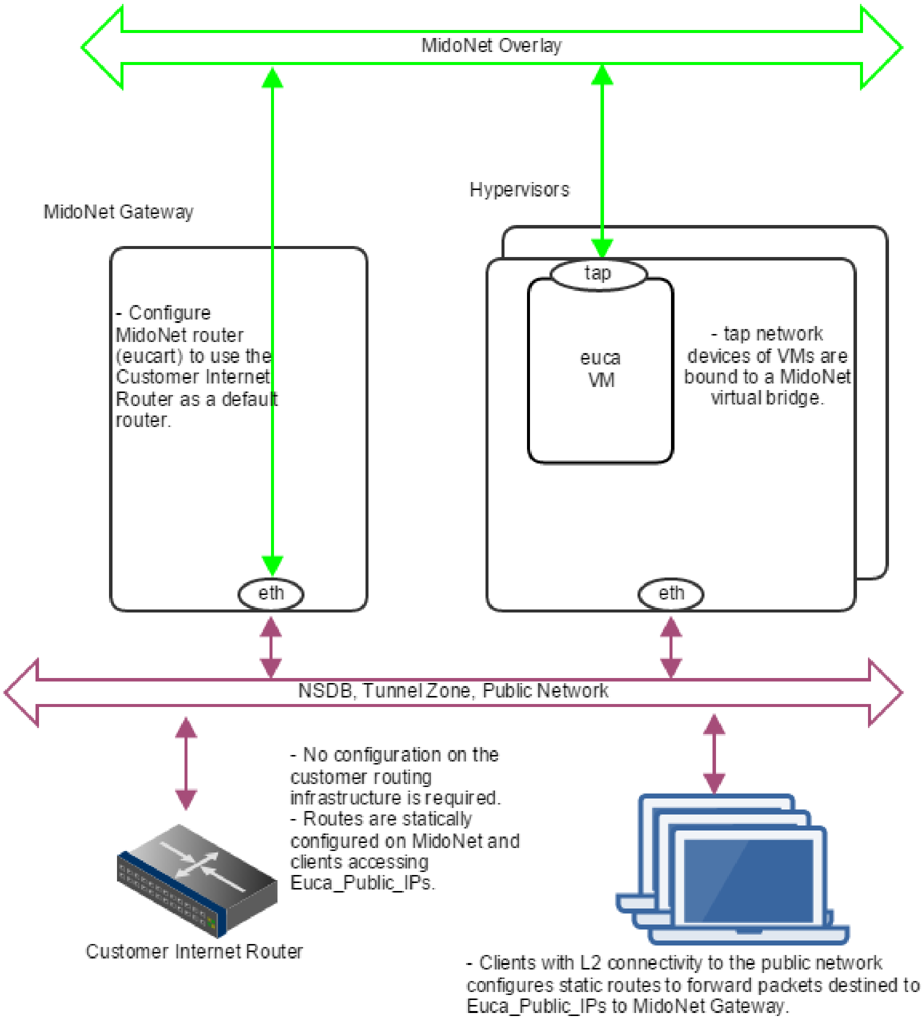

MidoNet Overlay
MidoNet Gateway
Hypervisors
- Configure MidoNet router (eucart) to use the Customer Internet Router as a default router.
tap
euca VM
- tap network devices of VMs are bound to a MidoNet virtual bridge.
eth
eth
NSDB, Tunnel Zone, Public Network
- No configuration on the customer routing infrastructure is required.
- Routes are statically configured on MidoNet and clients accessing Euca_Public_IPs.
Customer Internet Router
- Clients with L2 connectivity to the public network configures static routes to forward packets destined to Euca_Public_IPs to MidoNet Gateway.

Figure 5: How servers are bound to MidoNet in a PoC deployment without routed Euca_Public_IPs. Clients that need communication with Euca_Public_IPs configure static routes using MidoNet Gateway as the router. MidoNet Gateway configures a static default route to customer router.

In the case nodes outside the public network broadcast domain (L2) needs to access Euca_Public_IPs, a setup using proxy_arp, as illustrated below, can be used.

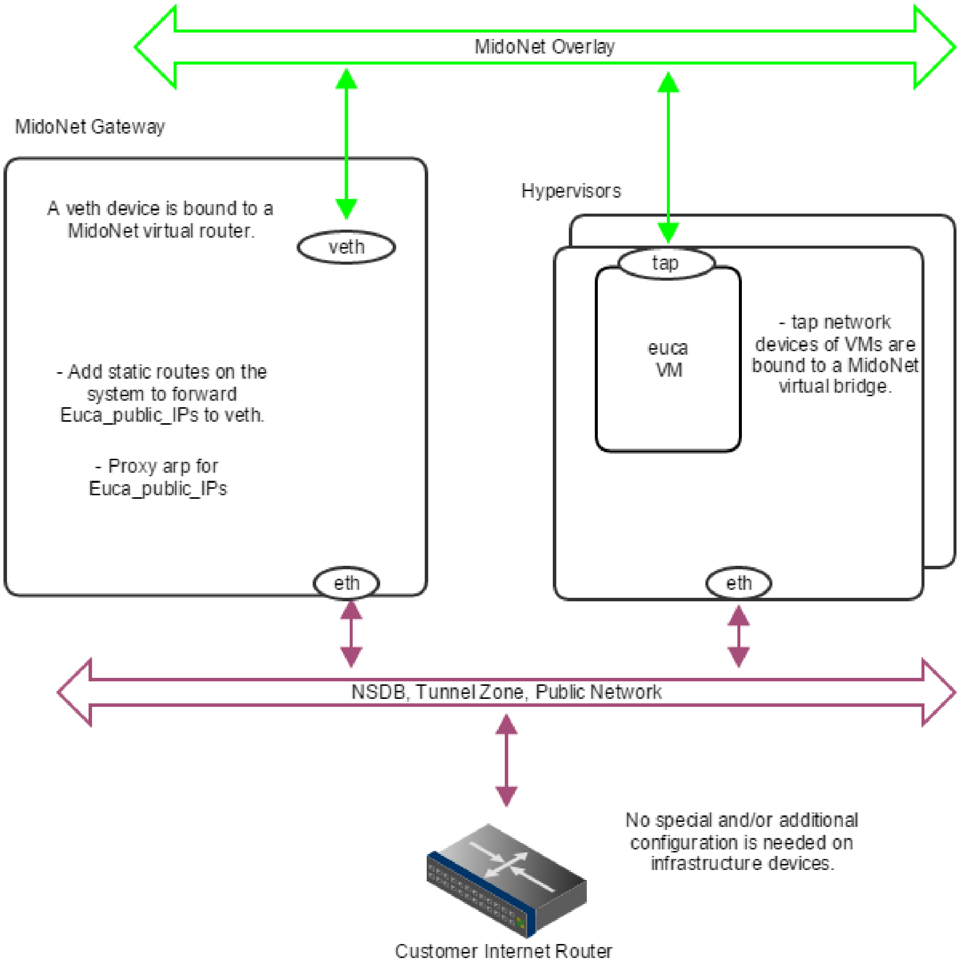

MidoNet Overlay
MidoNet Gateway
A veth device is bound to a MidoNet virtual router.
veth
- Add static routes on the system to forward Euca_public_IPs to veth.
- Proxy arp for Euca_public_IPs
eth
Hypervisors
tap
euca VM
- tap network devices of VMs are bound to a MidoNet virtual bridge.
eth
NSDB, Tunnel Zone, Public Network
No special and/or additional configuration is needed on infrastructure devices.
Customer Internet Router

Figure 6: How servers are bound to MidoNet in a PoC deployment with proxy_arp. When routed Euca_Public_IPs are not available, the gateway node should proxy arp for public IP addresses allocated for Eucalyptus, and forward to a veth device that is bound to a MidoNet virtual router. Virtual machine tap devices are bound to MidoNet virtual bridges.

Production: Small

The Production: Small reference architecture is designed for small scale production quality deployments. It supports MidoNet NSDB fault tolerance (partial failures), and limited MidoNet Gateways fail-over and load balancing/sharing.

Border Gateway Protocol (BGP) terminated uplinks are recommended for production quality deployments.

Requirements

Servers:

- Four (4) or more modern Intel cores or AMD modules - exclude logical cores that share CPU resources from the count (Hyperthreads and AMD cores within a module) - for gateway nodes, 4 or more cores should be dedicated to MidoNet agent (midolman)
- 4GB of RAM reserved for MidoNet Agent (when applicable), 8GB for Gateway nodes
- 4GB of free RAM reserved for MidoNet NSDB (when applicable)
- 4GB of free RAM reserved for MidoNet API (when applicable)
- 30GB of free disk space for NSDB (when applicable)
- Two (2) 10Gbps NICs per server
- Three (3) servers dedicated to MidoNet NSDB
- Two (2) servers as MidoNet Gateways

Physical Network:

- One (1) 10Gbps IP Network for public network (if upstream links are 1Gbps, this could be 1Gbps)
- One (1) 10Gbps IP Network for Tunnel Zone and NSDB
- Public Classless Inter-Domain Routing (CIDR) block (Euca_public_IPs)
- Two (2) BGP terminated uplinks

Limits:

- Thirty two (32) MidoNet agents (i.e., 2 Gateway nodes and 30 Hypervisors)
- Two (2) MidoNet Gateways
- Tolerate 1 NSDB server failure
- Tolerate 1 MidoNet Gateway/uplink failure
- Limited uplinks load sharing/balancing

Deployment Topology

- A 3-node cluster for NSDB (co-located Zookeeper and Cassandra)
- eucanetd co-located with MidoNet API Server (Tomcat)
- Two (2) MidoNet Gateway Nodes
- Hypervisors with midolman
- One 10Gbps IP network handling NSDB and Tunnel Zone traffic
- One 10Gbps IP Network handling Public Network traffic
- API communication via loopback/localhost network

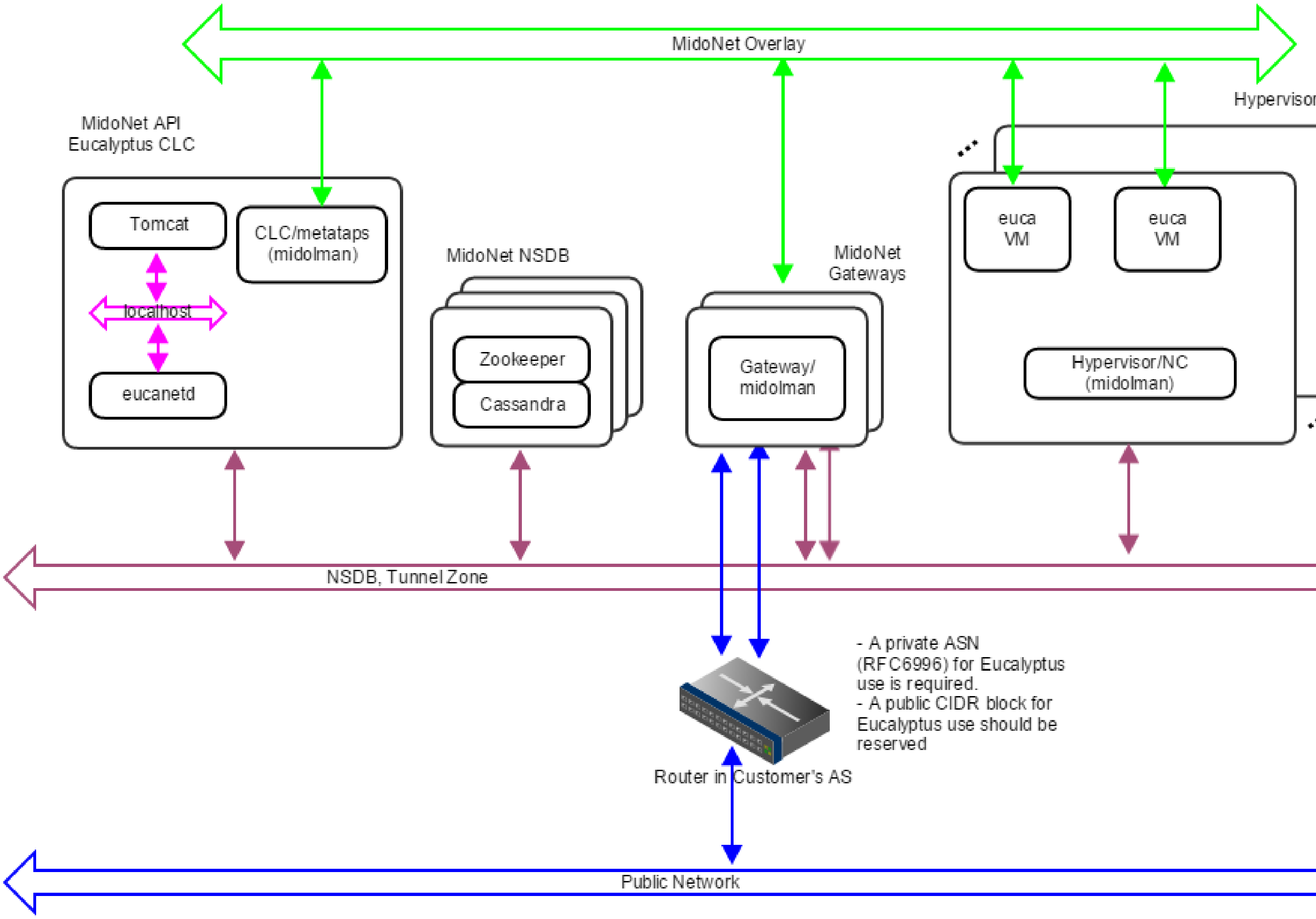
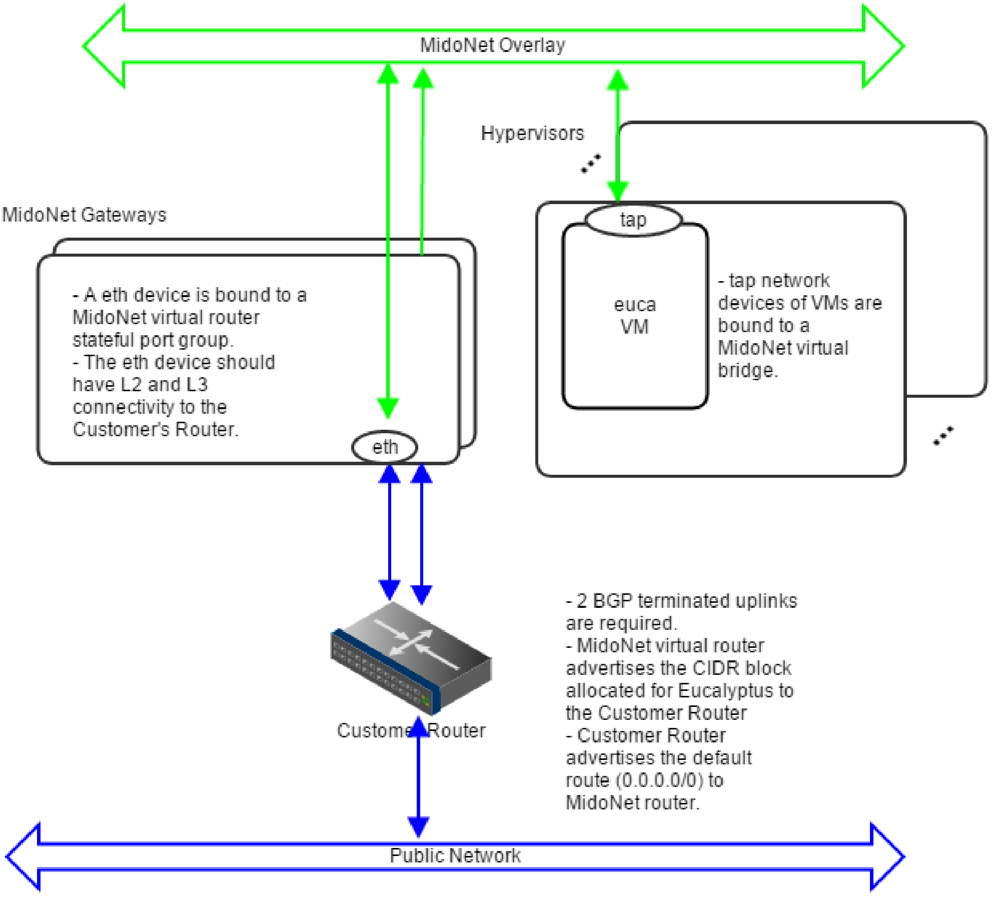


Figure 7: Production:Small deployment topology. A 10Gbps IP network carries NSDB and Tunnel Zone traffic. Another 10Gbps IP network carries Public Network traffic. A 3-node cluster for NSDB tolerates 1 server failure, and 2 gateways enable network fail-over and limited load balancing/sharing.



- A eth device is bound to a MidoNet virtual router stateful port group.
- The eth device should have L2 and L3 connectivity to the Customer's Router.

- tap network devices of VMs are bound to a MidoNet virtual bridge.

- 2 BGP terminated uplinks are required.
- MidoNet virtual router advertises the CIDR block allocated for Eucalyptus to the Customer Router
- Customer Router advertises the default route (0.0.0.0/0) to MidoNet router.

Figure 8: How servers are bound to MidoNet in a Production:Small deployment. Gateway Nodes have physical devices bound to a MidoNet virtual router. These devices should have L2 and L3 connectivity to the Customer's Router, and with BGP terminated links. Virtual machine tap devices are bound to MidoNet virtual bridges.

NSDB Data Replication

- NSDB is deployed in a cluster of 3 nodes
- Zookeeper and Cassandra both have built-in data replication
- One server failure is tolerated

MidoNet Gateway Failover

- Two paths are available to and from MidoNet, and failover is handled by BGP

MidoNet Gateway Load Balancing and Sharing

- Load Balancing from MidoNet is implemented by MidoNet agents (midolman): ports in a stateful port group with default routes out are used in a round-robin fashion.
- Partial load sharing from the Customer's router to MidoNet can be accomplished by:
  - Partition the allocated CIDR in 2 parts. For example, a /24 CIDR can be split into 2 /25 CIDRs.
  - One MidoNet BGP port should advertise the top half (/25) and /24; the other advertises the bottom half (/25) and /24.
  - When both ports are operational, routing will favor the most specific route (i.e., /25). If a port fails, the /24 will be used instead.

Production: Large

The Production:Large reference architecture is designed for large scale (500 to 600 MidoNet agents) production quality deployments. It supports MidoNet NSDB fault tolerance (partial failures), and MidoNet Gateways fail-over and load balancing/sharing.

Border Gateway Protocol (BGP) terminated uplinks are required. Each uplink should come from an independent router.

Requirements:

- Eight (8) or more modern Intel cores or AMD modules - exclude logical cores that share CPU resources from the count (Hyperthreads and AMD cores within a module) - for gateway nodes, 8 or more cores should be dedicated to MidoNet agent (midolman)
- 4GB of RAM reserved for MidoNet Agent (when applicable), 16GB for Gateway nodes
- 4GB of free RAM reserved for MidoNet NSDB (when applicable)
- 4GB of free RAM reserved for MidoNet API (when applicable)
- 30GB of free disk space for NSDB (when applicable)
- One 1Gbps and 2 10Gbps NICs per server
- Five (5) servers dedicated to MidoNet NSDB
- Three (3) servers as MidoNet Gateways

Physical Network:

- One 1Gbps IP Network for NSDB
- One 10Gbps IP Network for public network (if upstream links are 1Gbps, this could be 1Gbps)
- One 10Gbps IP Network for Tunnel Zone
- Public Classless Inter-Domain Routing (CIDR) block (Euca_public_IPs)
- Three (3) BGP terminated uplinks, each of which coming from an independent router

Limits:

- 500 to 600 MidoNet agents
- Three (3) MidoNet Gateways
- Tolerate 1 to 2 NSDB server failures
- Tolerate 1 to 2 MidoNet Gateway/uplink failures

Deployment Topology

- A 5-node cluster for NSDB (co-located Zookeeper and Cassandra)
- eucanetd co-located with MidoNet API Server (Tomcat)
- Three (3) MidoNet Gateway Nodes
- Hypervisors with midolman
- One 1Gbps IP network handling NSDB traffic
- One 10Gbps IP network handling Tunnel Zone traffic
- One 10Gbps IP network handling Public Network traffic
- API communication via loopback/localhost network

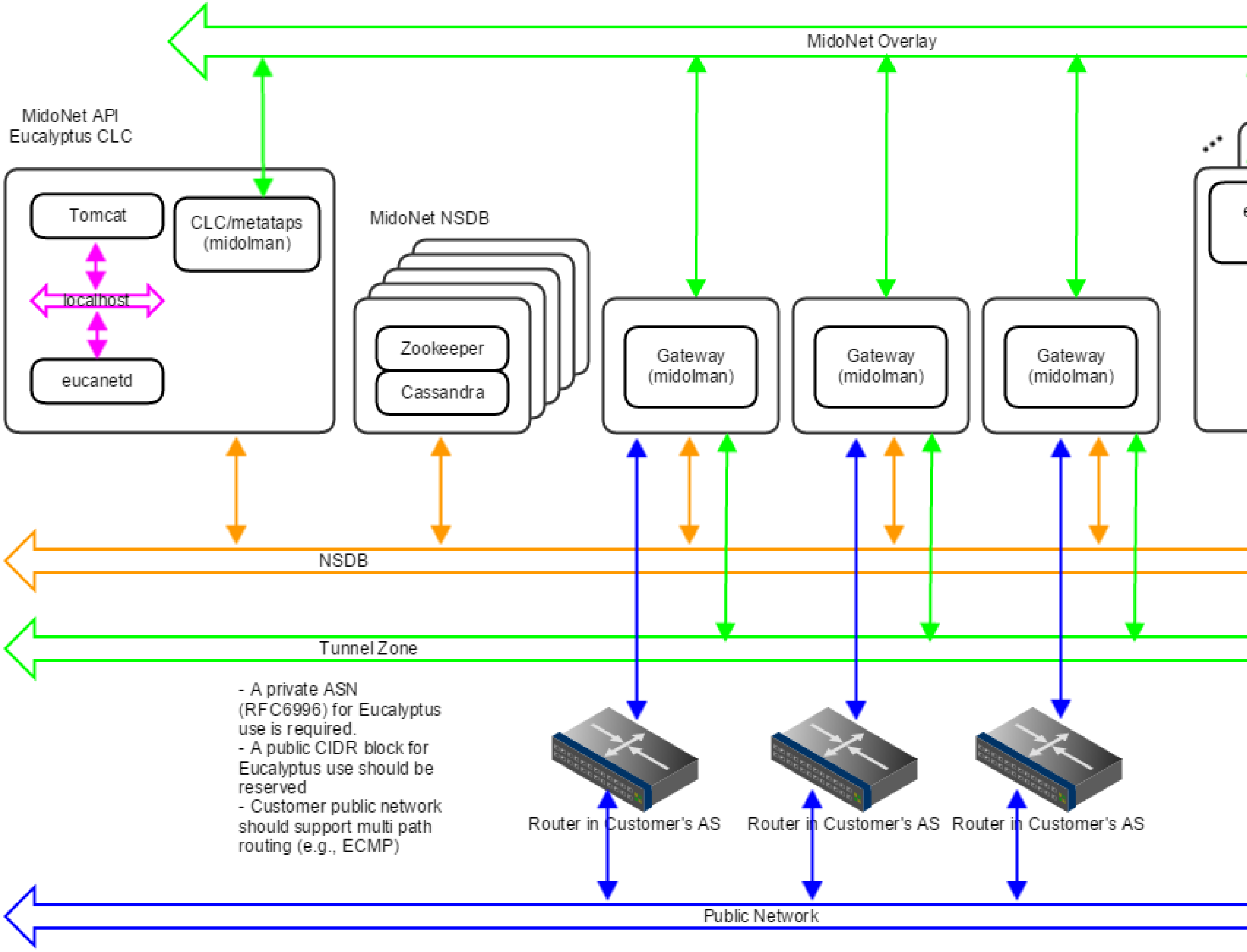


Figure 9: Production:Large deployment topology. A 1Gbps IP network carries NSDB; a 10Gbps IP network carries Tunnel Zone traffic; and another 10Gbps IP network carries Public Network traffic. A 5-node cluster for NSDB tolerates 2 server failures, and 3 gateways enable network fail-over and load balancing/sharing. Servers are bound to MidoNet in a way similar to Production:Small.

NSDB Data Replication

- NSDB is deployed in a cluster of 5 nodes
- Zookeeper and Cassandra both have built-in data replication
- Up to 2 server failures tolerated

MidoNet Gateway Failover

- Three paths are available to and from MidoNet, and failover is handled by BGP

MidoNet Gateway Load Balancing/Sharing

- Load Balancing from MidoNet is implemented by MidoNet agents (midolman): ports in a stateful port group with default routes out are used in a round-robin fashion.
- The customer AS should handle multi path routing in order to support load sharing/balancing to MidoNet; for example, Equal Cost Multi Path (ECMP).

Prepare the Network

In order for Eucalyptus to function in your local environment, be sure to prepare your network. To prepare your network, perform the tasks listed in this section.

Reserve Ports

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 8443	Port for getting user credentials on the CLC. Configurable with `euctl`.
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, user-facing services (UFS), object storage gateway (OSG), Walrus SC; also used for external and internal communications by the CLC and Walrus. Configurable with `euctl`.
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 8777	Database port on the CLC
TCP 8779 (or next available port, up to TCP 8849)	jGroups failure detection port on CLC, UFS, OSG, Walrus 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 8888	The default port for the Eucalyptus Management Console. Configured in the `/etc/eucalyptus-console/console.ini` file.
TCP 16514	TLS port on Node Controller, required for node migrations
UDP 7500	Port for diagnostic probing on CLC, UFS, OSG, Walrus SC
UDP 8773	Membership port for SC, any UFS, Walrus
UDP 8778	The bind port used to establish multicast communication
TCP/UDP 53	DNS port on UFS

Verify 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 to SC on TCP port 8773
4. Verify connection from the CLC, SC, and NC to Walrus on TCP port 8773
5. Verify connection from Walrus and SC to CLC on TCP port 8777
6. Verify connection from CLC to CC on TCP port 8774

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

Prepare VLAN

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.

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

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
   ```

Configure Dependencies

Before you install Eucalyptus, ensure you have the appropriate dependencies installed and configured.

Configure Bridges

For Managed (No VLAN) and EDGE modes, you must configure a Linux ethernet bridge on all NCs. 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
   ```

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.

Tip: 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: Cloud Controller, Walrus, Cluster Controller, Storage Controller, and Node Controllers.

Configure SELinux

Security-enabled Linux (SELinux) is a security feature for Linux that allows you to set access control through policies. Eucalyptus is not currently 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, if necessary, 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. Synchronize your server.

```
ntpdate -u YOUR_NTP_SERVER
```

5. Configure NTP to run at reboot.

```
chkconfig ntpd on
```

6. Start NTP.

```
service ntpd start
```

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 machine that will run a Eucalyptus service.

Configure an MTA

All machines running the Cloud Controller 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`.

Enable Packet Routing

Edit the `sysctl.conf` on each Cluster Controller (CC) and Node Controller (NC) host machine.

In the `sysctl.conf` file, set the following parameters and values:

1. Enable IP forwarding.

```
net.ipv4.ip_forward = 1
```

2. Enable the bridge to forward traffic based on iptables rules.

```
net.bridge.bridge-nf-call-iptables = 1
```

Install MidoNet

Eucalyptus requires MidoNet to enable VPC functionality. This section describes how to install MidoNet for use with Eucalyptus.

Before you begin:

- See the *Planning your Network* section of the guide to create a map of how MidoNet / Eucalyptus will be deployed into your environment.
- See the *MidoNet Installation Guide* to become familiar with the general MidoNet installation procedure and concepts.

Note: If you are not using VPC with Eucalyptus, you do not need to install MidoNet.

Prerequisites

This topic discusses the prerequisites for installing MidoNet.

Repository Access

In order to use MidoNet with Eucalyptus you will need access to the Midokura repositories. You can sign up here: *https://support.midokura.com/access/unauthenticated*.

Create `/etc/yum.repos.d/midokura.repo` on all machines that will run MidoNet components including Zookeeper. For example:

```
[midokura]
name=Midokura Repository
baseurl=http://USERNAME:PASSWORD@yum.midokura.com/repo/v1.9/stable/RHEL/6/
gpgkey=http://USERNAME:PASSWORD@yum.midokura.com/repo/RPM-GPG-KEY-midokura
gpgcheck=1
enabled=1

[midokura-misc]
name=midokura Misc Package Repo
baseurl=http://repo.midonet.org/misc/RHEL/6/misc/
gpgkey=http://repo.midonet.org/RPM-GPG-KEY-midokura
enabled=1
gpgcheck=1
metadata_expire=1
```

Zookeeper

Zookeeper is where MidoNet stores most of its running state. This service needs to be up and running before any other installation takes place.

Note: For advanced zookeeper installation (clustered for reliability), please see the *MidoNet NSDB Installation Guide*.

For a simple single-server installation, install the zookeeper package on any server that is IP accessible from all midolman agents (for example: on the Cloud Controller server itself), start the service and ensure that the service is enabled. For example:

```
yum install zookeeper
service zookeeper start
chkconfig zookeeper on
```

Cassandra

Cassandra is used to track flows in MidoNet. This service needs to be up and running before any other installation takes place. For a simple single-server installation, install Cassandra on any server that is IP accessible from all midolman agents (for example: on the Cloud Controller server itself), start the service and ensure that the service is enabled.

To install Cassandra, please refer to the documentation for Cassandra *installation* and *configuration*.

Note: For advanced MidoNet-specific installation of Cassandra, please refer to the *MidoNet NSDB Installation Guide*.

Midokura Component Topology

The following section lists topology recommendations for installing Midokura.

Note: See *Understanding VPC and MidoNet* for more information on Midonet.

- The midonet-api must run co-located with the Eucalyptus Cloud Controller
- Each Node Controller must run a Midolman agent
- The Cloud Controller must run a Midolman agent
- It is recommended that your User Facing Services host be used as the Midonet Gateway (i.e. running a Midolman agent) when configuring Eucalyptus
- The Midonet Gateway will take over which ever interface Eucalyptus GatewayInterface is configured for and block traffic that is not to/from Midonet.

  - If you only have 1 interface on your host then you need to follow the instructions from Midokura on setting up a veth pair so that Midonet can take over a virtual interface rather than a physical one, as in this example (skip step 6 for Eucalyptus installs): *http://docs.midonet.org/docs/latest/operations-guide/content/static_setup.html*

Eucalyptus Network JSON Example

The following example shows a Eucalyptus network JSON file that is configured for Midokura:

```
{
  "InstanceDnsServers": [
    "UFS_HOST"
  ],
  "Mido": {
    "EucanetdHost": "clcfrontend",
    "GatewayHost": "ufsfrontend",
    "GatewayIP": "172.19.0.2",
    "GatewayInterface": "veth1",
    "PublicGatewayIP": "172.19.0.1",
    "PublicNetworkCidr": "172.19.0.0/30"
  },
  "Mode": "VPCMIDO",
  "PublicIps": [
    "PUBLIC_IPS"
  ]
}
```

Install Midokura on Eucalyptus

This topic shows how to install Midokura for use in your Eucalyptus cloud.

Install the Midonet API on the Cloud Controller

To install the Midonet API on the cloud controller:

1. Add the Midonet repo file as discussed in *Prerequisites*.
2. Install `tomcat`.

   ```
   yum install tomcat
   ```

3. Install `midonet-api`.

   ```
   yum install midonet-api
   ```

4. Install `python-midonetclient`.

   ```
   yum install python-midonetclient
   ```

5. Configure the `/usr/share/midonet-api/WEB-INF/web.xml` file. See the example file below.
   a) Set rest_api-base_uri to "127.0.0.1"
   b) Set auth-auth_provider to org.midonet.cluster.auth.MockAuthService
   c) Set zookeeper-zookeeper_hosts. In the example config below, replace `ZOOKEEPER_IP`.
   d) Set rest_api-base_uri to "127.0.0.1"
   e) An example configuration:

   ```
   <!DOCTYPE web-app SYSTEM "http://java.sun.com/dtd/web-app_2_3.dtd">
   <web-app>
       <display-name>MidoNet API </display-name>
       <!-- REST API configuration -->
       <!-- This value overrides the default base URI. This is typically set
   if you are proxying the API server and the base URI that the clients use to
    access the API is different from the actual server base URI. -->
       <context-param>
           <param-name>rest_api-base_uri </param-name>
           <param-value>http://127.0.0.1:8080/midonet-api </param-value>
       </context-param>
       <!-- CORS configuration -->
       <context-param>
           <param-name>cors-access_control_allow_origin </param-name>
           <param-value>* </param-value>
       </context-param>
       <context-param>
           <param-name>cors-access_control_allow_headers </param-name>
           <param-value>Origin, X-Auth-Token, Content-Type, Accept,
   Authorization </param-value>
       </context-param>
       <context-param>
           <param-name>cors-access_control_allow_methods </param-name>
           <param-value>GET, POST, PUT, DELETE, OPTIONS </param-value>
       </context-param>
       <context-param>
           <param-name>cors-access_control_expose_headers </param-name>
           <param-value>Location </param-value>
       </context-param>
       <!-- Auth configuration -->
       <context-param>
           <param-name>auth-auth_provider </param-name>
           <!-- Specify the class path of the auth service -->
          <param-value> org.midonet.cluster.auth.MockAuthService </param-value>

       </context-param>
       <context-param>
   ```

```
        <param-name>auth-admin_role </param-name>
        <param-value>admin </param-value>
    </context-param>
    <!-- Mock auth configuration -->
    <context-param>
        <param-name>mock_auth-admin_token </param-name>
        <param-value>999888777666 </param-value>
    </context-param>
    <context-param>
        <param-name>mock_auth-tenant_admin_token </param-name>
        <param-value>999888777666 </param-value>
    </context-param>
    <context-param>
        <param-name>mock_auth-tenant_user_token </param-name>
        <param-value>999888777666 </param-value>
    </context-param>
    <!-- Keystone configuration -->
    <context-param>
        <param-name>keystone-service_protocol </param-name>
        <param-value>http </param-value>
    </context-param>
    <context-param>
        <param-name>keystone-service_host </param-name>
        <param-value>127.0.0.1 </param-value>
    </context-param>
    <context-param>
        <param-name>keystone-service_port </param-name>
        <param-value>999888777666 </param-value>
    </context-param>
    <context-param>
        <param-name>keystone-admin_token </param-name>
        <param-value>999888777666 </param-value>
    </context-param>
    <!-- This tenant name is used to get the scoped token from Keystone,
and should be the tenant name of the user that owns the token sent in the
request -->
    <context-param>
        <param-name>keystone-tenant_name </param-name>
        <param-value>admin </param-value>
    </context-param>
    <!-- CloudStack auth configuration -->
    <context-param>
        <param-name>cloudstack-api_base_uri </param-name>
        <param-value>http://127.0.0.1:8080 </param-value>
    </context-param>
    <context-param>
        <param-name>cloudstack-api_path </param-name>
        <param-value>/client/api? </param-value>
    </context-param>
    <context-param>
        <param-name>cloudstack-api_key </param-name>
        <param-value/>
    </context-param>
    <context-param>
        <param-name>cloudstack-secret_key </param-name>
        <param-value/>
    </context-param>
    <!-- Zookeeper configuration -->
    <!-- The following parameters should match the ones in midolman.conf
except 'use_mock' -->
    <context-param>
        <param-name>zookeeper-use_mock </param-name>
        <param-value>false </param-value>
    </context-param>
    <context-param>
        <param-name>zookeeper-zookeeper_hosts </param-name>
        <!-- comma separated list of Zookeeper nodes(host:port) -->
        <param-value>ZOOKEEPER_IP:2181, </param-value>
    </context-param>
    <context-param>
        <param-name>zookeeper-session_timeout </param-name>
        <param-value>30000 </param-value>
    </context-param>
    <context-param>
        <param-name>zookeeper-midolman_root_key </param-name>
        <param-value>/midonet/v1 </param-value>
    </context-param>
    <!-- VXLAN gateway configuration -->
    <context-param>
        <param-name>midobrain-vxgw_enabled </param-name>
        <param-value>false </param-value>
    </context-param>
    <!-- Servlet Listener -->
    <listener>
        <listener-class><!-- Use Jersey's Guice compatible context listener
 -->
            org.midonet.api.servlet.JerseyGuiceServletContextListener
</listener-class>
    </listener>
    <!-- Servlet filter -->
    <filter>
        <!-- Filter to enable Guice -->
        <filter-name>Guice Filter </filter-name>
       <filter-class>com.google.inject.servlet.GuiceFilter </filter-class>

    </filter>
    <filter-mapping>
        <filter-name>Guice Filter </filter-name>
        <url-pattern>/* </url-pattern>
    </filter-mapping>
</web-app>
```

6. Create the file `/etc/tomcat/Catalina/localhost/midonet-api.xml` with this content:

```
<Context path="/midonet-api" docBase="/usr/share/midonet-api"
antiResourceLocking="false" privileged="true"/>
```

7. Modify `/etc/tomcat/server.xml` to allow only localhost access to the API.

   Change the line:

   ```
   Connector port="8080" protocol="HTTP/1.1"
   ```

   ...to:

   ```
   Connector address="127.0.0.1" port="8080" protocol="HTTP/1.1"
   ```

8. Start tomcat:

```
service tomcat start
chkconfig tomcat on
```

9. After approximately one minute, you should be able to access the Midonet API using the Midonet shell:

   ```
   midonet-cli -A --midonet-url=http://127.0.0.1:8080/midonet-api
   ```

   Note: If this command does not work, check `/var/log/tomcat/catalina.out` for possible errors.

Install Midolman on the Cloud Controller

To install the Midolman agent on the node controllers and user facing service:

1. Update to the latest kernel and reboot for the kernel changes to take effect:

   ```
   yum upgrade kernel
   reboot now
   ```

2. Install iproute2, Midolman, and kmod-openvswitch:

   ```
   yum install iproute-netns midolman kmod-openvswitch
   ```

3. Edit the `/etc/midolman/midolman.conf` file and set the zookeeper and cassandra IPs (replace ZOOKEEPER_IP and CASSANDRA_IP in the following example):

   ```
   [zookeeper]
   zookeeper_hosts = ZOOKEEPER_IP:2181
   root_key = /midonet/v1
   [cassandra]
   servers = CASSANDRA_IP
   ```

4. Start midolman:

   ```
   service midolman start
   chkconfig midolman on
   ```

Create a tunnel zone in Midonet and add hosts

In Midonet, a tunnel zone groups hosts together for use as a SDN.

To create a tunnel zone in Midonet:

1. Log into the midonet shell. For example:

   ```
   midonet-cli -A --midonet-url=http://127.0.0.1:8080/midonet-api
   ```

2. Create a GRE tunnel zone:

   ```
   [root@clcfrontend mido-docs]# midonet-cli -A
   --midonet-url=http://127.0.0.1:8080/midonet-api
   midonet> tunnel-zone add name eucatz type gre
   midonet> tunnel-zone list
   tzone tzone0 name eucatz type gre
   midonet> host list
   host host0 name node1 alive true
   host host1 name clcfrontend alive true
   host host2 name node2 alive true
   ```

   You should see a host listed for each of your Node controllers and for your User Facing Service host; if not, check the `/var/log/midolman/midolman.log` log file on the missing hosts to ensure there are no error messages.

3. After verifying all your hosts are listed, add each host to your tunnel zone as follows. ReplacE HOST_N_IP with the IP of your Node Controller or User Facing Host that you used to register the component with Eucalyptus:

   ```
   midonet> tunnel-zone tzone0 add member host host0 address HOST_0_IP
   midonet> tunnel-zone tzone0 add member host host1 address HOST_1_IP
   midonet> tunnel-zone tzone0 add member host host2 address HOST_2_IP
   ```

You are now ready to install and configure Eucalyptus to use this Midonet installation.

Install Repositories

This section guides you through installing Eucalyptus from RPM package downloads.

The first step to installing Eucalyptus is to download the RPM packages. The following terminology might help you as you proceed through this section.

When you're ready, continue to *Software Signing*.

Eucalyptus open source software

Eucalyptus release packages include the freely available components, which enable you to deploy a Eucalyptus cloud.

Eucalyptus enterprise software

Paid subscribers have access to additional software features (for example, SAN support). If you are a subscriber, you receive an entitlement certificate and a private key that allow you to download Eucalyptus subscription software. You will also receive a GPG public key to be used to verify the software integrity.

Euca2ools CLI

Euca2ools is the Eucalyptus command line interface for interacting with web services. It is compatible with many Amazon AWS services, so can be used with Eucalyptus as well as AWS.

RPM and YUM and software signing

Eucalyptus CentOS and RHEL download packages are in RPM (Red Hat Package Manager) format and use the YUM package management tool. We use GPG keys to sign our software packages and package repositories.

EPEL software

EPEL (Extra Packages for Enterprise Linux) are free, open source software, which is fully separated from licensed RHEL distribution. It requires its own package.

Nightly releases

Eucalyptus nightly packages are the latest Eucalyptus builds, which are available for early testing or development work. Nightlies should not be used in production.

Software Signing

This topic describes Eucalyptus software signing keys.

We use a number of GPG keys to sign our software packages and package repositories. The necessary public keys are provided with the relevant products and can be used to automatically verify software updates. You can also verify the packages or package repositories manually using the keys on this page.

Use the `rpm --checksig` command on a download file to verify a RPM package for an HP Helion Eucalyptus product. For example:

```
rpm --checksig -v myfilename.rpm
```

Follow the procedure detailed on Debian's *SecureApt* web page to verify a deb package for an HP Helion Eucalyptus product.

Please do not use package signing keys to encrypt email messages.

The following keys are used for signing Eucalyptus software:

c1240596: Eucalyptus Systems, Inc. (release key) <security@eucalyptus.com>

This key is used for signing HP Helion Eucalyptus products released after July 2011 and their updates.

- *Download from Helion Eucalyptus*
- *Download from pgp.mit.edu*
- Fingerprint: `8639 B2D2 11BB 930D 16A2 D654 BE26 4D09 C124 0596`

0260cf4e: Eucalyptus Systems, Inc. (pre-release key) <security@eucalyptus.com>

This key is used for signing HP Helion Eucalyptus pre-release products due for release after July 2011.

- *Download from Helion Eucalyptus*
- *Download from pgp.mit.edu*
- Fingerprint: `7363 5F5A 9531 308B E83D 3413 8B94 DFB5 0260 CF4E`

9d7b073c: Eucalyptus Systems, Inc. (nightly release key) <security@eucalyptus.com>

This key is used for signing nightly builds of HP Helion Eucalyptus products published after July 2011.

- *Download from Helion Eucalyptus*
- *Download from pgp.mit.edu*
- Fingerprint: `708C DB7C B08A 204E C4CB FA6A 4882 7EF9 9D7B 073C`

Install Eucalyptus Release Packages

To install Eucalyptus from release packages, perform the tasks listed in this topic.

To install Eucalyptus from release packages

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

   ```
   yum install
   http://downloads.eucalyptus.com/software/eucalyptus/4.2/centos/6/x86_64/eucalyptus-release-4.2-1.el6.noarch.rpm
   ```

   Enter `y` when prompted to install this package.

2. (Optional) If you are a Eucalyptus subscriber, you will receive two RPM package files containing your license for subscription-only services. Install these packages on each host machine that will run a Eucalyptus service. Install the license files to access the enterprise repository.

   ```
   yum install eucalyptus-enterprise-license*.noarch.rpm
   http://downloads.eucalyptus.com/software/subscription/eucalyptus-enterprise-release-4.2-1.el6.noarch.rpm
   ```

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

   ```
   yum install
   http://downloads.eucalyptus.com/software/euca2ools/3.3/rhel/6/x86_64/euca2ools-release-3.3-1.el6.noarch.rpm
   ```

   Enter `y` when prompted to install this package.

4. Configure the EPEL package repository on each host machine that will run a Eucalyptus service or Euca2ools:

   ```
   yum install
   http://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
   ```

   Enter `y` when prompted to install this package.

5. If you are installing on RHEL 6, you must 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. The following steps should be performed on each NC host machine.

   a) Install the Eucalyptus Node Controller software on each NC host:

   ```
   yum install eucalyptus-nc
   ```

   b) Remove the default libvirt network. This step allows the eucanetd dhcpd server to start.

   ```
   virsh net-destroy default
   virsh net-autostart default --disable
   ```

   c) 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.

7. On each CLC host machine, install the Eucalyptus Cloud Controller software.

   ```
   yum install eucalyptus-cloud
   ```

8. **Note:** The VPCMIDO network mode currently requires nginx to be installed on the CLC.

   (Optional) If you are using VPCMIDO network mode, install the nginx package with the following command on the CLC:

   ```
   yum install nginx
   ```

   This installs nginx for metadata support.

9. Install the backend service image package on the machine hosting the CLC:

   ```
   yum install eucalyptus-service-image
   ```

   This installs worker images for both the load balancer and imaging services.

10. On the UFS host machine, install the Eucalyptus Cloud Controller software.

    ```
    yum install eucalyptus-cloud
    ```

11. (Optional) On the UFS host machine, also install the Management Console.

    ```
    yum install eucaconsole
    ```

    The Management Console can run on any host machine, even one that does not have other Eucalyptus services. For more information, see the *Console Guide*.

12. Install the software for the remaining Eucalyptus services. The following example shows services being installed on the same host machine. We recommend that you use a different host machine for each service, when possible:

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

    This installs the cloud controller (CC), storage controller (SC), and Walrus Backend services.

13. (Optional) If you are a subscriber and use a SAN, run the appropriate command for your device on each CLC host machine:

    For HP 3PAR SAN:

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

    For NetApp SAN:

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

    For Dell EqualLogic SAN:

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

14. (Optional) If you are a subscriber and use a SAN, run the appropriate command for your device on each SC host machine:

    For HP 3PAR SAN:

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

    For NetApp SAN:

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

For Dell EqualLogic SAN:

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

Your package installation is complete.

You are now ready to *Configure Eucalyptus*.

Install Nightly Release Packages

To install Eucalyptus from nightly builds, perform the tasks listed in this topic.

Prerequisites

- You should have an existing Eucalyptus deployment in a QA test or development environment.
- The prerequisite hardware and software should be in place and available to Eucalyptus.

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

To install Eucalyptus nightly builds:

1. On all host machines, run the following command:

   ```
   yum install
   http://downloads.eucalyptus.com/software/eucalyptus/nightly/4.2/centos/6/x86_64/eucalyptus-release-nightly-4.2-1.el6.noarch.rpm
   ```

   Enter y when prompted to install this package.

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

   ```
   yum install
   http://downloads.eucalyptus.com/software/euca2ools/nightly/3.3/rhel/6/x86_64/euca2ools-release-nightly-3.3-1.el6.noarch.rpm
   ```

   Enter y when prompted to install this package.

3. On all host machines, enter:

   ```
   yum update
   ```

4. Install Eucalyptus packages. The following example shows most services being installed all on the same host machine. You can use a different host for each service.

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

5. Install the backend service image package on the machine hosting the CLC:

   ```
   yum install eucalyptus-service-image
   ```

   This installs worker images for both the load balancer and imaging services.
6. On each planned NC host, install the NC package:

   ```
   yum install eucalyptus-nc
   ```

The nightly package installation is complete.

You are now ready to *Configure Eucalyptus*.

Configure Eucalyptus

This section describes the parameters you need to 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 three Eucalyptus networking modes. Eucalyptus requires network connectivity between its clients (end-users) and the cloud components (CC, CLC, and Walrus).

- In Edge mode, most networking configuration is handled through settings in a global Cloud Controller (CLC) property file. For more information, see *Configure for Edge Mode*.
- In Managed and Managed (No VLAN) modes, traffic to instances pass through the CC. In these two modes clients must be able to connect to the Cluster Controller (CC).

The `/etc/eucalyptus/eucalyptus.conf` file contains some network-related options 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 (8, 16, 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, in Managed mode (only), Eucalyptus uses VLANs at Layer 2 in the networking stack to ensure isolation.
		Managed, Managed (No VLAN)

Option	Description	Modes
`VNET_BRIDGE`	On an NC, this is the name of the bridge interface to which instances' network interfaces should attach. A physical interface that can reach the CC must be attached to this bridge. Common setting for KVM is `br0`.	Edge (on NC) Managed (No VLAN)
`VNET_DHCPDAEMON`	The ISC DHCP executable to use. This is set to a distro-dependent value by packaging. The internal default is `/usr/sbin/dhcpd3`.	Edge (on NC) 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`	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"`	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: `EDGE`, `MANAGED`, `MANAGED-NOVLAN`,	All
`VNET_PRIVINTERFACE`	The name of the network interface that is on the same network as the NCs. In Managed and Managed (No VLAN) modes this must be a bridge for instances in different clusters but in the same security group to be able to reach one another with their private addresses. Default: `eth0`	Edge (on NC) Managed
`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`	Edge (on NC) 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.	Managed, Managed (No VLAN)

Configure for Edge Mode

To configure Eucalyptus for Edge mode, you must edit `eucalyptus.conf` on the Cluster Controller (CC) and Node Controller (NC) hosts. You must also create a JSON file and upload it the Cloud Controller (CLC).

Configure 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="EDGE"
   ```

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

Configure the NC

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 parameters:

   ```
   VNET_MODE
   VNET_PRIVINTERFACE
   VNET_PUBINTERFACE"
   VNET_BRIDGE
   VNET_DHCPDAEMON
   ```

   For example:

   ```
   VNET_MODE="EDGE"
   VNET_PRIVINTERFACE="br0"
   VNET_PUBINTERFACE="br0"
   VNET_BRIDGE="br0"
   VNET_DHCPDAEMON="/usr/sbin/dhcpd"
   ```

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

Create the JSON File

To configure the rest of the Edge mode parameters, you must create a JSON configuration file. Later in the installation process you will upload this file to the CLC.

1. Create the JSON file.
   a) Open a text editor.
   b) Create a file similar to the following structure. Substitute comments for your system settings. See examples at the end of this topic.

      ```
      {
        "InstanceDnsDomain": ""
          "_comment": "Internal DNS domain used for instance private DNS names"
        "InstanceDnsServers": [],
          "_comment": "A list of servers that instances receive to resolve
                       DNS names"
        "PublicIps": [],
          "_comment": "List of public IP addresses"
        "Subnets":   [],
          "_comment": "Subnets you want Eucalyptus to route through the private
                       network rather than the public"
        "MacPrefix": "",
               "_comment": "First 2 octets of any VM's mac address launched"
        "Clusters":  [
          "_comment": "A list of cluster objects that define each
                       availability zone (AZ) in your cloud"
          {
             "Name": "",
               "_comment": "Name of the cluster as it was registered"

             "Subnet": {
               "_comment": "Subnet definition that this cluster will use for
      ```

```
                      private addressing"
         "Name": "",
           "_comment": "Arbitrary name for the subnet"
         "Subnet": "",
           "_comment": "The subnet that will be used for private
                        addressing"
         "Netmask": "",
           "_comment": "Netmask for the subnet defined above"
         "Gateway": "",
           _comment": "Gateway that will route packets for the
                       private subnet"
       },
       "PrivateIps": []
         "_comment": "Private IPs that will be handed out to instances
                      as they launch"
     },
  ]
}
```

2. Save the file.

The following example is for a setup with one cluster (AZ), called PARTI00, with a flat network topology.

```
{
    "InstanceDnsDomain": "eucalyptus.internal",
    "InstanceDnsServers": ["10.1.1.254"],
    "MacPrefix": "d0:0d",
    "PublicIps": [
        "10.111.101.84",
        "10.111.101.91",
        "10.111.101.92",
        "10.111.101.93"
    ],
    "Subnets": [
    ],
    "Clusters": [
        {
            "Name": "PARTI00",
            "Subnet": {
                "Name": "10.111.0.0",
                "Subnet": "10.111.0.0",
                "Netmask": "255.255.0.0",
                "Gateway": "10.111.0.1"
            },
            "PrivateIps": [
                "10.111.101.94",
                "10.111.101.95"
            ]
        },
    ]
}
```

For a multi-cluster deployment, add an additional cluster to your configuration for each cluster you have. The following example has an two clusters, PARTI00 and PARTI01.

```
{
    "InstanceDnsDomain": "eucalyptus.internal",
    "InstanceDnsServers": ["10.1.1.254"],
    "PublicIps": [
        "10.111.101.84",
        "10.111.101.91",
        "10.111.101.92",
        "10.111.101.93"
    ],
    "Subnets": [
    ],
    "Clusters": [
        {
            "Name": "PARTI00",
            "MacPrefix": "d0:0d",
            "Subnet": {
                "Name": "10.111.0.0",
                "Subnet": "10.111.0.0",
                "Netmask": "255.255.0.0",
                "Gateway": "10.111.0.1"
            },
            "PrivateIps": [
                "10.111.101.94",
                "10.111.101.95"
            ]
        },
        {
            "Name": "PARTI01",
            "MacPrefix": "d0:0d",
            "Subnet": {
                "Name": "10.111.0.0",
                "Subnet": "10.111.0.0",
                "Netmask": "255.255.0.0",
                "Gateway": "10.111.0.1"
            },
            "PrivateIps": [
                "10.111.101.96",
                "10.111.101.97"
            ]
        }
    ]
}
```

For more information about multi-cluster configuration, see *Configure Multi-Cluster Networking*.

Configure for Managed Mode

To configure Eucalyptus for Managed mode, you must edit `eucalyptus.conf` on the Cluster Controller (CC) and Node Controller (NC) hosts. You must also create a JSON file and upload it the Cloud Controller (CLC).

Configure 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"
   ```

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

Configure the NC

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 parameters:

   ```
   VNET_MODE
   VNET_PRIVINTERFACE
   VNET_PUBINTERFACE"
   VNET_BRIDGE
   ```

```
VNET_DHCPDAEMON
VNET_SUBNET
VNET_NETMASK
VNET_ADDRSPERNET
VNET_DNS
```

For example:

```
VNET_MODE="MANAGED"
VNET_PRIVINTERFACE="br0"
VNET_PUBINTERFACE="br0"
VNET_BRIDGE="br0"
VNET_DHCPDAEMON="/usr/sbin/dhcpd"
VNET_SUBNET="172.16.0.0"
VNET_NETMASK="255.255.0.0"
VNET_ADDRSPERNET="32"
VNET_DNS="8.8.8.8"
```

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

Create the JSON File

To configure the rest of the Managed mode parameters, you must create a JSON configuration file. Later in the installation process you will upload this file to the CLC.

1. Create the JSON file.
   a) Open a text editor.
   b) Create a file similar to the following structure. Substitute comments for your system settings. See examples at the end of this topic.

```
{
  "InstanceDnsServers": [
    “10.1.1.254"
  ],
  "Clusters": [
    {
      "MacPrefix": "d0:0d",
      "Name": “<clustername>"
    }
  ],
  "PublicIps": [
    "10.111.101.31",
    "10.111.101.40",
    "10.111.101.42",
    "10.111.101.43",
    "10.111.101.132",
    "10.111.101.133",
    "10.111.101.134",
    "10.111.101.135"
  ],
  "Mode": "MANAGED",
  "ManagedSubnet": {
    "Subnet": "172.16.0.0",
    "Netmask": "255.255.0.0",
    "MinVlan": "512",
    "MaxVlan": "639"
  }
}
```

2. Save the file.

Configure for Managed (No-VLAN) Mode

To configure Eucalyptus for Managed (No-VLAN) mode, you must edit `eucalyptus.conf` on the Cluster Controller (CC) and Node Controller (NC) hosts. You must also create a JSON file and upload it the Cloud Controller (CLC).

Configure 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-NOVLAN"
   ```

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

Configure the NC

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 parameters:

   ```
   VNET_MODE
   VNET_PRIVINTERFACE
   VNET_PUBINTERFACE"
   VNET_BRIDGE
   VNET_DHCPDAEMON
   VNET_SUBNET
   VNET_NETMASK
   VNET_ADDRSPERNET
   VNET_DNS
   ```

   For example:

   ```
   VNET_MODE="MANAGED-NOVLAN"
   VNET_PRIVINTERFACE="br0"
   VNET_PUBINTERFACE="br0"
   VNET_BRIDGE="br0"
   VNET_DHCPDAEMON="/usr/sbin/dhcpd"
   VNET_SUBNET="172.16.0.0"
   VNET_NETMASK="255.255.0.0"
   VNET_ADDRSPERNET="32"
   VNET_DNS="8.8.8.8"
   ```

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

Create the JSON File

To configure the rest of the MANAGED-NOVLAN mode parameters, you must create a JSON configuration file. Later in the installation process you will upload this file to the CLC.

1. Create the JSON file.
   a) Open a text editor.
   b) Create a file similar to the following structure. Substitute comments for your system settings. See examples at the end of this topic.

      ```
      {
        "Clusters": [
          {
            "MacPrefix": "d0:0d",
            "Name": "one"
          }
        ],
        "InstanceDnsServers": [
          "10.111.1.56"
        ],
      ```

```
  "ManagedSubnet": {
    "Netmask": "255.255.0.0",
    "Subnet": "172.16.0.0"
  },
  "Mode": "MANAGED-NOVLAN",
  "PublicIps": [
    "10.111.31.177",
    "10.111.31.178",
    "10.111.31.179",
    "10.111.31.180",
    "10.111.31.181",
    "10.111.31.182",
    "10.111.31.183",
    "10.111.31.184"
  ]
}
```

2. Save the file.

Create Scheduling Policy

This topic describes how to set up the Cluster Controller (CC) to choose which Node Controller (NC) to run each new instance.

1. In the CC, open the `/etc/eucalyptus/eucalyptus.conf` file.
2. In the `SCHEDPOLICY=` parameter, set the value to one of the following:

Option	Description
`GREEDY`	When the CC receives a new instance run request, it runs the instance on the first NC in an ordered list of NCs that has capacity to run the instance. At partial capacity with some amount of churn, this policy generally results in a steady state over time where some nodes are running many instances, and some nodes are running few or no instances.
`ROUNDROBIN`	(Default) When the CC receives a new instance run request, it runs the instance on the next NC in an ordered list of NCs that has capacity. The next NC is determined by the last NC to have received an instance. At partial capacity with some amount of churn, this policy generally results in a steady state over time where instances are more evenly distributed across the set of NCs.

3. Save the file.

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.

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 topic briefly describes how Eucalyptus manages the networking aspect of a multi-cluster setup.

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

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

Managed Mode: During normal operation, you will see many tunnel interfaces being created and destroyed as virtual networks are constructed and torn down.

Managed (No VLAN) Mode: In order for VTun tunneling to work in this mode, you must configure each CC with a bridge as its primary, private interface (`VNET_PRIVINTERFACE`). All traffic from nodes in one cluster to nodes in another cluster is routed through the CCs. Each cluster requires that the interface that faces the nodes for the CC (the private interface) be a bridge device for the nodes themselves.

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

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: Note the following:

- You must set VNET_PUBLICIPS identically on all CCs.
- To enable tunneling, set `DISABLE_TUNNELING=N` in `eucalyptus.conf` on all CC hosts.
- 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.
- 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.

Start Eucalyptus

Start the Eucalyptus services in the order presented in this section.

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

Note: Eucalyptus 4.2 requires version 7 of the Java Virtual Machine. Make sure that your CLOUD_OPTS setting in the /etc/eucalyptus/eucalyptus.conf file either does not set `--java-home`, or that `--java-home` points to a version 7 JVM. This needs to happen before services are started.

Start the CLC

Prerequisites

You should have installed and configured Eucalyptus before starting the CLC.

To initialize and start the CLC

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

   Note: Make sure that the `eucalyptus-cloud` process is not running prior to executing this command.

   ```
   clcadmin-initialize-cloud
   ```

   This command might take a minute or more to finish. If it fails, check `/var/log/eucalyptus/cloud-output.log`.
3. If you want the CLC service to start at each boot-time, run this command:

   ```
   chkconfig eucalyptus-cloud on
   ```

4. Enter the following command to start the CLC:

   ```
   service eucalyptus-cloud start
   ```

Start the UFS

Prerequisites

You should have installed and configured Eucalyptus before starting the UFS.

To start the UFS

1. Log in to the User-Facing Services (UFS) host machine.
2. If you want the UFS service to start at each boot-time, run this command:

   ```
   chkconfig eucalyptus-cloud on
   ```

3. Enter the following command to start the UFS:

   ```
   service eucalyptus-cloud start
   ```

4. Repeat for each UFS host machine.

Start Walrus

Prerequisites

You should have installed and configured Eucalyptus before starting the Walrus Backend.

Note: If you not using Walrus as your object storage backend, or if you installed Walrus on the same host as the CLC, you can skip this.

To start the Walrus

1. If you want the Walrus Backend service to start at each boot-time, run this command:

   ```
   chkconfig eucalyptus-cloud on
   ```

2. Log in to the Walrus Backend host machine and enter the following command:

   ```
   service eucalyptus-cloud start
   ```

Start the CC

Prerequisites

You should have installed and configured Eucalyptus before starting the CC.

To start the CC

1. Log in to the Cluster Controller (CC) host machine.
2. If you want the CC service to start at each boot-time, run this command:

```
chkconfig eucalyptus-cc on
```

3. Enter the following command to start the CC:

```
service eucalyptus-cc start
```

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

Start the SC

Prerequisites

You should have installed and configured Eucalyptus before starting the SC.

Note: If you installed SC on the same host as the CLC, you can skip this.

To start the SC

1. Log in to the Storage Controller (SC) host machine.
2. If you want the SC service to start at each boot-time, run this command:

```
chkconfig eucalyptus-cloud on
```

3. Enter the following command to start the SC:

```
service eucalyptus-cloud start
```

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

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

Start the NC

Prerequisites

You should have installed and configured Eucalyptus before starting the NC.

To start the NC

1. Log in to the Node Controller (NC) host machine.
2. If you want the NC service to start at each boot-time, run this command:

```
chkconfig eucalyptus-nc on
```

3. Enter the following command to start the NC:

```
service eucalyptus-nc start
```

4. If you are running in Edge networking mode:
   a) If you want the Edge mode eucanetd service to start at each boot-time, run this command:

```
chkconfig eucanetd on
```

b) Start the Edge eucanetd service:

```
service eucanetd start
```

Note: For VPCMido, eucanetd is only on the CLC.

5. Repeat for each NC host machine.

Start the Management Console

Prerequisites

Before you start the Management Console, ensure that you have reviewed the *Things You Need to Do to Get the Console Running* in the *Management Console Guide*.

Tip: If you plan on running only one Management Console host machine, we recommend co-locating Memcached on that host for optimum speed. You'll find the steps for that in *Configure Memcached*.

1. Log in to the Management Console host machine.
2. If you want the console service to start at each boot-time, run this command:

```
chkconfig eucaconsole on
```

3. Enter the following command to start the console:

```
service eucaconsole start
```

4. Repeat for each Management Console host machine.

Verify the Startup

At this point, all Eucalyptus services 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, and the SC allocate ports 8773. The CC listens to port 8774, and the NC uses port 8775.

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

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

Register Eucalyptus Services

This section describes how to register Eucalyptus services.

Eucalyptus implements a secure protocol for registering separate services so that the overall system cannot be tricked into including a service run by an unauthorized administrator or user.

You need only register services once. Most registration commands run on the CLC server.

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

Registration commands need the following information:

- The **Type** `-t` of service you are registering. Required. For example: `cluster`.
- The **Host** `-h` of the service being registered. Required. The host must be specified by IP address to function correctly.

  Important: IP address is recommended.

  - You must specify public IP addresses.
  - We recommend that you use IP addresses rather than DNS host names when registering Eucalyptus services.

  Important: If you do register a Eucalyptus service with a DNS host name:

  - To avoid connectivity issues, do not change the DNS host name's underlying IP address.
  - The underlying IP address must NOT be a site-local, any-cast, loopback, link-local, or multicast address.
  - Always ensure that DNS is working properly, or populate `etc/hosts`.

- The **Zone** `-z` the service belongs to. This is roughly equivalent to the availability zone in AWS.
- The **Name** `SVCINSTANCE` you assign to each instance of a service, up to 256 characters. Required. This is the name used to identify the service in a human-friendly way. This name is also used when reporting system state changes that require administrator attention.

  Note: The `SVCINSTANCE` name must be globally-unique with respect to other service registrations. To ensure this uniqueness, we recommend using a combination of the service type (CLC, SC, CC, etc.) and system IP address (or DNS host name) when you choose your service instance names. For example: `clc-192.168.0.15` or `clc-eucahost15`.

Register User-Facing Services

This topic describes how to register the User-Facing Services (UFS) with the Cloud Controller (CLC).

Prerequisites

- The Cloud Controller must be properly installed and started.
- The User-Facing Services must be properly installed and started.

To register the User-Facing Services with the Eucalyptus cloud

1. On the CLC host machine, obtain your temporary access keys for the Eucalyptus set up by running the following command:

   ```
   eval `clcadmin-assume-system-credentials`
   ```

   Note: You will create longer-lived and fully functional access keys later.

2. Also on the CLC host machine, run the following command:

   ```
   euserv-register-service -t user-api -h IP SVCINSTANCE
   ```

   where:

   - `IP` is the IP address of the UFS you are registering.
   - `SVCINSTANCE` must be a unique name for the User-Facing service.

   For example:

   ```
   euserv-register-service -t user-api -h 10.111.5.183 user-api-1
   ```

3. Repeat for each UFS host, replacing the UFS IP address and UFS name.
4. Copy the security credentials from the CLC to each machine running User-Facing Services. Run this command on the CLC host machine:

   ```
   clcadmin-copy-keys HOST [HOST ...]
   ```

For example:

```
clcadmin-copy-keys 10.111.5.183
```

5. Verify that the User-Facing service is registered with the following command for each instance of the UFS:

```
euserv-describe-services SVCINSTANCE
```

The registered UFS instances are now ready for your cloud.

Register the Walrus Backend

This topic describes how to register the Walrus Backend service with the Cloud Controller (CLC).

Prerequisites

- You must be using the Walrus Backend service as your object storage provider.
- The Cloud Controller must be properly installed and started.

To register the Walrus Backend service with the Eucalyptus cloud

Note: This task is not necessary if you are using Riak CS instead of Walrus.

1. On the CLC host machine, run the following command:

```
euserv-register-service -t walrusbackend -h IP SVCINSTANCE
```

where:

- `IP` is the IP of the Walrus Backend you are registering with this CLC.
- `SVCINSTANCE` must be a unique name for the Walrus Backend service. We recommend that you use a short-hand name of the hostname or IP address of the machine, for example: `walrus-HOSTNAME` or `walrus-IP_ADDRESS`.

For example:

```
euserv-register-service -t walrusbackend -h 10.111.5.182 walrus-10.111.5.182
```

2. Copy the security credentials from the CLC to each machine running a Walrus Backend service. Run this command on the CLC host machine:

```
clcadmin-copy-keys HOST [HOST ...]
```

For example:

```
clcadmin-copy-keys 10.111.5.182
```

3. Verify that the Walrus Backend service is registered with the following command:

```
euserv-describe-services SVCINSTANCE
```

The registered Walrus Backend service is now ready for your cloud.

Register the Cluster Controller

This topic describes how to register a Cluster Controller (CC) with the Cloud Controller (CLC).

Prerequisites

- The Cloud Controller must be properly installed and started.
- The Cluster Controller service must be properly installed and started.

To register the Cluster Controller service with the Eucalyptus cloud

1. On the CLC host machine, run the following command:

```
euserv-register-service -t cluster -h IP -z ZONE SVCINSTANCE
```

where:

- `IP` is the IP address of the CC you are registering with this CLC.
- `ZONE` name should be a descriptive name for the zone controlled by the CC. For example: `zone-1`.
- `SVCINSTANCE` must be a unique name for the CC service. We recommend that you use the IP address of the machine, for example: `cc-IP_ADDRESS`.

For example:

```
euserv-register-service -t cluster -h 10.111.5.182 -z zone-1 cc-10.111.5.182
```

2. Copy the security credentials from the CLC to each machine running Cluster Controller services. Run this command on the CLC host machine:

   ```
   clcadmin-copy-keys -z ZONE HOST
   ```

   For example:

   ```
   clcadmin-copy-keys -z zone-1 10.111.5.182
   ```

3. Repeat the above steps for each Cluster Controller in each zone.
4. Verify that the Cluster Controller service is registered with the following command:

   ```
   euserv-describe-services SVCINSTANCE
   ```

The registered Cluster Controller service is now ready for your cloud.

Register the Storage Controller

This topic describes how to register a Storage Controller (SC) with the Cloud Controller (CLC).

Prerequisites

- The Cloud Controller must be properly installed and started.
- The Storage Controller service must be properly installed and started.

To register the Storage Controller service with the Eucalyptus cloud

1. Copy the security credentials from the CLC to each machine running Storage Controller services. Run this command on the CLC host machine:

   ```
   clcadmin-copy-keys -z ZONE HOST
   ```

   For example:

   ```
   clcadmin-copy-keys -z zone-1 10.111.5.182
   ```

2. On the CLC host machine, run the following command:

   ```
   euserv-register-service -t storage -h IP -z ZONE SVCINSTANCE
   ```

   where:

   - `IP` is the IP address of the SC you are registering with this CLC.
   - `ZONE` name should be a descriptive name for the zone controlled by the CC. For example: `zone-1`. An SC must have the same `ZONE` name as the CC in the same zone.
   - `SVCINSTANCE` must be a unique name for the SC service. We recommend that you use a short-hand name of the IP address or hostname of the machine, for example: `sc-IP_ADDRESS` or `sc-HOSTNAME`.

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

   For example:

   ```
   euserv-register-service -t storage -h 10.111.5.182 -z zone-1 sc-10.111.5.182
   ```

> **Important:** The SC automatically goes to the `BROKEN` state after being registered with the CLC; it will remain in that state until you explicitly configure the SC by configuring the backend storage provider (later). For more information, see *About the BROKEN state*.

3. Repeat the above steps for each Storage Controller in each zone.
4. Verify that the Storage Controller service is registered with the following command:

   ```
   euserv-describe-services SVCINSTANCE
   ```

The registered Storage Controller service is now ready for your cloud.

Register the Node Controllers

This topic describes how to register a Node Controller (NC) with a Cluster Controller (CC).

Prerequisites

- The Cluster Controller service must be properly installed and started.
- The Node Controller service must be properly installed and started.

To register the Node Controller service with the Eucalyptus cloud

1. SSH to the Cluster Controller in the zone.
2. On the CC, register all NCs using the following command with the IP address of each NC host machine:

   ```
   clusteradmin-register-nodes node0_IP_address ... [nodeN_IP_address]
   ```

   For example:

   ```
   clusteradmin-register-nodes 10.111.5.160 10.111.5.161 10.111.5.162
   ```

3. Copy the CC's security credentials using the following command:

   ```
   clusteradmin-copy-keys node0_IP_address ... [nodeN_IP_address]
   ```

   For example:

   ```
   clusteradmin-copy-keys 10.111.5.160 10.111.5.161 10.111.5.162
   ```

4. Repeat the steps for each zone in your cloud.

The registered Node Controller service is now ready for your cloud.

Configure the Runtime Environment

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

Now that you have installed Eucalyptus, you're ready to begin configuring and using it.

Configure DNS

Eucalyptus provides a DNS service that maps service names, bucket names, and more to IP addresses. This section details how to configure the Eucalyptus DNS service.

> **Important:** Eucalyptus administration tools are designed to work with DNS-enabled clouds, so configuring this service is highly recommended. The remainder of this guide is written with the assumption that your cloud is DNS-enabled.

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 must disable these services.

Configure the Domain and Subdomain

Before using the DNS service, configure the DNS subdomain name that you want Eucalyptus to handle using the steps that follow.

1. Log in to the CLC and enter the following:

   ```
   euctl system.dns.dnsdomain=mycloud.example.com
   ```

2. You can configure the load balancer DNS subdomain. To do so, log in to the CLC and enter the following:

   ```
   euctl services.loadbalancing.dns_subdomain=lb
   ```

Turn on IP Mapping

To enable mapping of instance IPs to DNS host names:

1. Enter the following command on the CLC:

   ```
   euctl bootstrap.webservices.use_instance_dns=true
   ```

   When this option is enabled, public and private DNS entries are created for each launched instance 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.objectstorage.mycloud.example.com`.

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

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

   ```
   euctl 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.mycloud.example.com`.

   Note: The code example above correctly begins with "." before `custom-dns-subdomain`.

Enable DNS Delegation

DNS delegation allows you to forward DNS traffic for the Eucalyptus subdomain to the Eucalyptus CLC host. This host acts as a name server. This allows interruption-free access to Eucalyptus cloud services in the event of a failure. The CLC host is capable of mapping cloud host names to IP addresses of the CLC and UFS / OSG host machines.

For example, if the IP address of the CLC is `192.168.5.1`, and the IP address of Walrus is `192.168.6.1`, the host `compute.mycloud.example.com` will resolve to `192.168.5.1` and `objectstorage.mycloud.example.com` will resolve to `192.168.6.1`.

To enable DNS delegation:

Enter the following command on the CLC:

```
euctl bootstrap.webservices.use_dns_delegation=true
```

Configure the Master DNS Server

Set up your master DNS server to delegate the Eucalyptus subdomain to the UFS host machines, which act as name servers.

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

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

```
zone "example.com" IN {
       type master;
       file "/etc/bind/db.example.com";
       };
```

2. Create `/etc/bind/db.example.com` if it does not exist. If your master DNS is already set up for `example.com`, you will need to add a name server entry for UFS host machines. For example:

   ```
   $ORIGIN example.com.
   $TTL 604800

   @ IN    SOA ns1 admin.example.com 1 604800 86400 2419200 604800
           NS  ns1
   ns1     A   MASTER.DNS.SERVER_IP
   ufs1    A   UFS1_IP
   mycloud NS  ufs1
   ```

   After this, you will be able to resolve your instances' public DNS names such as `euca-A-B-C-D.eucalyptus.mycloud.example.com`.

3. Restart the bind nameserver `service named restart`.
4. Verify your setup by pointing `/etc/resolv.conf` on your client to your primary DNS server and attempt to resolve `compute.example.com` using ping or nslookup. It should return the IP address of a UFS host machine.

Advanced DNS Options

Recursive lookups and split-horizon DNS are available in Eucalyptus.

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

Create the Eucalyptus Cloud Administrator User

After your cloud is running and DNS is functional, create a user and access key for day-to-day cloud administration.

Prerequisites

- Eucalyptus cloud services must be installed and registered.
- Eucalyptus DNS must be configured.

To create a cloud admin user

1. Eucalyptus admin tools and Euca2ools commands need configuration from `~/.euca`. If the directory does not yet exist, create it:

   ```
   mkdir ~/.euca
   ```

2. Choose a name for the new user and create it along with an access key:

   ```
   euare-usercreate -wld DOMAIN USER >~/.euca/FILE.ini
   ```

   where:

   - `DOMAIN` must match the DNS domain chosen in *Configure DNS*.
   - `USER` is the name of the new admin user.
   - `FILE` can be anything; we recommend a descriptive name that includes the user's name.

   This creates a file with a region name that matches that of your cloud's DNS domain; you can edit the file to change the region name if needed.

3. Switch to the new admin user:

```
eval `euare-releaserole`
export AWS_DEFAULT_REGION=REGION
```

where:

- `REGION` must match the region name from the previous step. By default, this is the same as the cloud's DNS domain chosen in *Configure DNS*.

As long as this file exists in `~/.euca`, you can use it by repeating the `export` command above. The remainder of this guide assumes you have completed the above steps. These `euca2ools.ini` configuration files are a flexible means of managing cloud regions and users. See the *Euca2ools Reference Guide* for more information.

Upload the JSON Network Configuration File

This topic describes how to upload the JSON file you configured earlier in the installation process.

To upload the JSON file with your networking configuration:

Important: This step can only be run after getting your credentials in *Create the Eucalyptus Cloud Administrator User*.

Run the following command to upload the configuration file to the CLC (with valid Eucalyptus admin credentials):

```
euctl cloud.network.network_configuration=@/path/to/your/json_config_file
```

Configure Eucalyptus Storage

These are the types of storage available for your Eucalyptus cloud.

Object storage

Eucalyptus provides an AWS S3 compatible object storage service that provides users with web-based general purpose storage, designed to be scalable, reliable and inexpensive. You choose the object storage backend provider: Walrus or Riak. The Object Storage Gateway (OSG) provides access to objects via the backend provider you choose.

Block storage

Eucalyptus provides an AWS EBS compatible block storage service that provides block storage for EC2 instances. Volumes can be created as needed and dynamically attached and detached to instances as required. EBS provides persistent data storage for instances: the volume, and the data on it, can exist beyond the lifetime of an instance. You choose the block storage backend provider, which can be using freely available resources in your cloud, or via a SAN, if you have a subscription (paid).

Configure Object Storage

This topic describes how to configure object storage on the Object Storage Gateway (OSG) for the backend of your choice.

The OSG passes requests to object storage providers and talks to the persistence layer (DB) to authenticate requests. You can use Walrus, Riak CS, or Ceph-RGW as the object storage provider.

- **Walrus** - the default backend provider. It is a single-host Eucalyptus-integrated provider which provides basic object storage functionality for the small to medium scale. Walrus is intended for light S3 usage.

- **Riak Cloud Storage (CS)** - an open source scalable general purpose data platform created by Basho Technologies. It is intended for deployments which have heavy S3 usage requirements where a single-host system like Walrus would not be able to serve the volume of operations and amount of data required.
- **Ceph Rados Gateway (RGW)** - an object storage interface built on top of librados to provide applications with a RESTful gateway to Ceph Storage Clusters. Ceph-RGW uses the Ceph Object Gateway daemon (radosgw), which is a FastCGI module for interacting with a Ceph Storage Cluster. Since it provides interfaces compatible with OpenStack Swift and Amazon S3, the Ceph Object Gateway has its own user management. Ceph Object Gateway can store data in the same Ceph Storage Cluster used to store data from Ceph Filesystem clients or Ceph Block

Device clients. The S3 and Swift APIs share a common namespace, so you may write data with one API and retrieve it with the other.

You must configure the OSG to use one of the backend provider options.

Note: If OSG has been registered but not yet properly configured, it will be listed in the `broken` state when listed with the euserv-describe-services command. For example:

```
[root@g-26-03 ~]# euserv-describe-services --show-headers --filter
service-type=objectstorage
SERVICE  TYPE               ZONE     NAME                      STATE
SERVICE  objectstorage      user-api-1  user-api-1.objectstorage  broken
```

Use Walrus Backend

This topic describes how to configure Walrus as the object storage backend provider for the Object Storage Gateway (OSG).

Prerequisites

- Successful completion of all the install sections prior to this section.
- The UFS must be registered and enabled.
- You must execute the steps below as a Eucalyptus administrator.

To configure Walrus object storage for the OSG

1. Enter `walrus` as the storage provider using the `euctl` command.

   ```
   euctl objectstorage.providerclient=walrus
   ```

2. Check that the OSG is enabled.

   ```
   euserv-describe-services
   ```

   If the state appears as `disabled` or `broken`, check the cloud-*.log files in the `/var/log/eucalyptus` directory. A `disabled` state generally indicates that there is a problem with your network or credentials. See *Eucalyptus Log Files* for more information.

The Walrus backend and OSG are now ready for production.

Use Riak CS

This topic describes how to configure Riak CS as the object storage backend provider for the Object Storage Gateway (OSG).

Prerequisites

- Successful completion of all the install sections prior to this section.
- The UFS must be registered and enabled.
- You must have a functioning Riak CS cluster.
- You must execute the steps below as a Eucalyptus administrator.

For more information on Riak CS, see the *Riak CS documentation*.

To configure Riak CS object storage for the OSG

1. Enter `riakcs` as the storage provider using the `euctl` command.

   ```
   euctl objectstorage.providerclient=riakcs
   ```

2. Specify the RiakCS/S3 endpoint that you want to use with Eucalyptus. For example:

   ```
   euctl objectstorage.s3provider.s3endpoint=riakcs-01.riakcs-cluster.myorg.com
   ```

3. Provide your RiakCS credentials to access your RiakCS installation:

   ```
   euctl objectstorage.s3provider.s3accesskey=RIAK_CS_ACCESS_KEY_ID
   euctl objectstorage.s3provider.s3secretkey=RIAK_CS_SECRET_ACCESS_KEY
   ```

4. After successful configuration, check to ensure that the state of the OSG is `enabled` by running the `euserv-describe-services` command. For example:

   ```
   [root@g-26-03 ~]# euserv-describe-services --show-headers --filter
   service-type=objectstorage
   SERVICE  TYPE                ZONE      NAME                        STATE
   SERVICE  objectstorage       user-api-1  user-api-1.objectstorage  enabled
   ```

   If the state appears as `disabled` or `broken`, check the cloud-*.log files in the `/var/log/eucalyptus` directory. A `disabled` state generally indicates that there is a problem with your network or credentials. See *Eucalyptus Log Files* for more information.

The Riak CS backend and OSG are now ready for production.

Use Ceph-RGW

This topic describes how to configure Ceph Rados Gateway (RGW) as the backend for the Object Storage Gateway (OSG).

Prerequisites

- Successful completion of all the install sections prior to this section.
- The UFS must be registered and enabled.
- A Ceph storage cluster is available.
- The ceph-radosgw service has been installed (on the UFS or any other host) and configured to use the Ceph storage cluster. Eucalyptus recommends using civetweb with ceph-radosgw service. *Civetweb* is a lightweight web server and is included in the ceph-radosgw installation. It is relatively easier to install and configure than the alternative option – a combination of Apache and Fastcgi modules.
- You must execute the steps below as a Eucalyptus administrator.

For more information on Ceph-RGW, see the *Ceph-RGW documentation*.

To configure Ceph-RGW object storage for the OSG

1. Configure `objectstorage.providerclient` to ceph-rgw:

   ```
   euctl objectstorage.providerclient=ceph-rgw
   ```

2. Configure `objectstorage.s3provider.s3endpoint` to the ip:port of the host running the ceph-radosgw service:

   Note: Depending on the front end web server used by ceph-radosgw service, the default port is 80 for apache and 7480 for civetweb.

   ```
   euctl
   objectstorage.s3provider.s3endpoint=<radosgw-host-ip>:<radosgw-webserver-port>
   ```

3. Configure `objectstorage.s3provider.s3accesskey` and `objectstorage.s3provider.s3secretkey` with the radosgw user credentials:

   ```
   euctl objectstorage.s3provider.s3accesskey=<radosgw-user-accesskey>
   ```

   ```
   euctl objectstorage.s3provider.s3secretkey=<radosgw-user-secretkey>
   ```

The Ceph-RGW backend and OSG are now ready for production.

Configure Block Storage

This topic describes how to configure block storage on the Storage Controller (SC) for the backend of your choice.

The Storage Controller (SC) provides functionality similar to the Amazon Elastic Block Store (Amazon EBS). The SC can interface with various storage systems. Eucalyptus block storage (EBS) exports storage volumes that can be attached by a VM and mounted or accessed as a raw block device. EBS volumes can persist past VM termination and are commonly used to store persistent data.

Eucalyptus provides the following open source (free) backend providers for the SC:

- **Overlay** - using the local file system
- **Direct Attached Storage** - DAS-JBOD (just a bunch of disks)
- **Ceph-RBD** - leverages RADOS block device

Eucalyptus also offers the following subscription-based (paid) storage area network (SAN) backend providers for the SC:

- **HP 3PAR** - StorageServ storage systems
- **NetApp** - Clustered Data ONTAP and 7-mode storage systems
- **Dell EqualLogic** - stacked or unstacked storage arrays

You must configure the SC to use one of the backend provider options.

About the BROKEN State

This topic describes the initial state of the Storage Controller (SC) after you have registered it with the Cloud Controller (CLC).

The SC automatically goes to the `broken` state after being registered with the CLC; it will remain in that state until you explicitly configure the SC by telling it which backend storage provider to use.

You can check the state of a storage controller by running `euserv-describe-services --expert` and note the state and status message of the SC(s). The output for an unconfigured SC looks something like this:

```
SERVICE storage         ZONE1         SC71            BROKEN     37
http://192.168.51.71:8773/services/Storage arn:euca:eucalyptus:ZONE1:storage:SC71/
SERVICEEVENT 6c1f7a0a-21c9-496c-bb79-23ddd5749222
arn:euca:eucalyptus:ZONE1: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,ceph
```

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

This indicates that the SC is not yet configured. It can be configured by setting the `ZONE.storage.blockstoragemanager` property to 'das', 'overlay', or 'ceph'.

If you have installed the (paid) Eucalyptus Enterprise packages for your EBS adapter, you will also see additional options in the output line above, and can set the block storage manager to 'netapp', 'equallogic', or 'threepar' as appropriate.

You can verify that the SC block storage manager is unset using:

```
euctl ZONE.storage.blockstoragemanager
```

Use the Overlay Local Filesystem

This topic describes how to configure the local filesystem as the block storage backend provider for the Storage Controller (SC).

Prerequisites

- Successful completion of all the install sections prior to this section.
- The SC must be installed, registered, and running.
- The local filesystem `/var/lib/eucalyptus/volumes` must have enough space to hold volumes and snapshots created in the cloud.
- You must execute the steps below as a Eucalyptus administrator.

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.

To configure overlay block storage for the zone, run the following commands on the CLC

1. Configure the SC to use the local filesystem for EBS.

   ```
   euctl ZONE.storage.blockstoragemanager=overlay
   ```

   The output of the command should be similar to:

   ```
   one.storage.blockstoragemanager=overlay
   ```

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

   ```
   euctl ZONE.storage.blockstoragemanager
   ```

3. Verify that the SC is listed; note that it may be in the `broken` state:

   ```
   euserv-describe-services --filter service-type=storage
   ```

Your local filesystem (overlay) backend is now ready to use with Eucalyptus.

Use Direct Attached Storage (JBOD)

This topic describes how to configure the DAS-JBOD as the block storage backend provider for the Storage Controller (SC).

Prerequisites

- Successful completion of all the install sections prior to this section.
- The SC must be installed, registered, and running.
- Direct Attached Storage requires that `/var/lib/eucalyptus/volumes` have enough space for locally cached snapshots.
- You must execute the steps below as a Eucalyptus administrator.

To configure DAS-JBOD block storage for the zone, run the following commands on the CLC

1. Configure the SC to use the Direct Attached Storage for EBS.

   ```
   euctl ZONE.storage.blockstoragemanager=das
   ```

   The output of the command should be similar to:

   ```
   one.storage.blockstoragemanager=das
   ```

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

   ```
   euctl ZONE.storage.blockstoragemanager
   ```

3. Verify that the SC is listed; note that it may be in the `broken` state:

   ```
   euserv-describe-services --filter service-type=storage
   ```

4. 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.

   ```
   euctl ZONE.storage.dasdevice=DEVICE_NAME
   ```

   For example:

   ```
   euctl one.storage.dasdevice=/dev/sdb
   ```

Your DAS-JBOD backend is now ready to use with Eucalyptus.

Use Ceph-RBD

This topic describes how to configure Ceph-RBD as the block storage backend provider for the Storage Controller (SC).

Prerequisites

- Successful completion of all the install sections prior to this section.
- The SC must be installed, registered, and running.
- You must execute the steps below as a Eucalyptus administrator.

- You must have a functioning Ceph cluster.
- Ceph user credentials with the following privileges are available to Eucalyptus SCs and NCs (different user credentials can be used for the SCs and NCs).
  - Ceph user privileges for credentials assigned to SCs
    - Read, write, execute (rwx) access to the pools used for storing EBS volumes and snapshots
    - Execute (x) access to all pools (Ceph users must have execute permissions to use Ceph’s administrative commands such as unprotecting snapshots)
    - Read (r) access to all monitors
  - Ceph user privileges for credentials assigned to NCs
    - Read, write, execute (rwx) access to the pools used for storing EBS volumes only
    - Read (r) access to all monitors
- Hypervisor support for Ceph-RBD on NCs. Node Controllers (NCs) are designed to communicate with the Ceph cluster via libvirt. This interaction requires a hypervisor that supports Ceph-RBD. See *Configure Hypervisor Support for Ceph-RBD* to satisfy this prerequisite.

To configure Ceph-RBD block storage for the zone, run the following commands on the CLC

1. Configure the SC to use Ceph-RBD for EBS.

   ```
   euctl ZONE.storage.blockstoragemanager=ceph-rbd
   ```

   The output of the command should be similar to:

   ```
   one.storage.blockstoragemanager=ceph-rbd
   ```

2. Verify that the property value is now `ceph-rbd`:

   ```
   euctl ZONE.storage.blockstoragemanager
   ```

3. Check the SC to be sure that it has transitioned out of the `BROKEN` state and is in the `NOTREADY`, `DISABLED` or `ENABLED` state before configuring the rest of the properties for the SC.
4. The ceph-rbd provider will assume defaults for the following properties for the SC:

   ```
   euctl ZONE.storage.ceph

   PROPERTY        one.storage.cephconfigfile  /etc/ceph/ceph.conf
   DESCRIPTION    one.storage.cephconfigfile  Absolute path to Ceph configuration
    (ceph.conf) file. Default value is '/etc/ceph/ceph.conf'

   PROPERTY        one.storage.cephkeyringfile
   /etc/ceph/ceph.client.eucalyptus.keyring
   DESCRIPTION     one.storage.cephkeyringfile Absolute path to Ceph keyring
   (ceph.client.eucalyptus.keyring) file. Default value is
   '/etc/ceph/ceph.client.eucalyptus.keyring'

   PROPERTY        one.storage.cephsnapshotpools       rbd
   DESCRIPTION     one.storage.cephsnapshotpools       Ceph storage pool(s) made
    available to Eucalyptus for EBS snapshots. Use a comma separated list for
   configuring multiple pools. Default value is 'rbd'

   PROPERTY        one.storage.cephuser        eucalyptus
   DESCRIPTION    one.storage.cephuser       Ceph username employed by Eucalyptus
    operations. Default value is 'eucalyptus'

   PROPERTY        one.storage.cephvolumepools rbd
   DESCRIPTION    one.storage.cephvolumepools Ceph storage pool(s) made available
    to Eucalyptus for EBS volumes. Use a comma separated list for configuring
   multiple pools. Default value is 'rbd'
   ```

5. The following steps are optional if the default values do not work for your cloud:

    a) To set the Ceph username (the default value for Eucalyptus is 'eucalyptus'):

    ```
    euctl ZONE.storage.cephuser=myuser
    ```

    b) To set the absolute path to keyring file containing the key for the 'eucalyptus' user (the default value is '/etc/ceph/ceph.client.eucalyptus.keyring'):

    ```
    euctl ZONE.storage.cephkeyringfile='/etc/ceph/ceph.client.myuser.keyring'
    ```

    Note: If cephuser was modified, ensure that cephkeyringfile is also updated with the location to the keyring for the specific cephuser:

    c) To set the absolute path to ceph.conf file (default value is '/etc/ceph/ceph.conf'):

    ```
    euctl ZONE.storage.cephconfigfile=/path/to/ceph.conf
    ```

    d) To change the comma-delimited list of Ceph pools assigned to Eucalyptus for managing EBS volumes (default value is 'rbd') :

    ```
    euctl ZONE.storage.cephvolumepools=rbd,myvolumes
    ```

    e) To change the comma-delimited list of Ceph pools assigned to Eucalyptus for managing EBS snapshots (default value is 'rbd') :

    ```
    euctl ZONE.storage.cephsnapshotpools=mysnapshots
    ```

6. Every NC will assume the following defaults:

    ```
    CEPH_USER_NAME="eucalyptus"
    CEPH_KEYRING_PATH="/etc/ceph/ceph.client.eucalyptus.keyring"
    CEPH_CONFIG_PATH="/etc/ceph/ceph.conf"
    ```

    a) To override the above defaults, add/edit the following properties in the `/etc/eucalyptus/eucalyptus.conf` on the specific NC file:

    ```
    CEPH_USER_NAME="ceph-username-for-use-by-this-NC"
    CEPH_KEYRING_PATH="path-to-keyring-file-for-ceph-username"
    CEPH_CONFIG_PATH="path-to-ceph.conf-file"
    ```

    b) Repeat this step for every NC in the specific Eucalyptus zone.

Your Ceph backend is now ready to use with Eucalyptus.

Configure Hypervisor Support for Ceph-RBD

This topic describes how to configure the hypervisor for Ceph-RBD support.

The following instructions will walk you through steps for verifying and or installing the required hypervisor for Ceph-RBD support.

Repeat this process for every NC in the Eucalyptus zone

1. Verify if `qemu-kvm` and `qemu-img` are already installed.

    ```
    rpm -q qemu-kvm qemu-img
    ```

    Proceed to the preparing the RHEV qemu packages step if they are not installed.

2. Verify qemu support for the `ceph-rbd` driver.

    ```
    qemu-img --help
    qemu-img version 0.12.1, Copyright (c) 2004-2008 Fabrice Bellard
    ...
    Supported formats: raw cow qcow vdi vmdk cloop dmg bochs vpc vvfat qcow2 qed
    vhdx parallels nbd blkdebug host_cdrom
    host_floppy host_device file gluster gluster gluster gluster rbd
    ```

Note: If 'rbd' is listed as one of the supported formats, no further action is required; otherwise proceed to the next step.

3. If the `eucalyptus-nc` service is running, terminate/stop all instances. After all instances are terminated, stop the eucalyptus-nc service.

   ```
   service eucalyptus-nc stop
   ```

4. Prepare the RHEV qemu packages:

   - If this NC is a RHEL system and the RHEV subscription to qemu packages is available, consult the RHEV package procedure to install the qemu-kvm-rhev and qemu-img-rhev packages. Blacklist the RHEV packages in the Eucalyptus repository to ensure that packages from the RHEV repository are installed.
   - If this NC is a RHEL system and RHEV subscription to qemu packages is unavailable, Eucalyptus built and maintained qemu-rhev packages may be used. These packages are available in the same yum repository as other Eucalyptus packages. Note that using Eucalyptus built RHEV packages voids the original RHEL support for the qemu packages.
   - If this NC is a non-RHEL (CentOS) system, Eucalyptus-built and maintained qemu-rhev packages may be used. These packages are available in the same yum repository as other Eucalyptus packages.

5. Install Eucalyptus-built RHEV packages: `qemu-kvm-rhev` and `qemu-img-rhev`, which can be found in the same yum repository as other Eucalyptus packages.

   ```
   yum install qemu-kvm-rhev qemu-img-rhev
   ```

6. Start the `libvirtd` service.

   ```
   service libvirtd start
   ```

7. Verify `qemu` support for the `ceph-rbd` driver.

   ```
   qemu-img --help
   qemu-img version 0.12.1, Copyright (c) 2004-2008 Fabrice Bellard
   ...
   Supported formats: raw cow qcow vdi vmdk cloop dmg bochs vpc vvfat qcow2 qed
   vhdx parallels nbd blkdebug host_cdrom
   host_floppy host_device file gluster gluster gluster gluster rbd
   ```

8. Make sure the eucalyptus-nc service is started.

   ```
   service eucalyptus-nc start
   ```

Your hypervisor is ready for Eucalyptus Ceph-RBD support.

You are now ready to *configure Ceph-RBD* for Eucalyptus.

Use an HP 3PAR SAN

This topic describes how to configure the HP 3PAR SAN as the block storage backend provider for the Storage Controller (SC).

Prerequisites

- Successful completion of all the install sections prior to this section.
- The SC must be installed, registered, and running.
- You must have a paid subscription to Eucalyptus in order to use a SAN backend.
- You must have a functioning 3PAR device available to Eucalyptus cloud.
- Network access must be available from the:
  - SC to 3PAR management and data ports.
  - NC to 3PAR data ports.
- Verify this 3PAR device checklist:

  - Network access for management and data operations is set up.
  - 3PAR Web Services API service is turned on.
  - Common Provisioning Groups (CPG) is created and configured. Recommend one CPG for user data and another for snapshot data.
  - A user exists with "edit" role in the "all" domain.
  - (Optional) Eucalyptus can operate within the scope of a 3PAR virtual domain. This virtual domain should have the necessary CPGs assigned to it. A user with "edit" role in the specific domain as well as "edit" role in the "all" domain must be configured.

- You must execute the steps below as a Eucalyptus administrator.

To configure HP 3PAR SAN block storage for the zone, run the following commands on the CLC

1. Configure the SC to use the 3PAR for EBS.

   ```
   euctl ZONE.storage.blockstoragemanager=threepar
   ```

   The output of the command should be similar to:

   ```
   one.storage.blockstoragemanager=threepar
   ```

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

   ```
   euctl ZONE.storage.blockstoragemanager
   ```

3. Verify that the SC is listed; note that it may be in the `broken` state:

   ```
   euserv-describe-services --filter service-type=storage
   ```

4. On the CLC, enable SAN support in Eucalyptus by entering your SAN's hostname or IP address, the username, password, and the paths:

   ```
   euctl ZONE.storage.sanhost=3PAR_IP_address
   euctl ZONE.storage.sanuser=3PAR_admin_user_name
   euctl ZONE.storage.sanpassword=3PAR_admin_password
   euctl ZONE.storage.scpaths=3PAR_iSCSI_IP
   euctl ZONE.storage.ncpaths=3PAR_iSCSI_IP
   ```

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

5. Assign any string to the `chap_username` property.

   ```
   euctl ZONE.storage.chapuser=chap_username
   ```

6. Assign the 3PAR CPG that should be used for creating virtual volumes to the `threeparusercpg` property.

   ```
   euctl ZONE.storage.threeparusercpg=3PAR_user_cpg
   ```

7. Assign the 3PAR CPG that should be used for creating virtual volume snapshot space to the `threeparcopycpg` property.

   ```
   euctl ZONE.storage.threeparcopycpg=3PAR_copy_cpg
   ```

8. (Optional) These properties are available for advanced configuration.

   ```
   euctl ZONE.storage.threepar
   PROPERTY        one.storage.threepardomain     {}
   DESCRIPTION     one.storage.threepardomain     Name of the virtual domain
   containing threeparusercpg and threeparcopycpg. If threeparusercpg and
   threeparcopycpg don't belong to a specific virtual domain leave this property
    unset

   PROPERTY        one.storage.threeparmultihostaccess     false
   DESCRIPTION     one.storage.threeparmultihostaccess     Configure multi host
   ```

```
access to virtual volume. Value must be true to enable multi host access.
Default value is false

PROPERTY        one.storage.threeparoptimizesnaptovol     true
DESCRIPTION     one.storage.threeparoptimizesnaptovol     If set to true,
snapshot to volume creation path is optimized. If set to false, volume to
snapshot path is optimized. Default value is true

PROPERTY        one.storage.threeparpersona     2
DESCRIPTION     one.storage.threeparpersona     Persona (integer value) to be
 used when exporting a VLUN to host. Default value is 2 and represents a Linux
 initiator

PROPERTY        one.storage.threeparphysicalcopytimeout     120
DESCRIPTION     one.storage.threeparphysicalcopytimeout     Time duration in
minutes to wait for physical copy operation to complete. Default value is 120

PROPERTY        one.storage.threeparusetpvv     true
DESCRIPTION     one.storage.threeparusetpvv     Configure virtual volumes to
be either thinly (TPVV) or fully provisioned (FPVV). Value must be true for
TPVV and false for FPVV. Default value is true

PROPERTY        one.storage.threeparvluncachesize     10000
DESCRIPTION     one.storage.threeparvluncachesize     Maximum number of VLUNs
 that can be cached by the provider. Default value is 10000
```

For more information about the `threeparoptimizesnaptovol` property, and how to configure it, see *About Operation Mode Optimization*.

Your 3PAR SAN backend is now ready to use with Eucalyptus.

About Operation Mode Optimization

This topic describes the operation modes available for 3PAR SAN backends for Eucalyptus cloud.

The Eucalyptus 3PAR backend provider implements a mapping between EBS and 3PAR concepts. Operation mode optimization allows you to make storage operations more efficient in your cloud. Predominant use cases are described below.

Important: Operation modes are mutually exclusive and cannot be combined. Choose the strategy that is best for your 3PAR storage operations. The setting is at the SC level.

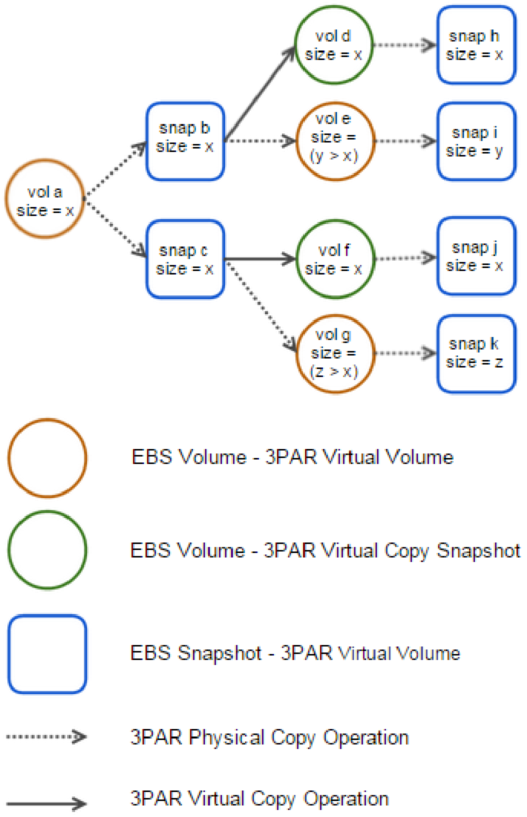
Snapshot to volume optimization

The default use case assumes that you snapshot rarely, create volumes from snapshots (without growing volumes) often, and you run EBS-backed instances often.

Supporting this use case, the default `threeparoptimizesnaptovol` setting is `true`.

Summary of operations:

- EBS volumes and snapshots map to 3PAR virtual volumes
- EBS snapshot from EBS volume translates to 3PAR *physical* copy operation
- EBS volume from EBS snapshot translates to:
    - 3PAR virtual copy operation when both EBS volume and snapshot are of same size
    - 3PAR physical copy operation when EBS volume is greater in size than EBS snapshot


vol a
size = x
snap b
size = x
snap c
size = x
vol d
size = x
vol e
size =
(y > x)
vol f
size = x
vol g
size =
(z > x)
snap h
size = x
snap i
size = y
snap j
size = x
snap k
size = z
EBS Volume - 3PAR Virtual Volume
EBS Volume - 3PAR Virtual Copy Snapshot
EBS Snapshot - 3PAR Virtual Volume
3PAR Physical Copy Operation
3PAR Virtual Copy Operation

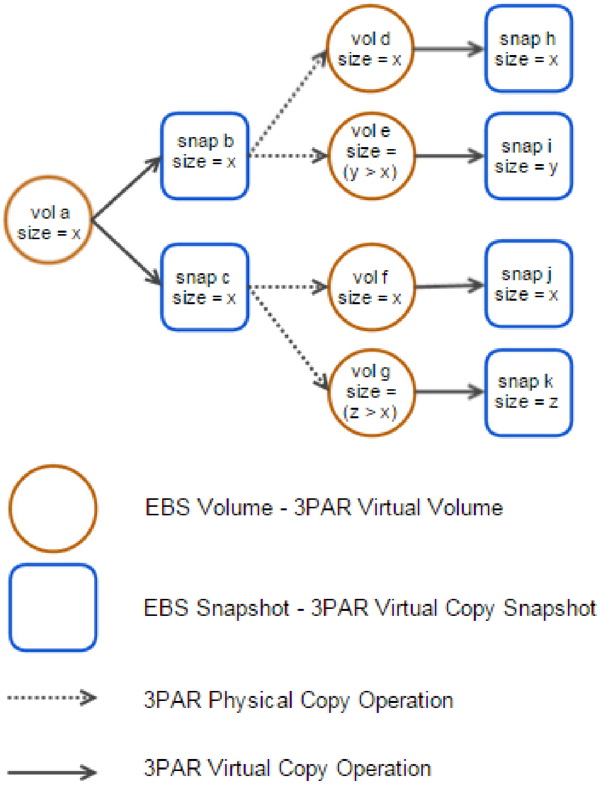
Volume to snapshot optimization

This use case assumes that you snapshot often, but create volumes from snapshots rarely.

If this use case is the best strategy for your storage operations, set `threeparoptimizesnaptovol` to `false`.

Summary of operations:

- EBS volumes and snapshots map to 3PAR virtual volumes
- EBS snapshot from EBS volume translates to 3PAR *virtual* copy operation
- EBS volume from EBS snapshot translates to 3PAR physical copy operation



Use a NetApp SAN

This topic describes how to configure the NetApp Data ONTAP SAN as the block storage backend provider on the Storage Controller (SC).

Eucalyptus supports both NetApp Clustered Data ONTAP and traditional 7-mode SANs. NetApp Vservers and 7-mode Filers are managed by Eucalyptus using NetApp Manageability Software Development Kit (NMSDK) and Data ONTAP APIs.

Enable NetApp 7-mode

This topic describes how to configure the NetApp 7-mode SAN block storage backend for the Storage Controller (SC).

Prerequisites

- Successful completion of all the install sections prior to this section.
- The SC must be installed, registered, and running.
- You must have a paid subscription to Eucalyptus in order to use a SAN backend.
- You must have a functioning NetApp 7-mode device available to Eucalyptus cloud.
- A supported version of the Data ONTAP operating system must be installed on the SAN. See the Compatibility Matrix in the *Release Notes* for supported versions.
- FlexClone and iSCSI licenses must be enabled on the setup.
- One or more aggregates with sufficient space should be available and iSCSI service should be started.
- Administrator account credentials for NetApp Filer must be available to be configured in Eucalyptus.
- You must execute the steps below as a Eucalyptus administrator.

To configure NetApp 7-mode SAN block storage for the zone, run the following commands on the CLC

1. Configure the SC to use the Netapp for EBS.

```
euctl ZONE.storage.blockstoragemanager=netapp
```

The output of the command should be similar to:

```
ZONE.storage.blockstoragemanager=netapp
```

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

   ```
   euctl ZONE.storage.blockstoragemanager
   ```

3. Verify that the SC is listed; note that it may be in the `broken` state:

   ```
   euserv-describe-services --filter service-type=storage
   ```

4. Wait for the SC to transition to the 'notready' or 'disabled' state.
5. On the CLC, 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 was added in Eucalyptus 3.3. An SC will not transition to ENABLED state until the CHAP username is configured.

   ```
   euctl ZONE.storage.sanhost=Filer_IP_address
   euctl ZONE.storage.sanuser=Filer_admin_username
   euctl ZONE.storage.sanpassword=Filer_admin_password
   euctl ZONE.storage.chapuser=Chap_username
   ```

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

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

7. 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:

   ```
   euctl ZONE.storage.aggregate=aggregate_1_name,aggregate_2_name,...
   ```

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

   ```
   euctl ZONE.storage.uselargestaggregate=false
   ```

8. 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.

   ```
   euctl ZONE.storage.ncpaths=IP
   ```

9. 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 objectstorage 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.

   ```
   euctl ZONE.storage.scpaths=IP
   ```

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

Enable NetApp Clustered Data ONTAP

This topic describes how to configure the NetApp Clustered Data ONTAP SAN block storage backend for the Storage Controller (SC).

Prerequisites

- Successful completion of all the install sections prior to this section.
- The SC must be installed, registered, and running.
- You must have a paid subscription to Eucalyptus in order to use a SAN backend.
- You must have a functioning NetApp Clustered Data ONTAP device available to Eucalyptus cloud. See the Compatibility Matrix in the *Release Notes* for supported versions.
- You must have a data Vserver with one or more aggregates and iSCSI protocol for storing and retrieving and data.
- Vserver user with administrator privileges to the specific Vserver should be set up and made available to Eucalyptus.
- FlexClone and iSCSI licenses must be enabled on the setup.
- Management (only) Logical Interface (LIF) should be configured for the Vserver and an IP address or hostname is assigned to it.
- Data LIFs should be configured on the Vserver.
- One or more aggregates with necessary capacity is assigned to the Vserver.
- Network connectivity:
  - 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.
- You must execute the steps below as a Eucalyptus administrator.

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

To configure NetApp Clustered Data ONTAP block storage for the zone, run the following commands on the CLC

1. Configure the SC to use NetApp for EBS:

   ```
   euctl ZONE.storage.blockstoragemanager=netapp
   ```

   The output of the command should be similar to:

   ```
   ZONE.storage.blockstoragemanager=netapp
   ```

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

   ```
   euctl ZONE.storage.blockstoragemanager
   ```

3. Verify that the SC is listed; note that it may be in the `broken` state:

   ```
   euserv-describe-services --filter service-type=storage
   ```

4. Wait for the SC to transition to 'notready' or 'disabled' states.
5. On the CLC, 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 was added in Eucalyptus 3.3. The SC will not transition to ENABLED state until the CHAP username is configured.

   ```
   euctl ZONE.storage.sanhost=Vserver_IP_address
   euctl ZONE.storage.sanuser=Vserver_admin_username
   euctl ZONE.storage.sanpassword=Vserver_admin_password
   euctl ZONE.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:

```
euctl ZONE.storage.vservername=Vserver_name
```

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

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

7. 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:

   ```
   euctl ZONE.storage.aggregate=aggregate_1_name, aggregate_2_name,...
   ```

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

   ```
   euctl ZONE.storage.uselargestaggregate=false
   ```

8. Set an IP address for the iSCSI data LIF on the ENABLED CLC. This is used for NCs performing disk operations on the Vserver.

   ```
   euctl ZONE.storage.ncpaths=IP
   ```

9. 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 objectstorage during snapshot operations.

   ```
   euctl ZONE.storage.scpaths=IP
   ```

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

Use a Dell EqualLogic SAN

This topic describes how to configure the Dell EqualLogic SAN as the block storage backend provider on the Storage Controller (SC).

This task assumes the following:

- Successful completion of all the install sections prior to this section.
- The SC must be installed, registered, and running.
- You must have a paid subscription to Eucalyptus in order to use a SAN backend.
- You must have a functioning EqualLogic device available to Eucalyptus cloud.
- You must execute the steps below as a Eucalyptus administrator.

To configure Dell EqualLogic block storage for the zone, run the following commands on the CLC

1. Configure the SC to use Equallogic for EBS.

   ```
   euctl ZONE.storage.blockstoragemanager=equallogic
   ```

   The output of the command should be similar to:

   ```
   one.storage.blockstoragemanager=equallogic
   ```

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

   ```
   euctl ZONE.storage.blockstoragemanager
   ```

3. Verify that the SC is listed; note that it may be in the `broken` state:

   ```
   euserv-describe-services --filter service-type=storage
   ```

4. Enable SAN support in Eucalyptus by entering your SAN's hostname or IP address, the username, password, and the name of the chap user:

```
euctl ZONE.storage.sanhost=SAN_IP_address
euctl ZONE.storage.sanuser=SAN_admin_user_name
euctl ZONE.storage.sanpassword=SAN_admin_password
euctl ZONE.storage.chapuser=chap_username
```

5. (Optional) If your EqualLogic setup has dedicated paths for data access that are different from the management IP address supplied for the `ZONE.storage.sanhost` property, the following properties must also be configured in Eucalyptus:

   ```
   euctl ZONE.storage.scpaths=data-IP-address ZONE.storage.ncpaths=data-IP-address
   ```

   The SC and NC data IP address might be the same; or they can be different, if EqualLogic has multiple data interfaces.

Your Dell EqualLogic SAN backend is now ready to use with Eucalyptus.

Install and Configure the Imaging Service

The Eucalyptus Imaging Service, introduced in Eucalyptus 4.0, makes it easier to deploy EBS images in your Eucalyptus cloud and automates many of the labor-intensive processes required for uploading data into EBS images.

The Eucalyptus Imaging Service is implemented as a system-controlled "worker" virtual machine that is monitored and controlled via Auto Scaling. Once the Imaging Service is configured, the Imaging Service VM will be started automatically upon the first request that requires it: such as an EBS volume ingress. Specifically, in this release of Eucalyptus, these are the usage scenarios for the Eucalyptus Imaging Service:

- ***Importing a raw disk image as a volume:*** If you have a raw disk image (containing either a data partition or a full operating system with a boot record, e.g., an HVM image), you can use the Imaging Service to import this into your Eucalyptus cloud as a volume. This is accomplished with the `euca-import-volume` command. If the volume was populated with a bootable disk, that volume can be snapshotted and registered as an image.
- ***Importing a raw disk image as an instance:*** If you have a raw disk image containing a bootable operating system, you can import this disk image into Eucalyptus as an instance: the Imaging Service automatically creates a volume, registers the image, and launches an instance from the image. This is accomplished with the `euca-import-instance` command, which has options for specifying the instance type and the SSH key for the instance to use.

Install and Register the Imaging Worker Image

Eucalyptus provides a command-line tool for installing and registering the Imaging Worker image. Once you have run the tool, the Imaging Worker will be ready to use.

1. Run the following commands on the machine where you installed the `eucalyptus-service-image` RPM package (it will set the `imaging.imaging_worker_emi` property to the newly created EMI of the imaging worker):

   ```
   esi-install-image --region localhost --install-default
   ```

2. Consider setting the `imaging.imaging_worker_keyname` property to an SSH keyname (previously created with the `euca-create-keypair` command), so that you can perform troubleshooting inside the Imaging Worker instance, if necessary:

   ```
   euctl services.imaging.worker.keyname=mykey
   ```

Managing the Imaging Worker Instance

Eucalyptus automatically starts Imaging Worker instances when there are tasks for workers to perform.

1. The cloud administrator can list the running Imaging Worker instances, if any, by running the command:

   ```
   euca-describe-instances --filter tag-value=euca-internal-imaging-workers
   ```

2. To delete / stop the imaging worker:

   ```
   esi-manage-stack -a delete imaging
   ```

3. To create / start the imaging worker:

   ```
   esi-manage-stack -a create imaging
   ```

4. Consider setting the `imaging.imaging_worker_instance_type` property to an Instance Type with enough ephemeral disk to convert any of your paravirtual images. The Imaging Worker root filesystem takes up about 2GB, so the maximum paravirtual image that the Imaging Worker will be able to convert is the disk allocation of the Instance Type minus 2GBs.

   ```
   euctl services.imaging.worker.instance_type=m3.xlarge
   ```

Troubleshooting Imaging Worker

If the Imaging Worker is configured correctly, users will be able to import data into EBS volumes with `euca-import-*` commands, and paravirtual EMIs will run as instances. In some cases, though, paravirtual images may fail to convert (e.g., due to intermittent network failures or a network setup that doesn't allow the Imaging Worker to communicate with the CLC), leaving the images in a special state. To troubleshoot:

1. If the Imaging Worker Instance Type does not provide sufficient disk space for converting all paravirtual images, the administrator may have to change the Instance Type used by the Imaging Worker. After changing the instance type, the Imaging Worker instance should be restarted by terminating the old Imaging Worker instance:

   ```
   euctl services.imaging.worker.instance_type=m2.2xlarge
   euca-terminate-instances $(euca-describe-instances --filter
   tag-value=euca-internal-imaging-workers | grep INSTANCE | cut -f 2)
   ```

2. If the status of the conversion operation is 'Image conversion failed', but the image is marked as 'available' (in the output of euca-describe-images), the conversion can be retried by running the EMI again:

   ```
   euca-run-instances ...
   ```

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.

Install and Register the Load Balancer Image

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

Note: This command is not necessary if you've already performed the steps in *Install and Configure the Imaging Service*.

Run the following commands on the machine where you installed the `eucalyptus-service-image` RPM package (it will set the `imaging.imaging_worker_emi` property to the newly created EMI of the imaging worker):

```
esi-install-image --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:

   ```
   esi-describe-images
   ```

   This will produce output similar to the followin:

   ```
   SERVICE     VERSION  ACTIVE     IMAGE      INSTANCES
       imaging       2.2      *     emi-573925e5      0
   ```

```
 loadbalancing    2.2      *     emi-573925e5     0
   database       2.2      *     emi-573925e5     0
```

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

```
euctl services.loadbalancing.worker.image
```

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

```
euctl services.loadbalancing.worker.image=emi-12345678
```

Configure Node Controllers

To prevent potential problems, we recommend that you perform the steps listed in this topic on each NC.

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.

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.
   a) Set `CONCURRENT_DISK_OPS` to `1` to serialize all disk-intensive operations. Or ...
   b) Set it to a higher number to increase the amount of disk-intensive operations the NC will perform in parallel.

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 security groups.

Tip: The number of instances per security group can be calculated as follows:

subnets (SGs) = no. hosts / addrspernet
instances per subnet (SG) = addrspernet - 10

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.

   ```
   euctl cluster.maxnetworktag
   euctl cluster.minnetworktag
   ```

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

   ```
   euctl cloud.network.global_max_network_tag=max_vlan_tag
   euctl 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.

Eucalyptus Network Migration and Upgrade

This section details how migrate and upgrade network modes.

Eucalyptus Migration to Edge Networking Mode

You can configure your existing cloud to use Edge networking mode. This topic provides instructions for configuring and installing additional Eucalyptus components in an existing environment for the purpose of moving to Edge.

Important: Migrating to Edge will require downtime of your cloud platform.

1. Terminate all running instances.
   a) Find out which instances are running:

   ```
   euca-describe-instances
   ```

   b) List the instances to terminate:

   ```
   euca-terminate-instances instance_id [instance_id ...]
   ```

2. Shut down all Eucalyptus services. For more information, see *Shutdown Services*.

   ```
   service eucalyptus-cloud stop
   ```

3. Edit all the config files on NC and CC for Edge networking mode. For more information, see *Configure for Edge Mode*.
4. Install `eucanetd` on all NCs.

   ```
   yum install eucanetd
   ```

5. Start `eucanetd` on all NCs

   ```
   service eucanetd start
   ```

6. Start all Eucalyptus services: CLC, CC, WS, SC, NCs. For more information, see *Start Eucalyptus*.
7. Set the Edge JSON property. For more information, see *Create the JSON File*.

Your Edge networking mode is now properly configured.

Upgrade Managed Network Modes

You must generate the JSON network property/configuration string to use managed networking modes in Eucalyptus 4.2. The creation of the network configuration JSON file should be done prior to upgrade. This topic describes how to upgrade managed networking modes (MANAGED and MANAGED-NOVLAN) for Eucalyptus 4.2.

To upgrade managed network modes for Eucalyptus 4.2:

1. Before upgrading, retrieve and note VNET settings from your current installation of Eucalyptus. These are contained in the `/etc/eucalyptus/eucalyptus.conf` file. For example:

   ```
   VNET_PUBLICIPS="10.111.101.31 10.111.101.40 10.111.101.42 10.111.101.43
   10.111.101.132 10.111.101.133 10.111.101.134 10.111.101.135"
   VNET_SUBNET="172.16.0.0"
   VNET_NETMASK="255.255.0.0"
   VNET_ADDRSPERNET="16"
   VNET_DNS="10.1.1.254"
   ```

2. Retrieve cluster properties from your current installation using either the `euctl` command. For example:

```
euctl ZONE.cluster.maxnetworktag=639
euctl ZONE.cluster.minnetworktag=512
```

3. Create the JSON configuration. For this example, save the file as `network.json`. Examples for both MANAGED and MANAGED-NOVLAN are shown below.

a) The following shows an example JSON configuration file for MANAGED mode:

```
{
  "InstanceDnsServers": [
    “10.1.1.254"
  ],
  "Clusters": [
    {
      "MacPrefix": "d0:0d",
      "Name": “<clustername>"
    }
  ],
  "PublicIps": [
    "10.111.101.31",
    "10.111.101.40",
    "10.111.101.42",
    "10.111.101.43",
    "10.111.101.132",
    "10.111.101.133",
    "10.111.101.134",
    "10.111.101.135"
  ],
  "Mode": "MANAGED",
  "ManagedSubnet": {
    "Subnet": "172.16.0.0",
    "Netmask": "255.255.0.0",
    "MinVlan": "512",
    "MaxVlan": "639"
  }
}
```

b) The following shows an example JSON configuration file for MANAGED-NOVLAN mode:

```
{
  "Clusters": [
    {
      "MacPrefix": "d0:0d",
      "Name": "one"
    }
  ],
  "InstanceDnsServers": [
    "10.111.1.56"
  ],
  "ManagedSubnet": {
    "Netmask": "255.255.0.0",
    "Subnet": "172.16.0.0"
  },
  "Mode": "MANAGED-NOVLAN",
  "PublicIps": [
    "10.111.31.177",
    "10.111.31.178",
    "10.111.31.179",
    "10.111.31.180",
    "10.111.31.181",
    "10.111.31.182",
    "10.111.31.183",
    "10.111.31.184"
  ]
}
```

4. Stop all cloud components using the `service` *`component_name`* `stop` command. For example:

   ```
   service eucalyptus-cc stop
   service eucalyptus-cloud stop
   service eucalyptus-nc stop
   ```

5. On the machine for each Eucalyptus service, upgrade Eucalyptus. For example:

   ```
   yum upgrade `euca*`
   ```

6. Start the Eucalyptus services on each of the Eucalyptus host machines. For example:

   ```
   service eucalyptus-cloud start
   ```

7. When the CLC completes database upgrade and becomes enabled, set the 'cloud.network.network_configuration' property to point to the JSON file that was created. For example:

   ```
   euctl cloud.network.network_configuration=@network.json
   ```

8. Upgrade the CC and SC machines. For example:

   ```
   yum upgrade `euca*`
   ```

9. On the SC machine, start the SC services:

   ```
   service eucalyptus-cloud start
   ```

10. On the CC machine, start the CC services:

    ```
    service eucalyptus-cloud start
    ```

11. On the CCs, start EUCANETD.

    ```
    service eucanetd start
    ```

12. Upgrade each NC.

    ```
    yum upgrade `euca*`
    ```

13. Start the NC services on each NC:

    ```
    service eucalyptus-nc start
    ```

14. Start the EUCANETD service on each NC:

    ```
    service eucanetd start
    ```

You have now upgraded your managed network mode for Eucalyptus 4.2.

Eucalyptus Upgrade

This section details the tasks to upgrade your current version of Eucalyptus.

You can upgrade to Eucalyptus 4.2.2 from 4.1.2 or 4.2.1. If your current version is earlier than 4.2.1, see the prescribed paths below. Follow the directions in that version's Installation Guide in the *documentation archive*, and then upgrade to 4.2.2 using the directions in this section.

Warm upgrade

Eucalyptus supports warm upgrade as of the 3.4.2 release. This means you do not need to shut down EBS-backed or instance-store-backed instances in order to upgrade. Auto Scaling instances will likely shut down and be replaced, based on each group's scaling policy and health check criteria.

Note: When you upgrade the underlying OS (RHEL or Centos), this requires a reboot and therefore warm upgrade is not available in any release when you also upgrade your OS.

Prescribed upgrade paths

The following are the prescribed upgrade paths for Eucalyptus versions prior to 4.2.1:

- Upgrade from 3.1.2 -> 3.2.2
- Upgrade from 3.2.2 -> 3.3.2
- Upgrade from 3.3.2 -> 3.4.3
- Upgrade from 3.4.3 -> 4.0.2
- Upgrade from 4.0.2 -> 4.1.2
- Upgrade from 4.1.2 -> 4.2.1

Prepare for Upgrade

This topic helps you prepare for upgrading Eucalyptus.

Prerequisites

Before starting the upgrade, ensure that you have:

- Verified that your hardware and software are compatible with 4.2. See the Compatibility Matrix in the *Release Notes* for supported versions.
- Verified that the required hardware and software are ready and available to Eucalyptus.
- Followed the *prescribed path* of prior Eucalyptus versions, if needed, to prepare for this upgrade.
- Backed up your data and followed best practices for your environment. See *RHEL documentation*.
- Prepared to upgrade *all Eucalyptus services*. Eucalyptus does not support services that are on different release versions. For example, you cannot have a CLC at 4.2.2 and a Walrus at 4.2.1.
- Verified that you already have the repositories installed for Euca2ools and EPEL from your previous installation. If you do not have these installed, see the installation instructions for that version's Installation Guide in the *documentation archive* to find out how to add these to your host machines.
- Fully updated your existing (pre-4.2.2) Eucalyptus services using `yum update` where possible.
- Removed any hand-written repository files for earlier versions of Eucalyptus and Euca2ools from `/etc/yum.repos.d`.

Important:

- Unless otherwise noted, perform the upgrade steps on *every* Eucalyptus host machine.
- It is recommended that you also 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.

- Federated Eucalyptus clouds began with 4.2.0; you can upgrade a 4.2.x cloud to a federated setup. If you have a 4.1.x or earlier cloud, it cannot have any non-Eucalyptus services accounts created, nor can it be an LDAP integrated cloud. For more information, see *Manage Regions in the Administration Guide*.

Tip: You can preview the install and its dependencies by running the following commands. *Be sure and respond with 'N' so you do not start the install before you are ready.*

To pre-test the upgrade of Eucalyptus cloud

The following steps are an optional preview of what the upgrade command will do. If you do not want to do this, continue to *Shutdown Services*.

1. (Optional) Test the new Eucalyptus release package on each host machine that runs a Eucalyptus service:

   ```
   yum install
   http://downloads.eucalyptus.com/software/eucalyptus/4.2/centos/6/x86_64/eucalyptus-release-4.2-1.el6.noarch.rpm
   ```

   Review the dependencies and install package information.

   Enter `N` when prompted so you do **NOT**install the package.

2. (Optional) Test the new Euca2ools release package on each host machine that runs Euca2ools or a Eucalyptus service:

   ```
   yum install
   http://downloads.eucalyptus.com/software/euca2ools/3.3/rhel/6/x86_64/euca2ools-release-3.3-1.el6.noarch.rpm
   ```

   Review the dependencies and install package information.

   Enter `N` when prompted so you do **NOT**install the package.

3. (Optional) If you have a Eucalyptus subscription, test the new subscription release package on each host machine that runs a Eucalyptus service:

   ```
   yum install
   http://downloads.eucalyptus.com/software/subscription/eucalyptus-enterprise-release-4.2-1.el6.noarch.rpm
   ```

   Review the dependencies and install package information.

   Enter `N` when prompted so you do **NOT**install the package.

You are now ready to *Shutdown Services*.

Shutdown Services

This topic describes how to stop all Eucalyptus services.

Prerequisites

See *Prepare for Upgrade* for the complete list of upgrade prerequisites.

The steps you take depend upon where Eucalyptus services are hosted.

To shut down Eucalyptus services

1. Log in to the CLC host machine and shut down the CLC service:

   ```
   service eucalyptus-cloud stop
   ```

2. (Optional) If you have a separate SC host machine, log in and shut down the SC service:

   ```
   service eucalyptus-cloud stop
   ```

3. (Optional) If you have a separate Walrus host machine, log in and shut down the Walrus service:

   ```
   service eucalyptus-cloud stop
   ```

4. (Optional) If you have a separate UFS host machine, log in and shut down the UFS services:

   ```
   service eucalyptus-cloud stop
   ```

5. (Optional) If there are any other Eucalyptus services (for example Walrus, SC, UFS) co-located on the CC host machine, use this command to shut down the other services on the CC host, and in the correct order:

   ```
   service eucalyptus-cloud stop
   ```

6. Log in to each CC host machine and shut down the CC service:

   ```
   service eucalyptus-cc stop
   ```

7. Log in to each NC host machine and shut down eucanetd:

   ```
   service eucanetd stop
   ```

8. Also on each NC, shut down the NC service:

   ```
   service eucalyptus-nc stop
   ```

   Note: Running instances on the NC will continue running. For more information see *Warm Upgrade*.

9. Log in to each Management Console host machine and shut down the console service:

   ```
   service eucaconsole stop
   ```

   For more information, see *Upgrade the Management Console*.

You are now ready to *Upgrade Euca2ools Package Repositories*.

Upgrade Euca2ools Package Repositories

This topic describes the steps to upgrade the Euca2ools package repositories.

Prerequisites

See *Prepare for Upgrade* for the complete list of upgrade prerequisites.

It is recommended (but optional) that you upgrade Euca2ools to the version compatible with Eucalyptus 4.2.2. If you do not install the new version of Euca2ools, you will not be able to use new features from the command line.

To upgrade Euca2ools

1. Enter the following command on each host machine that runs a Eucalyptus service or uses Euca2ools:

   ```
   yum install 
   http://downloads.eucalyptus.com/software/euca2ools/3.3/rhel/6/x86_64/euca2ools-release-3.3-1.el6.noarch.rpm
   ```

   Review the dependencies and install package information.

   Enter Y when prompted to install the package.

2. Enter the following command on each host machine that runs a Eucalyptus service or uses Euca2ools:

   ```
   yum clean all
   ```

3. Enter the following command on each host machine that runs a Eucalyptus service or uses Euca2ools:

   ```
   yum update euca2ools
   ```

   Enter Y when prompted to upgrade Euca2ools.

   This retrieves the package verification keys; for more information, see *Software Signing*.

4. Repeat these steps for each host machine that runs a Eucalyptus service.

You are now ready to *Upgrade Eucalyptus Package Repositories*.

Upgrade Eucalyptus Package Repositories

This topic describes the steps to upgrade the Eucalyptus package repositories.

Prerequisites

See *Prepare for Upgrade* for the complete list of upgrade prerequisites.

You need to upgrade your existing Eucalyptus package repositories to use the new features in 4.2.2.

To upgrade Eucalyptus

1. Enter the following command on each host machine that runs a Eucalyptus service:

   ```
   yum install
   http://downloads.eucalyptus.com/software/eucalyptus/4.2/centos/6/x86_64/eucalyptus-release-4.2-1.el6.noarch.rpm
   ```

   Review the dependencies and install package information.

   Enter `Y` when prompted to install the package.

2. If you are not a Eucalyptus subscriber, skip this step. Install the Eucalyptus subscription package on each host that will run a Eucalyptus service:

   ```
   yum install
   http://downloads.eucalyptus.com/software/subscription/eucalyptus-enterprise-release-4.2-1.el6.noarch.rpm
   ```

   Review the dependencies and install package information.

   Enter `y` when prompted to install these packages.

3. Enter the following command on each host machine that runs a Eucalyptus service:

   ```
   yum clean all
   ```

4. Enter the following command on each host machine that runs a Eucalyptus service:

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

   Enter `Y` when prompted to upgrade Eucalyptus.

   This retrieves the package verification keys; for more information, see *Software Signing*.

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

5. Perform the steps in *Upgrade the Management Console* then return to this section.
6. Enter the following command on each NC:

   ```
   yum install qemu-kvm-rhev
   ```

You are now ready to *Restart Eucalyptus Services*.

Restart Eucalyptus Services

This topic describes how to restart all Eucalyptus services after upgrade.

Prerequisites

You should have successfully completed *Upgrade Eucalyptus Package Repositories* before you begin this process.

You need to restart all Eucalyptus services after upgrade. The steps you take depend upon where Eucalyptus services are hosted.

To restart Eucalyptus services

1. Log in to the CLC host machine and restart the services:

   ```
   service eucalyptus-cloud start
   ```

   If you are upgrading from 4.1.2 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 4.1.2
   at /var/lib/eucalyptus/upgrade/eucalyptus.backup.1446434585...
   #                           UPGRADE INFORMATION
   #============================================================================
   # Old Version:              4.1.2
   # New Version:              4.2.0
   # Upgrade keys:             false              using:

   # Upgrade configuration:    false              using:

   # Upgrade database:         true               using: upgrade_db
   # Same version:             false              using:

   # Start upgrading: db

   Upgrading your database...
   .
   .
   .
   Done upgrading:  db
   done.
   .
   .
   .
   done.
   ```

2. (Optional) If you have a separate SC host machine, log in and restart the services:

   ```
   service eucalyptus-cloud start
   ```

3. (Optional) If you have a separate Walrus host machine, log in and restart the services:

   ```
   service eucalyptus-cloud start
   ```

4. (Optional) If you have a separate UFS host machine, log in and restart the services:

   ```
   service eucalyptus-cloud start
   ```

5. (Optional) If there are any other Eucalyptus services (for example Walrus, SC, UFS) co-located on the CC host machine, use this command to restart the other services on the CC host, and in the correct order:

   ```
   service eucalyptus-cloud start
   ```

6. Log in to each CC host machine and restart the service:

   ```
   service eucalyptus-cc start
   ```

7. If you have a multi-cluster setup, repeat the previous step for each cluster.
8. Log in to each NC server and restart the service:

```
service eucalyptus-nc start
```

9. Log in to each Management Console host machine and restart the service:

   ```
   service eucaconsole start
   ```

   For more information, see *Upgrade the Management Console*.

You are now ready to *Verify the Services*.

Verify the Services

This topic describes how to verify all the services after upgrading.

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 Walrus backend service:

   ```
   euserv-describe-services --filter service-type=walrusbackend
   ```

   Eucalyptus returns a result, as in the following example.

   ```
   SERVICE walrusbackend walrus enabled
   ```

2. Verify your CCs:

   ```
   euserv-describe-services --filter service-type=cluster
   ```

   Eucalyptus returns a list, as in the following example.

   ```
   SERVICE cluster one one-cc enabled
   SERVICE cluster two two-cc enabled
   ```

3. Verify your SCs:

   ```
   euserv-describe-services --filter service-type=storage
   ```

   Eucalyptus returns a list, as in the following example.

   ```
   SERVICE storage one one-sc enabled
   SERVICE storage one one-sc enabled
   ```

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

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

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

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

```
AVAILABILITYZONE        |- m1.xlarge    0002 / 0002   2   1024    20
AVAILABILITYZONE        |- c1.xlarge    0001 / 0001   4   2048    20
```

You are now ready to *Update the Service Images*.

Update the Service Images

This topic describes how to update the service images after the Eucalyptus software upgrade.

As of Eucalyptus 4.2.0, service images are templates for imaging workers, load balancers, and database images, all using the same service image.

1. Install the imaging worker image. Run the following command on the machine where you installed the Eucalyptus imaging worker image:

   ```
   esi-install-image --install-default
   ```

2. **Important:** Notes about the following steps.

   - Perform these steps as a Eucalyptus admin.
   - These steps need to be completed only once after upgrading to a 4.2.x release.

   Run the following commands to clean up the old imaging worker instance:

   ```
   # euscale-describe-auto-scaling-groups
   AUTO-SCALING-GROUP asg-euca-internal-imaging-worker-01
   lc-euca-internal-imaging-worker-01 one  1 1 1 Default
   INSTANCE i-ce92fd76 one InService Healthy lc-euca-internal-imaging-worker-01
   TAG auto-scaling-group asg-euca-internal-imaging-worker-01 Name
   euca-internal-imaging-workers true

   # euscale-update-auto-scaling-group asg-euca-internal-imaging-worker-01
   --launch-configuration lc-euca-internal-imaging-worker-01 --max-size 1
   --min-size 0 --desired-capacity 0
   ```

3. Once the imaging worker instance is terminated, delete the related autoscaling group and launch config:

   ```
   # euscale-delete-auto-scaling-group asg-euca-internal-imaging-worker-01
   # euscale-delete-launch-config lc-euca-internal-imaging-worker-01
   ```

Your upgrade is now complete.

Downgrade a Failed Upgrade

If your upgrade fails, this topic describes how to downgrade your Eucalyptus cloud to an earlier release.

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 Services* and *Upgrade Eucalyptus Package Repositories*.

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

You must *Shutdown Services* before downgrading Eucalyptus.

1. Downgrade to the Eucalyptus 4.2.1 release package on each host machine:

   ```
   yum downgrade
   http://downloads.eucalyptus.com/software/eucalyptus/4.2/centos/6/x86_64/eucalyptus-release-4.2-1.el6.noarch.rpm
   ```

   Enter `y` when prompted, to downgrade the release package.

2. If you have a Eucalyptus subscription, downgrade your subscription release package on each host machine to the release package you used for Eucalyptus 4.2.1:

   ```
   yum downgrade
   http://downloads.eucalyptus.com/software/subscription/eucalyptus-enterprise-release-4.2-1.el6.noarch.rpm
   ```

   Enter `y` when prompted, to downgrade the subscription release package.

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

   ```
   yum clean expire-cache
   ```

4. Log in to each NC host and downgrade it:

   ```
   yum downgrade eucalyptus eucalyptus-admin-tools eucalyptus-axis2c-common
   eucalyptus-blockdev-utils eucalyptus-imaging-toolkit eucalyptus-nc eucanetd
   ```

   Enter `y` when prompted, to downgrade the NC packages.

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 service installed on the machine host. If more than one service requires the same transactional command, you only need to specify that command once per machine host.

Transaction commands for a combined machine host with CLC, Walrus, CC, and SC:

```
downgrade eucalyptus
downgrade eucalyptus-admin-tools
downgrade eucalyptus-axis2c-common
downgrade eucalyptus-blockdev-utils
downgrade eucalyptus-cc
downgrade eucalyptus-cloud
downgrade eucalyptus-common-java
downgrade eucalyptus-common-java-libs
downgrade eucalyptus-sc
downgrade eucalyptus-service-image
downgrade eucalyptus-walrus
downgrade eucanetd
```

CLC transaction commands:

```
downgrade eucalyptus
downgrade eucalyptus-admin-tools
downgrade eucalyptus-axis2c-common
downgrade eucalyptus-blockdev-utils
downgrade eucalyptus-cloud
downgrade eucalyptus-common-java
downgrade eucalyptus-common-java-libs
downgrade eucalyptus-service-image
downgrade eucanetd
```

UFS transaction commands:

```
downgrade eucalyptus
downgrade eucalyptus-admin-tools
downgrade eucalyptus-cloud
downgrade eucalyptus-common-java
downgrade eucalyptus-common-java-libs
downgrade eucanetd
```

CC transaction commands:

```
downgrade eucalyptus
downgrade eucalyptus-admin-tools
downgrade eucalyptus-cc
```

SC transaction commands:

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

Walrus Backend transaction commands:

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

SAN EqualLogic transaction commands:

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

SAN NetApp transaction commands:

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

7. When you have entered all the appropriate yum transaction commands, 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 yum transaction shell:

```
run
```

9. Exit the yum transaction shell using the following command:

```
exit
```

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

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

Enter `y` when prompted, to remove this file.

Important:

Remove this file from every Eucalyptus host machine.

11. Clear out the `/var/run/eucalyptus/classcache/` directory on all Eucalyptus host machines:

```
rm -rf /var/run/eucalyptus/classcache/
```

This deletes 4.2 class file artifacts; they will be regenerated as needed for your downgraded cloud.

Downgrade Euca2ools

If Euca2ools is not the source of upgrade failure, you are not required to downgrade Euca2ools.

1. Downgrade to the Euca2ools 3.3.0 release package on each host machine:

```
yum downgrade
http://downloads.eucalyptus.com/software/euca2ools/3.3/centos/6/x86_64/euca2ools-release-3.3-1.el6.noarch.rpm
```

Enter `y` when prompted, to downgrade the release package.

2. Expire the cache for the yum repositories on each host machine:

```
yum clean expire-cache
```

3. Downgrade to Euca2ools 3.3.0 on each host machine:

```
yum downgrade euca2ools
```

Enter `y` when prompted, to downgrade Euca2ools.

Verify the Downgrade

1. Restart your downgraded cloud.
2. Verify the Eucalyptus versions. For example:

```
# euca-version
euca2ools 3.3.0
eucalyptus 4.2.1
```

3. Verify that all services are `ENABLED`.

Find More Information

This topic explains what to do once you have installed Eucalyptus, including further reading and other resources for understanding your cloud.

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 services, like the Cloud Controller; and resources, like instances and images.
- The *Identity and Access Management (IAM) Guide* provides information to help you securely control access to services and resources for your Eucalyptus cloud users. Refer to this guide to learn more about managing identities, authentication and access control best practices, and specifically managing your users and groups.
- 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 *Image Management Guide* describes how to create and manage images for your cloud.
- The *Management Console Guide* describes how to create and manage cloud resources using the Eucalyptus Management Console.
- The *Euca2ools Reference Guide* describes the Euca2ools commands. Refer to this guide for more information about required and optional parameters for each command. Also includes `euca2ools.ini` information.

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*.
- Subscribe to one or more of the *Eucalyptus mailing lists*, which provide ways to ask questions and get assistance from the community.
- Search for technical articles in the *Knowledge Base* to find answers to your questions and learn about best practices.
- Check out the *Eucalyptus Support* pages for more ideas.

Install Eucalyptus from a Local Package Repository

This topic describes downloading and installing Eucalyptus from a local repository.

In certain situations, you might need to install Eucalyptus from a local repository. For example if:

- Your cloud is behind a firewall
- Your change management requires a local repo
- You have limited access to the Internet

This procedure augments the standard installation instructions, and includes additional instructions for downloading and installing Eucalyptus from a local repository.

To install Eucalyptus from a local repository

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

```
wget -r --no-parent \
http://downloads.eucalyptus.com/software/eucalyptus/4.2/centos/6/x86_64/ \
-P /tmp/eucalyptus
```

2. Download Euca2ools:

```
wget -r --no-parent \
http://downloads.eucalyptus.com/software/euca2ools/3.3/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/4.2/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/3.3/centos/6/x86_64
```

5. Run `yum update`.

Euca2ools Standalone Installation

Euca2ools is the Eucalyptus command line interface for interacting with Eucalyptus. 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://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
   ```

2. Configure the Euca2ools package repository:

   ```
   yum install
   http://downloads.eucalyptus.com/software/euca2ools/3.3/rhel/6/x86_64/euca2ools-release-3.3-1.el6.noarch.rpm
   ```

3. Install Euca2ools:

   ```
   yum install euca2ools
   ```

You've now performed a standalone installation of Euca2ools.

Installation Guide History

This section contains information about changes to the installation documentation in this release.

Section / Topic	Description of Change	Date Changed
Storage and Install	Updates and corrections for the release of Eucalyptus 4.2.2.	April 26, 2016
Dependencies, Imaging, Network, Storage	Updates and corrections.	March 4, 2016
Create the Eucalyptus Cloud Administrator User	New section.	February 29, 2016
Credentials, DNS, Starting, Install Repos, NTP	Updates and corrections.	February 29, 2016
Registering, Planning, Config Dependencies, VPC	Updates and corrections.	January 31, 2016
VPC, Overview, Introduction, Planning, Architecture	Updates and corrections.	December 31, 2015
Upgrade	Updates and corrections.	December 7, 2015
Downgrade	Updates and corrections.	November 6, 2015
Networking	Added Midokura Midonet for VPC support.	October 22, 2015
Storage Controller (SC)	Added HP 3PAR SAN backend.	October 22, 2015
Storage Controller (SC)	Changed Ceph-RBD backend from tech preview to full support.	October 22, 2015
Storage Controller (SC)	Deprecated EMC and multipathing.	October 22, 2015
Storage Controller (SC)	Reorganized the section.	October 22, 2015
High Availability (HA)	Deprecated high availability.	October 22, 2015
Global	Replaced deprecated commands.	October 22, 2015

Index