EUCALYPTUS

Eucalyptus 4.4.5 Installation Guide

Contents

Eucalyptus Network Migration and Upgrade...80

Eucalyptus Upgrade...81

Find More Information...91

Install Eucalyptus from a Local Package Repository..........................92

Euca2ools Standalone Installation..93

Installation Guide History...94

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 7.6 minimal install and a few IP addresses to spare, try the FastStart script. Run the following command as root: ```bash <(curl -Ls https://eucalyptus.cloud/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 9 (2018-12-27 19:47:59 UTC)

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 7
- Red Hat Enterprise Linux (RHEL) 7

Note: References to RHEL in this guide apply equally to CentOS unless otherwise specified.

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, CloudFormation, 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 a high-level architecture of Eucalyptus with its main components.
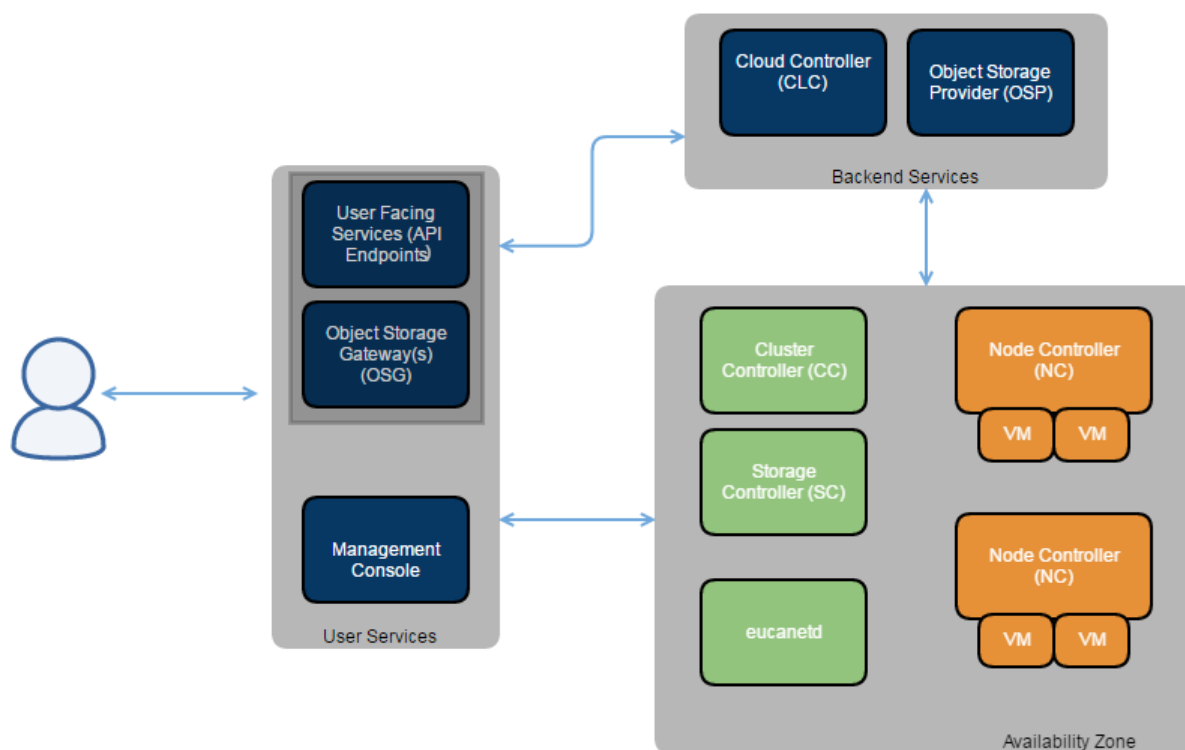
A detailed description of each Eucalyptus component follows.

Cloud Controller

In many deployments, the Cloud Controller (CLC) service 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.

Object Storage Gateway

The Object Storage Gateway (OSG) is part of the UFS. 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.

Object Storage Provider

The Object Storage Provider (OSP) can be either the Eucalyptus Walrus backend, Riak CS, or Ceph-RGW. Walrus is intended for light S3 usage and is a single service. Riak is an open source scalable general purpose data platform; it is intended for deployments with heavy S3 usage. Ceph-RGW is an object storage interface built on top of Librados.

Note: Ceph RGW is a Controlled Availability feature. For information about Controlled Availability features in Eucalyptus, see Special Status Release Features.

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.

Cluster Controller

The Cluster Controller (CC) service must run on a host machine that has network connectivity to the host machines running the Node Controllers (NCs) and to the host machine for the CLC. CCs gather information about a set of NCs and schedules virtual machine (VM) execution on specific NCs. All NCs associated with a single CC must be in the same subnet.

Storage Controller

The Storage Controller (SC) service provides functionality similar to 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) service runs 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.

Eucanetd

The eucanetd service implements artifacts to manage and define Eucalyptus cloud networking. Eucanetd runs alongside the CLC or NC services, depending on the configured networking mode.

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

Compute Requirements

- Physical Machines: All Eucalyptus services must be installed on physical servers, not virtual machines.
- Central Processing Units (CPUs): We recommend that each host machine in your Eucalyptus cloud contain either an Intel or AMD processor with a minimum of 4 2GHz cores.
- Operating Systems: Eucalyptus supports the following Linux distributions: CentOS 7 and RHEL 7. 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 160GB of storage.
- We recommend at least 500GB for Walrus and SC hosts running Linux VMs. We recommend at least 250GB for Walrus and SC hosts running Windows VMs.
- We recommend 160GB 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 8GB RAM. However, we recommend more RAM for improved caching.
- Host machines running multiple services (e.g., CLC, CC and SC) likely need more than the minimum amounts of RAM and storage.

Network Requirements

- All NCs must have access to a minimum of 1Gb Ethernet network connectivity.
- All Eucalyptus components must have at least one Network Interface Card (NIC) for a base-line deployment. For better network isolation and scale, the CC should have two NICs (one facing the CLC/user network and one facing the NC/VM network).
- For EDGE networking mode, Eucalyptus needs at least one existing network.
- For VPCMIDO networking mode, Eucalyptus needs Midokura Enterprise MidoNet to be installed. For more information, see Install Midokura Enterprise MidoNet.
- The network connecting machines that host Eucalyptus components (except the CC and NC) must support UDP multicast for IP address 239.193.7.3. 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.

Note: 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: EC2 Classic Networking or EC2 VPC Networking. Think about the public network (not necessarily the IPv4 public space, but the network that Eucalyptus users will access), delegating IP addresses to Eucalyptus, and firewall rules.

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.

Eucalyptus Architecture Overview

This topics describes the relationship of the components in a Eucalyptus installation.
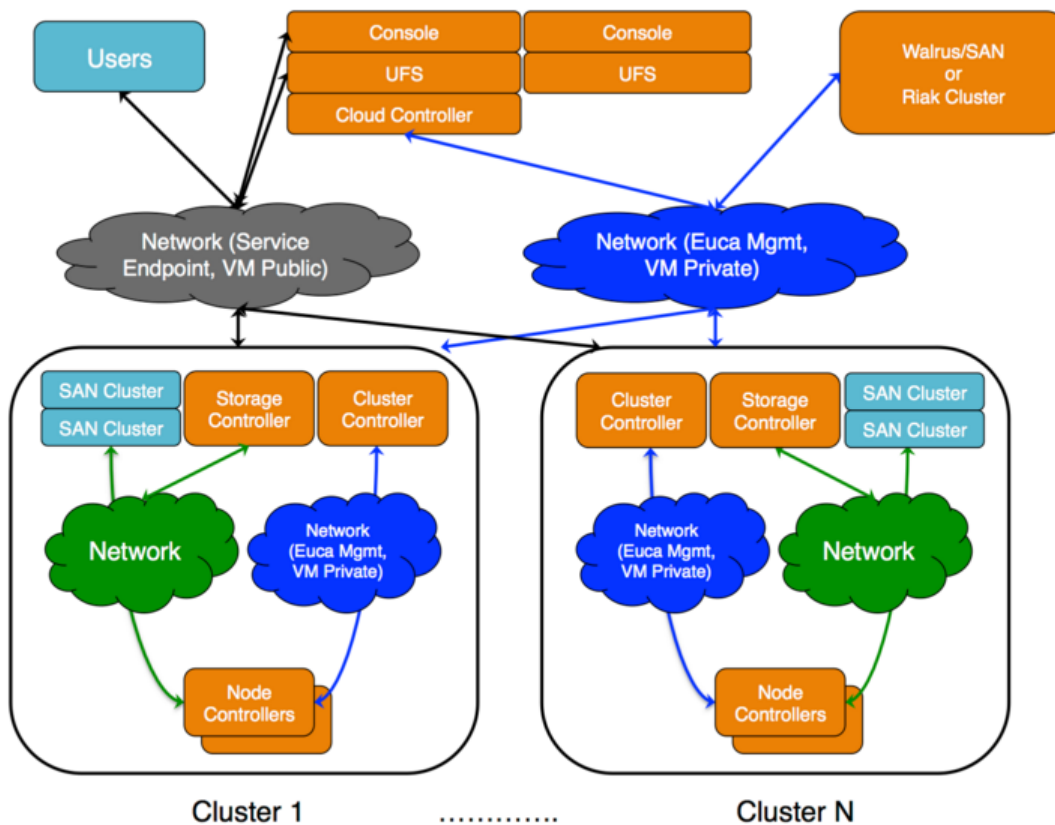
The cloud components: Cloud Controller (CLC) and Walrus, as well as user components: User-Facing Services (UFS) and the Management Console, communicate with cluster components: the Cluster Controllers (CCs) and Storage Controllers (SCs). The CCs and SCs, in turn, communicate with the Node Controllers (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, available to Eucalyptus subscribers, are alternatives to direct-attached storage and Ceph as block storage providers.

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 (CLC), Walrus, CC, and SC on one host machine, and NCs 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 (CLC) and zone (CC) 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 zones, each of which contains a Cluster Controller, a Storage Controller, and one or more Node Controllers.
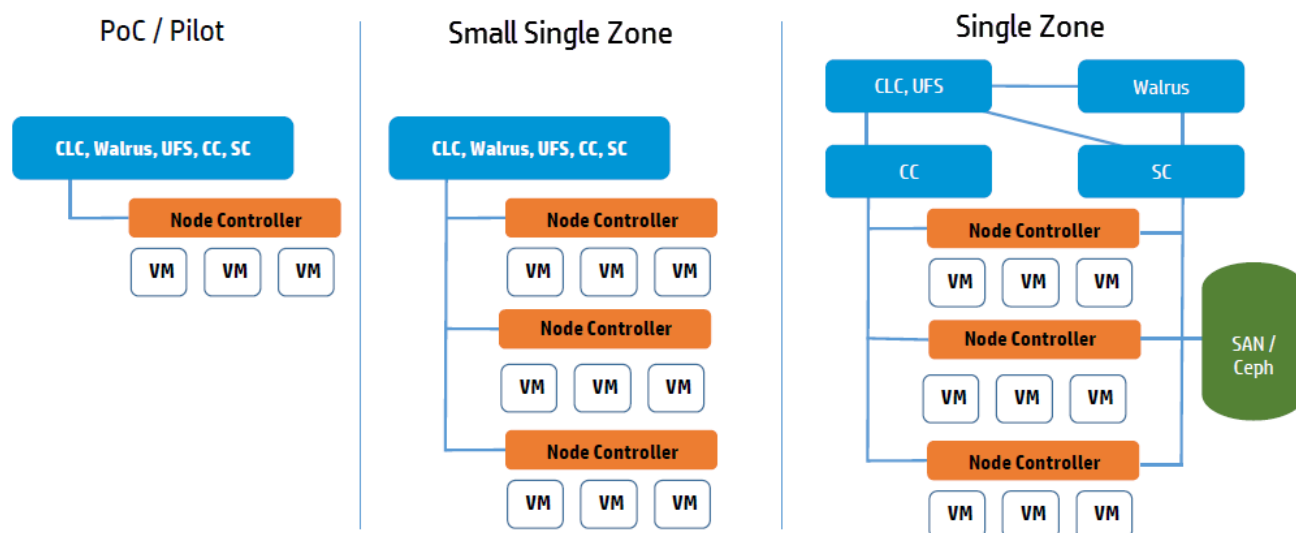
Cloud Services

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 all 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.

Zone Services

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

You can install all zone 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 number of instances (CC) and EBS volume access (SC).

Things to consider for CC placement:

- Place the CC on a server that has TCP/IP connectivity to the Eucalyptus front-end servers and the NC servers in its zone.
- Each CC can manage a maximum of 4000 instances.

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 zone 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

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)	`/var/lib/eucalyptus`	5GB
UFS logging	`/var/log/eucalyptus`	2GB

Service	Directory	Minimum Size
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.

Note: Storage SANs are deprecated as of Eucalyptus 4.4; to be removed in a future release.

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.
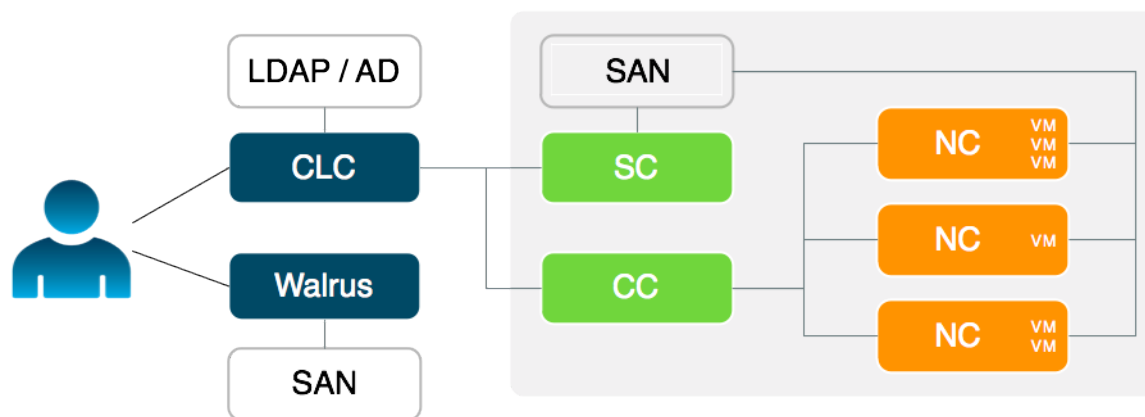
Eucalyptus with SAN support allows you to:

*   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 local availability zones.

An availability zone for AWS denotes a large subset of their cloud environment. Eucalyptus refines this definition to denote a subset of the cloud that shares a local area network. Each Eucalyptus zone has its own Cluster Controller and Storage Controller.
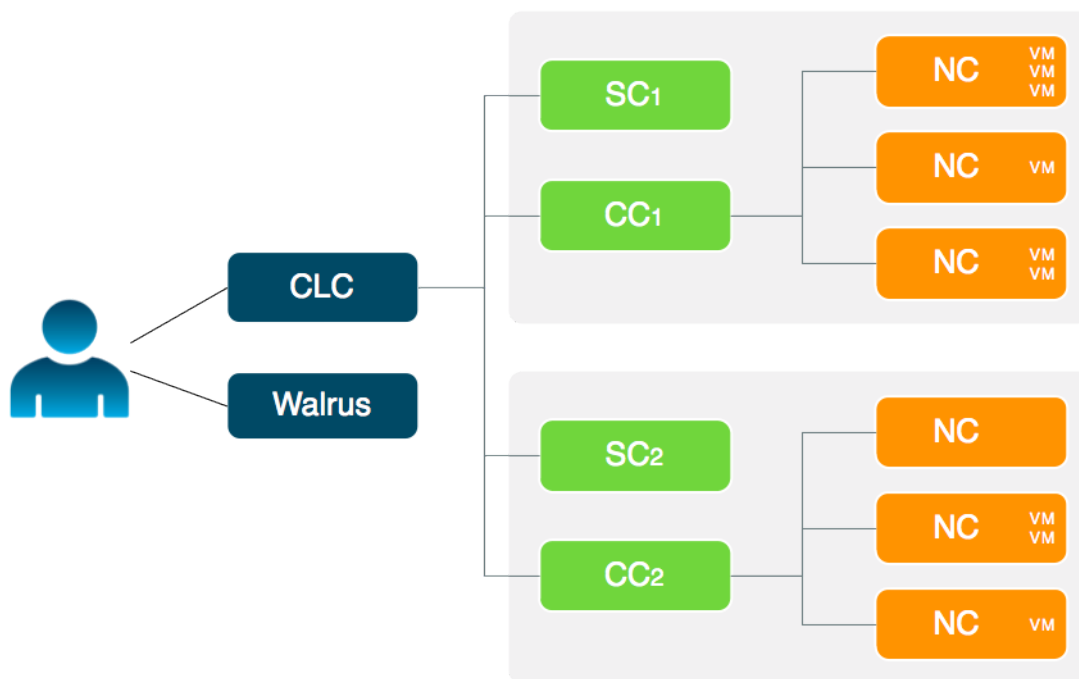
Object Storage

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 these networking modes: EDGE (AWS EC2 Classic compatible) and VPCMIDO (AWS VPC compatible).

These networking modes are designed to allow you to choose an appropriate level of security and flexibility for your cloud. The purpose 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.

Eucalyptus networking modes are generally modeled after AWS networking capabilities. In legacy AWS accounts, you have the ability to choose EC2 Classic network mode or VPC network mode. New AWS accounts do not have this flexibility and are forced into using VPC. Eucalyptus VPCMIDO mode is similar to AWS VPC in that it allows users to fully manage their cloud network, including the definition of a Classless Inter-Domain Routing (CIDR) block, subnets, and security groups with rules for additional protocols beyond the default three (UDP, TCP, and ICMP) available in EC2 Classic networking.

Your choice of networking mode depends on the following considerations:

*   Does your Eucalyptus cloud need to mimic behavior in your AWS account? If you need EC2-Classic behavior, select EDGE mode. If you need EC2-VPC behavior, select VPCMIDO mode.
*   Do you need to create security group rules with additional protocols (e.g., all protocols, RDP, XTP, etc.)? If so, choose VPCMIDO mode.
*   If there is no specific requirement for either mode, then VPCMIDO mode is recommended given its flexibility and networking features.

Each networking mode is described in the following sections.

About Eucanetd

The eucanetd service implements artifacts to manage and define Eucalyptus cloud networking. Eucanetd runs alongside the CLC, CC, and/or NC services, depending on the configured networking mode.

Eucanetd manages network functionality. For example:

- Installs network artifacts (iptables, ipsets, ebtables, dhcpd)
- Performs state management for the installed network artifacts
- Maintains the eucanetd.log file
- Updates network artifact configuration as needed
- In VPCMIDO mode:

    - Interacts with MidoNet via the MidoNet API
    - Defines network artifacts in MidoNet

Where to deploy eucanetd

On a Eucalyptus 4.4 cloud:

Host Machine	EDGE mode	VPCMIDO mode
CLC	No	Only on CLC
CC	No	No
NC	On each NC	No

On a Eucalyptus 4.3 cloud:

Host Machine	EDGE mode	MANAGED modes (deprecated)	VPCMIDO mode
CLC	No	No	Only on CLC
CC	No	On each CC	No
NC	On each NC	On each NC	No

Understanding Eucalyptus EDGE Mode

Eucalyptus 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 EDGE networking mode, the components responsible for implementing Eucalyptus VM networking artifacts are running at the edge of a Eucalyptus deployment: the Linux host machines acting as Node Controllers (NCs). On each NC host machine, a Eucalyptus stand-alone service, eucanetd, runs side-by-side with the NC service. The eucanetd service receives dynamically changing Eucalyptus networking views and is responsible for configuring the Linux networking subsystem 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 integrates with networks that already exist. If the network, netmask, and router don't already exist, you must create them outside Eucalyptus before configuring EDGE mode.

EDGE Mode Requirements

- Each NC host machine 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 host machine (where eucanetd runs) and the CLC host machine, so that network path from instances to the metadata server (running on the CLC host machine) can be established.
- 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 NC host machines 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 NC host machines 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 zones, each zone 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, the CLC host machine must have a route back to the VM private network.

EDGE Mode Limitations

- 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.

Understanding VPCMIDO and MidoNet

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

Eucalyptus VPCMIDO mode resembles the Amazon Virtual Private Cloud (VPC) product wherein the network is fully configurable by users. In Eucalyptus, it is implemented with a Software-Defined Networking (SDN) technology developed by Midokura, called MidoNet. Midokura Enterprise MidoNet is a 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.

This network mode requires configuration of MidoNet in order to make cloud networking functional. It offers the most advanced networking capabilities and therefore it is recommended to be used on all new Eucalyptus installations.

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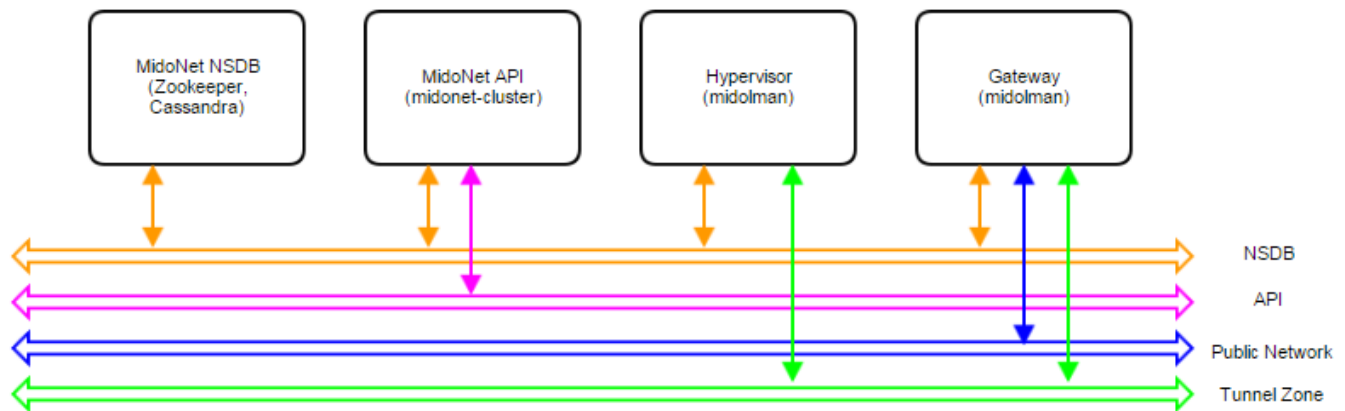
Figure 1: Logical view of a MidoNet deployment. Four components are connected via four networks.

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 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

Eucalyptus has been tested with Midokura Enterprise MidoNet (commercial version with 24/7 support - 30 day evaluation available).

Important: See the Compatibility Matrix in the Release Notes for supported versions.

Eucalyptus with MidoNet

A Eucalyptus with MidoNet deployment consists of the following components:
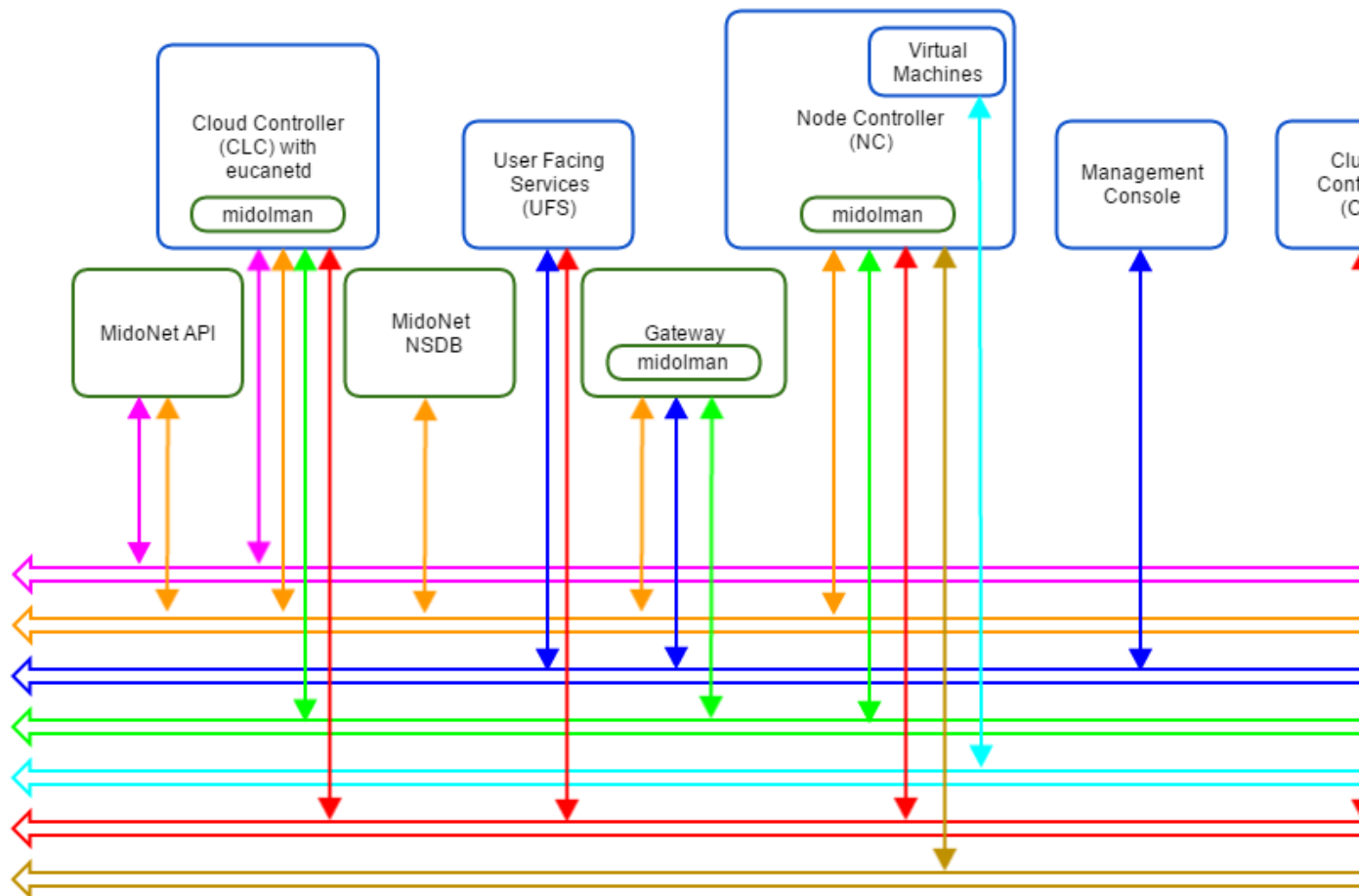
Figure 2: Logical view of a Eucalyptus with MidoNet deployment. VM private network is created/virtualized by MidoNet, and 'software-defined' by eucanetd. Ideally, each component and network should have its own set of independent resources. In practice, components are grouped and consolidated into a set 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/RHEL7 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
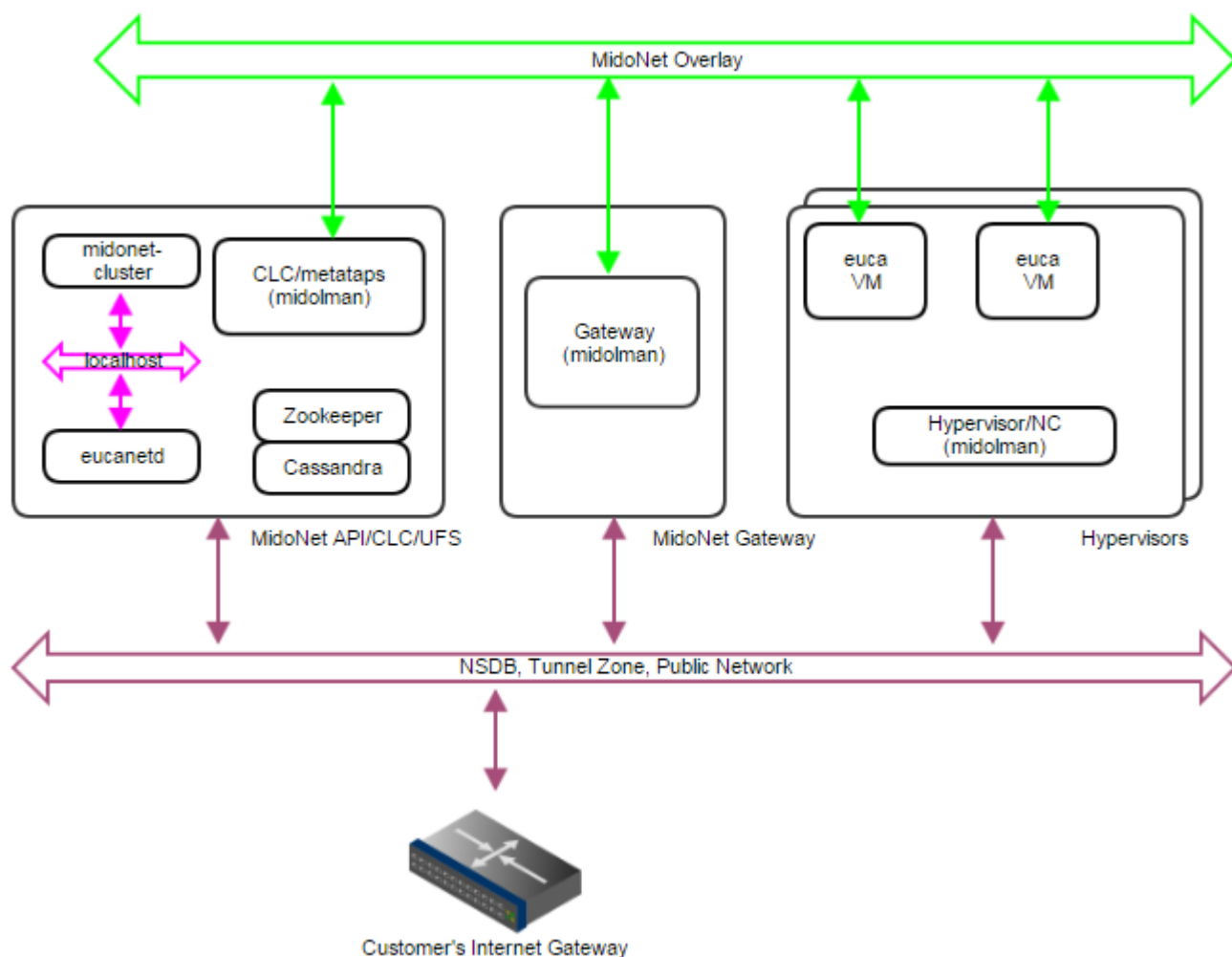
Figure 3: PoC deployment topology. A single IP network carries NSDB, Tunnel Zone, and Public Network traffic. A single server handles MidoNet NSDB, API (and possibly Gateway) functionality.

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

Hypervisors

MidoNet Gateway

- Configure MidoNet router (eucart) to use the Customer Internet Router as a gateway.

tap

euca VM

- tap networ devices of VMs bound to a Mido 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 n
devices o
bound to
virtual

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 p
configures static routes to forward p
Euca_Public_IPs to MidoNet Gatew

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

Hypervisors

tap

-
devic
boun
vi

euca
VM

eth

eth

NSDB, Tunnel Zone, Public Network

No special and/or a
configuration is nee
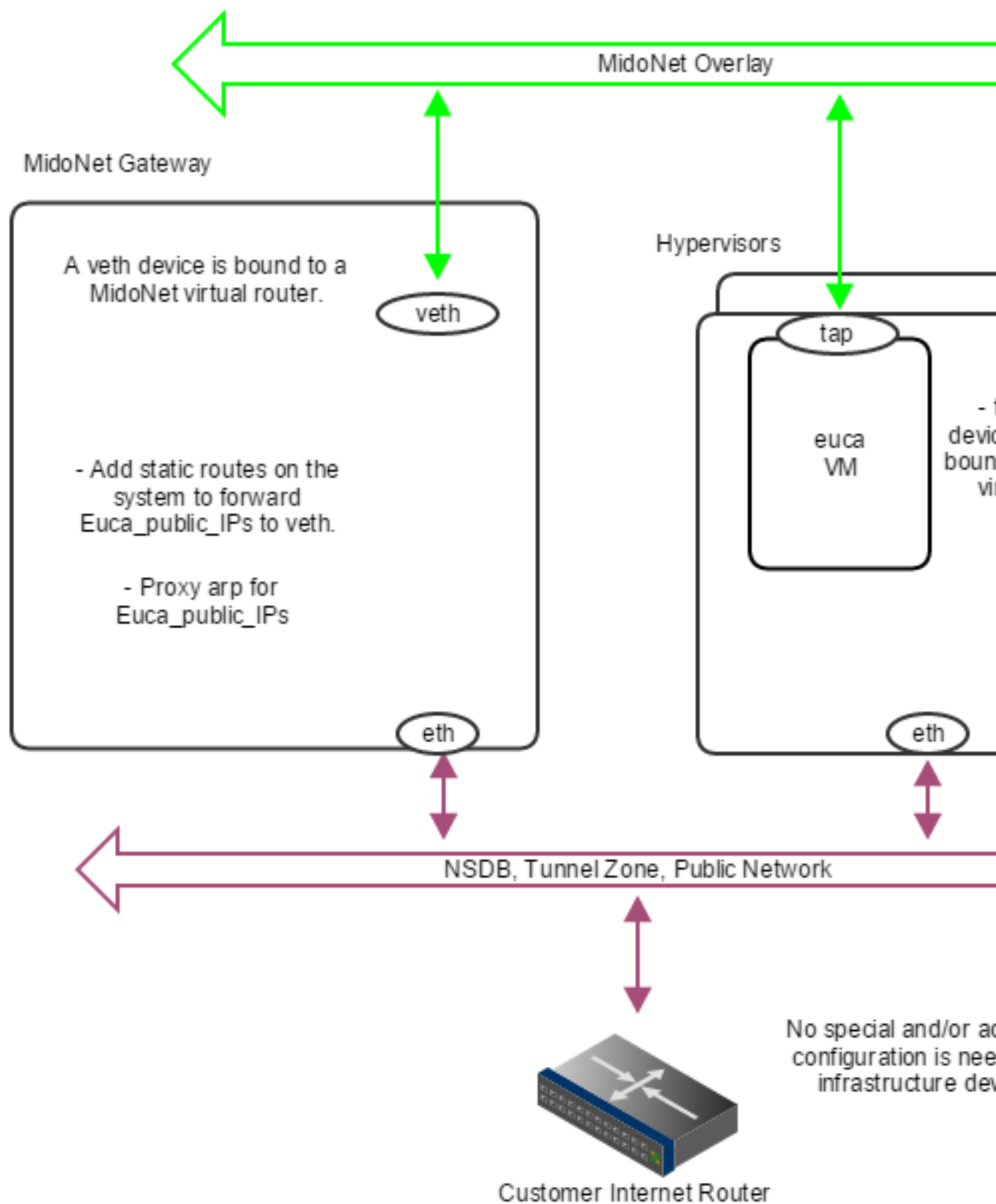infrastructure dev

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 Gateway failover 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
- 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 failover and limited load balancing/sharing.
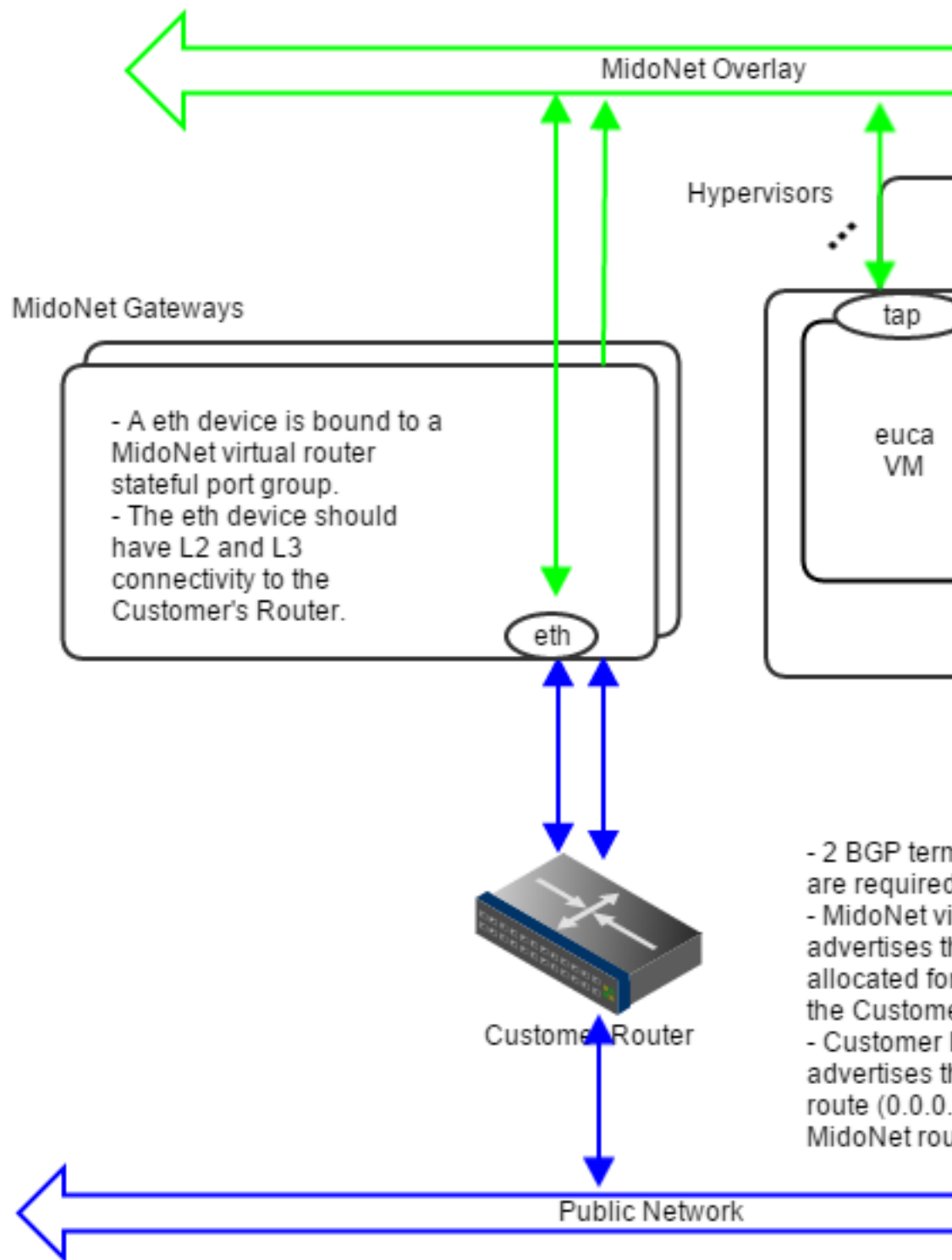
Figure 8: How servers are bound to MidoNet in a Production:Small deployment. Gateway Nodes have physical devices bound to a MidoNet virtual router. These devices should have L2 and L3 connectivity to the Customer's Router, and with BGP terminated links. Virtual machine tap devices are bound to MidoNet 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 Gateway failover 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
- ZooKeeper performance recommendations:

    - Use dedicated physical disk drives for ZooKeeper snapshots
    - Use dedicated physical disk drives for ZooKeeper commit logs (SSD highly recommended)

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
- 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
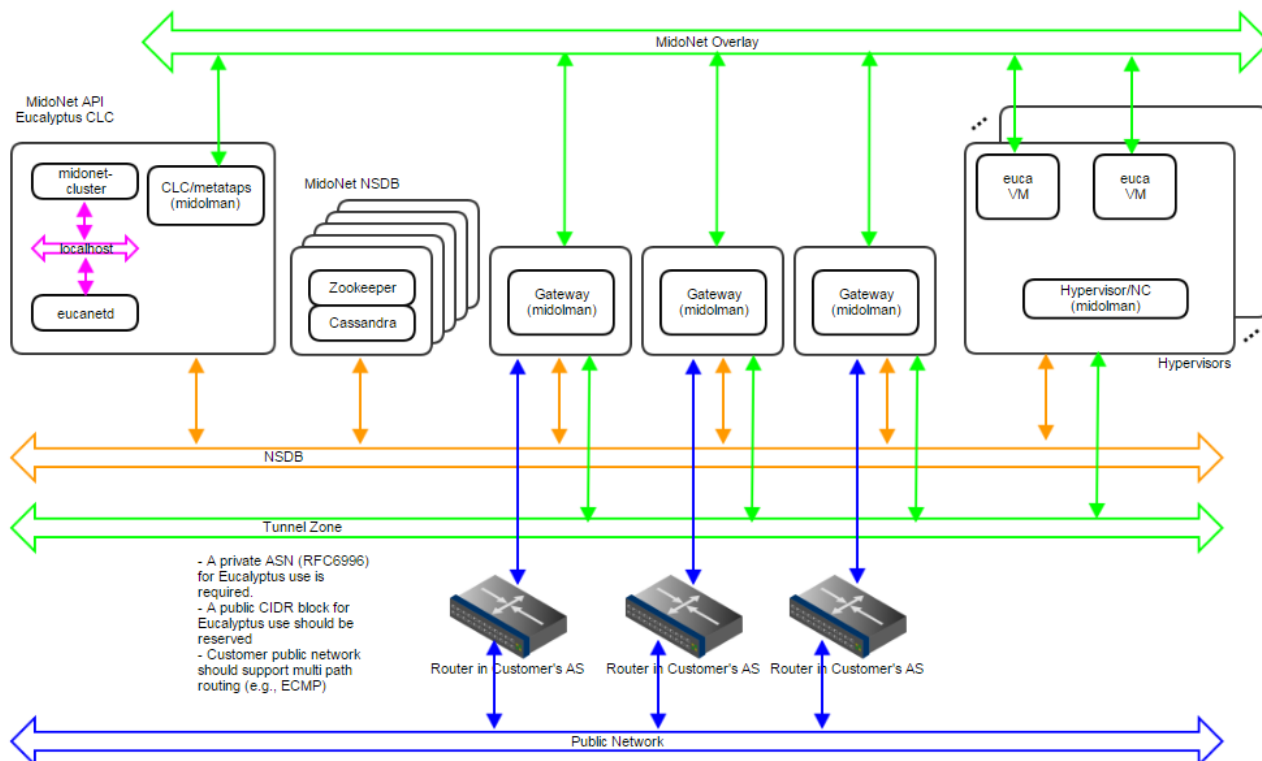
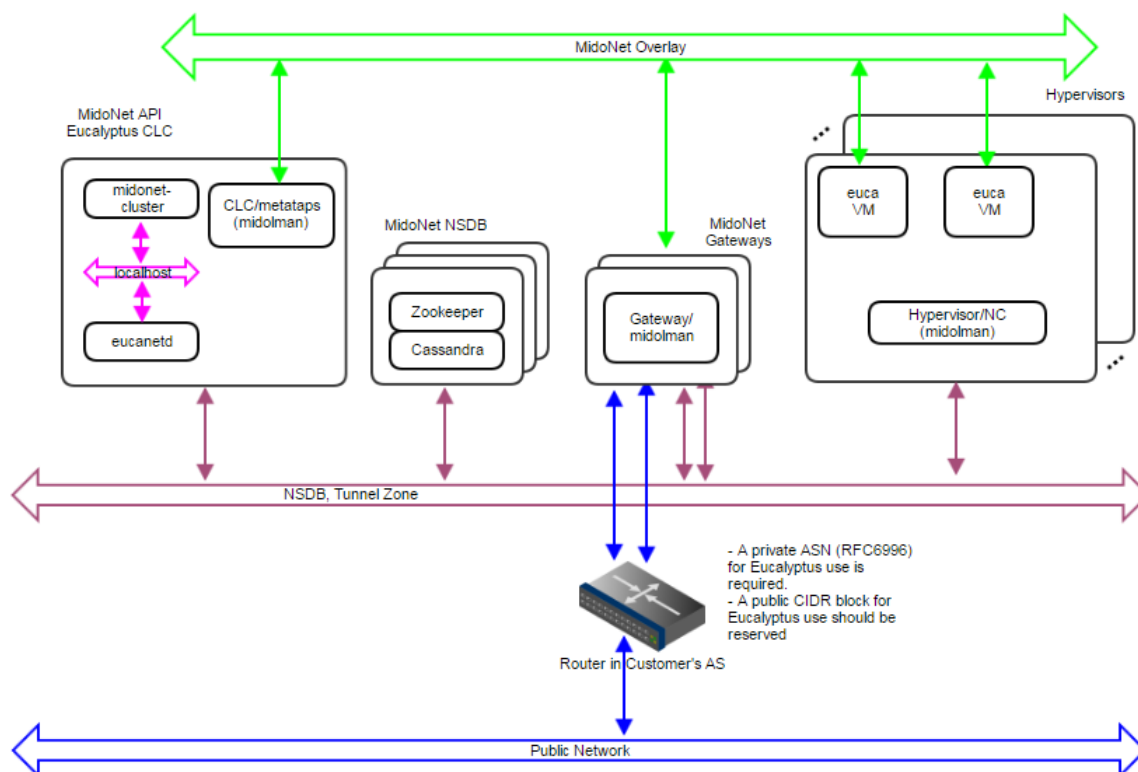*Figure 9: Production:Large deployment topology. A 1Gbps IP network carries NSDB; a 10Gbps IP network carries Tunnel Zone traffic; and another 10Gbps IP network carries Public Network traffic. A 5-node cluster for NSDB tolerates 2 server failures, and 3 gateways enable network failover and load balancing/sharing. 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 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 instance migrations
UDP 7500	Port for diagnostic probing on CLC, UFS, OSG, Walrus SC
UDP 8773	Membership port for any UFS, OSG, Walrus, and SC
UDP 8778	The bind port used to establish multicast communication
TCP/UDP 53	DNS port on UFS
UDP 63822	eucanetd binds to localhost port 63822 and uses it to detect and avoid running multiple instances (of eucanetd)

Note: For information about ports used by MidoNet, see the MidoNet Operations Guide (Category OpenStack can be ignored).

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 port 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 239.193.7.3 between CLC and UFS, OSG, Walrus, 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 in DAS or Overlay modes, verify connection from NC to SC on TCP port 3260

Configure Dependencies

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

Configure Bridges

For EDGE networking mode, 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 on CentOS 7 or RHEL 7, 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 7 and RHEL 7. 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
```

Note: The device name may vary. See the RedHat Networking Guide, Consistent Network Device Naming.

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).

Important: Choose names and use them consistently for all NCs (both the file name `ifcfg-br0` and the `DEVICE=br0` in the file).

- 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:

```
systemctl restart network.service
```

Disable FirewallD on RHEL 7

This topic describes how to stop and disable FirewallD on RHEL 7.

Prerequisites

- You should have successfully installed RHEL 7 before this task.

If you have existing firewall rules on your host machines, you must disable the firewall in order to install Eucalyptus. You should re-enable it after installation.

Note: The firewall on a RHEL 7 system is enabled by default. Before you restart the CLC, you must disable the firewalld service on all host machines.

For more information, see FirewallD on RHEL 7 or FirewallD on CentOS.

To stop and disable FirewallD

1. Check the status of the firewalld service:

```
systemctl status firewalld.service
```

The status displays as `active (running)` or `inactive (dead)`.

2. If the firewall is active / running, enter this command to stop it:

```
systemctl stop firewalld.service
```

3. To completely disable the firewalld service, so it does not reload when you restart the host machine:

```
systemctl disable firewalld.service
```

4. Verify the status of the firewalld service:

```
systemctl status firewalld.service
```

The status should display as `disabled` and `inactive (dead)`.

```
firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled;
vendor preset: enabled)
  Active: inactive (dead)
```

5. Repeat these steps for all host machines.

The firewalld service is stopped and disabled. You can now start the CLC and other host machines.

Postrequisites

- You should re-enable the firewall after installation is complete.

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.

```
systemctl enable ntpd.service
```

6. Start NTP.

```
systemctl start ntpd.service
```

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 Java

Eucalyptus Java services include the CLC, UFS, OSG, Walrus, and SC.

For the supported version of the Java Virtual Machine (JVM), see the Compatibility Matrix in the Release Notes.

As of Eucalyptus 4.3, JVM 8 is required. Eucalyptus RPM packages require java-1.8.0-openjdk, which will be installed automatically.

Note: If your network mode is VPCMIDO, MidoNet will install JVM 1.7 as a dependency (it is acceptable to have both JVM 1.7 and JVM 1.8 installed).

To use Java with Eucalyptus cloud:

1. Open the `/etc/eucalyptus/eucalyptus.conf` file.

2. Verify that the CLOUD_OPTS setting does not set `--java-home`, or that `--java-home` points to a supported JVM version.

   Note: Although it is possible to set `--java-home`, we do not recommend it unless there is a specific reason to do so.

3. If you are upgrading to Eucalyptus 4.3, note that Java 8 does not have permanent generation memory. Remove any JAVA_OPTS `MaxPermSize` settings at upgrade time.

4. Save and close the file.

5. 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`.

Install Midokura Enterprise MidoNet

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

Note: If you are not using VPC with Eucalyptus, you do not need to install Midokura Enterprise MidoNet. See Configure EDGE Network Mode.

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 Midokura Enterprise MidoNet installation procedure and concepts.

Prerequisites

This topic discusses the prerequisites for installing Midokura Enterprise MidoNet.

Important: See the Compatibility Matrix in the Release Notes for supported versions.

You need to configure software repositories and install Network State Database (NSDB) services: ZooKeeper and Cassandra.

Repository Access

In order to use MidoNet with Eucalyptus you need access credentials to the Midokura repositories. Contact Midokura Sales for access.

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

```
[midokura]
name=MidoRepo
baseurl=http://USERNAME:PASSWORD@repo.midokura.com/mem-5.2/stable/el7/
enabled=1
gpgcheck=1
gpgkey=https://repo.midokura.com/midorepo.key
```

See MidoNet Repository Configuration.

ZooKeeper

MidoNet uses Apache ZooKeeper to store critical path data about the virtual and physical network topology.

For a simple single-server installation, install ZooKeeper on any server that is IP accessible from all Midolman agents (for example: on the CLC host machine itself). You can also cluster ZooKeeper for fault tolerance. See MidoNet NSDB ZooKeeper Installation.

Enable and start the ZooKeeper service before installing the other MidoNet services.

Cassandra

MidoNet uses Apache Cassandra to store flow state information.

For a simple single-server installation, install Cassandra on any server that is IP accessible from all Midolman agents (for example: on the CLC host machine itself). You can also cluster Cassandra for fault tolerance. See MidoNet NSDB Cassandra Installation.

Enable and start the Cassandra service before installing the other MidoNet services.

MidoNet Component Topology
This topic lists topology recommendations for installing Midokura Enterprise MidoNet.

Note: See Understanding VPCMIDO and MidoNet for more information on MidoNet.

* The midonet-api must run co-located with the Eucalyptus Cloud Controller (CLC).
* Each Node Controller (NC) must run a Midolman agent.
* The Cloud Controller (CLC) must run a Midolman agent.
* It is recommended that your User Facing Services (UFS) host be used as the MidoNet Gateway (i.e., running a Midolman agent) when configuring Eucalyptus.
* The network interface(s) specified as `ExternalDevice*` (in the `network.json` configuration file) should be dedicated for Eucalyptus/MidoNet (for configuration/operation/use).
* Eucalyptus/MidoNet expects exclusive use of the network interface specified in `ExternalDevice`.
* If the main network interface of a server is specified in `ExternalDevice`, most likely the connectivity to that server will be lost once Eucalyptus is deployed.

    * If you have only one 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 (for Eucalyptus installs, you can skip step 6, "Create a port on the MidoNet Provider Router..." in the Midokura Static Setup instructions).

* In Eucalyptus 4.4, `ExternalDevice` replaces `GatewayInterface` in 4.3 and prior releases.

Eucalyptus 4.3 Network JSON Example

Note: This Eucalyptus 4.3 version of the VPCMIDO `network.json` file is still accepted, however we strongly encourage use of the updated Eucalyptus 4.4 template instead. See Configure VPCMIDO Network Mode.

The following Eucalyptus 4.3 network JSON template file shows a sample VPCMIDO mode configuration:

```
{
  "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 MidoNet for Eucalyptus
This topic shows how to install Midokura Enterprise MidoNet for use in your Eucalyptus cloud.
Install the MidoNet Cluster on the Cloud Controller (CLC)
This topic describes how to install the MidoNet Cluster.

MidoNet Cluster services provide a means to manage MidoNet functions that MidoNet agents (Midolman) are unable to perform on their own. MidoNet Cluster services include state synchronization of VxLAN gateways and the MidoNet REST API. A MidoNet v5 deployment requires at least one MidoNet cluster node, and since it replaces the MidoNet API webapp (which was hosted by tomcat in MidoNet v1 series), it must be co-located on the CLC host machine in Eucalyptus deployments.

For security reasons, the MidoNet REST API is accessed only on the CLC (localhost interface).

To install the MidoNet Cluster on the CLC

1. Add the MidoNet repo file as described in <u>Prerequisites</u>.
2. Install MidoNet Cluster packages.

```
yum install midonet-cluster python-midonetclient
```

3. Edit the `/etc/midonet/midonet.conf` file to set the ZooKeeper host IP(s). Replace ZOOKEEPER_HOST_IP in the following example:

```
[zookeeper]
zookeeper_hosts = ZOOKEEPER_HOST_IP:2181
```

4. Configure cloud-wide access to the NSDB services:

```
cat << EOF | mn-conf set -t default
zookeeper {
  zookeeper_hosts = "ZOOKEEPER_HOST:2181"
}

cassandra {
  servers = "CASSANDRA_HOST"
}
EOF
```

5. Enable and start the MidoNet Cluster:

```
systemctl enable midonet-cluster.service
systemctl start midonet-cluster.service
```

6. Set the midonet-api end point:

```
mn-conf set cluster.rest_api.http_port=8080
mn-conf set cluster.rest_api.http_host="127.0.0.1"
```

7. Restart the Midonet Cluster so the rest_api parameters take effect:

```
systemctl restart midonet-cluster.service
```

Install Midolman on Eucalyptus components

This topic describes how to install the Midolman agent.

Midolman is the MidoNet Agent, which is a daemon that runs on all hosts where traffic enters and leaves MidoNet. The Midolman agent is required on the Cloud Controller (CLC), Node Controllers (NCs), and any host that is a MidoNet Gateway node (e.g., UFS).

To install Midolman agent

1. Edit the `/etc/midolman/midolman.conf` file to set the ZooKeeper host IP(s). Replace ZOOKEEPER_HOST_IP in the following example:

```
[zookeeper]
zookeeper_hosts = ZOOKEEPER_HOST_IP:2181
```

2. Enable and start Midolman:

```
systemctl enable midolman.service
systemctl start midolman.service
```

3. Configure a Midolman resource usage template.

   For large Eucalyptus clouds, use the agent-compute-large template. For standard (small or medium) Eucalyptus clouds, use the default template. For gateway nodes, use the agent-gateway templates.

   Tip: For production environments, large templates are recommended.

   See the <u>Midolman Installation documentation</u> for more information.

   a) Choose the Midolman resource usage template name, based on the size and type of installation:

   ```
   agent-compute-large
   ```

```
agent-compute-medium
agent-gateway-large
agent-gateway-medium
default
```

b) Run this command, replacing TEMPLATE_NAME with your chosen template:

```
mn-conf template-set -h local -t TEMPLATE_NAME
```

Create a tunnel zone in MidoNet and add hosts

This topic describes how to create a MidoNet tunnel zone.

In MidoNet, a tunnel zone is an isolation zone for hosts. Physical hosts that are members of a given tunnel zone communicate directly with one another and establish network tunnels as needed, and on demand. These network tunnels are used to transport overlay traffic (e.g., inter-VM communication) and isolate the underlay physical network communication (i.e., inter-physical hosts communication). On a Eucalyptus deployment, one MidoNet tunnel zone is expected with the IP address on the physical network designated to carry VM traffic being used when configuring its members. Eucalyptus accepts the following tunnel zone names:

- eucatz
- euca-tz
- midotz
- mido-tz

For more information, see What are Tunnel Zones?

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 Service 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.

Additional ZooKeeper Configuration

Ongoing data directory cleanup is required for ZooKeeper.

The following parameters should be added in /etc/zookeeper/zoo.cfg for automatic purging of the snapshots and corresponding transaction logs:

```
autopurge.snapRetainCount=3  # The number of snapshots to retain in dataDir
autopurge.purgeInterval=1  # Purge task interval in hours
```

For more information, see ZooKeeper Admin Guide, Ongoing Data Directory Cleanup.

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

The following *terminology* might help you as you proceed through this section.

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. (Everybody receives 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.

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 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 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 Eucalyptus products released after July 2011 and their updates.

- Download from 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 Eucalyptus pre-release products due for release after July 2011.

- Download from 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 Eucalyptus products published after July 2011.

*   Download from 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.

Prerequisites

*   The prerequisite hardware and software should be in place and available to Eucalyptus.

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.cloud/software/eucalyptus/4.4/
    rhel/7/x86_64/eucalyptus-release-4.4-2.8.as.el7.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-4.4-1.el7.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.cloud/software/euca2ools/3.4/rhel/7/
    x86_64/euca2ools-release-3.4-2.2.as.el7.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:

    For RHEL/CentOS 7.3

    ```
    yum install http://dl.fedoraproject.org/pub/epel/epel-release-
    latest-7.noarch.rpm
    ```

    Enter `y` when prompted to install this package.

    For RHEL/CentOS 7.4 and higher

    ```
    yum install epel-release
    ```

    Enter `y` when prompted to install this package.

5. If you are installing on RHEL 7, 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** check-box is selected.
    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-node
    ```

    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. 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.

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

```
yum install eucalyptus-cloud
```

10. (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.

11. 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-cluster eucalyptus-sc eucalyptus-walrus
```

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

12. (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
```

13. (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.

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 SELinux

Security-Enhanced Linux (SELinux) is a security feature for Linux that allows you to set access control through policies. Eucalyptus 4.4 packages automatically install SELinux policy for Eucalyptus on RHEL 7 and CentOS 7.

We recommend enabling SELinux on host systems running Eucalyptus 4.4 services to improve their security on RHEL 7. Enabling SELinux, as described in this topic, can help contain break-ins. For more information, see RedHat SELinux documentation.

You need to set boolean values on Storage Controller (SC) and Management Console host machines. If your network mode is VPCMIDO, you also set a boolean value on the Cloud Controller (CLC) host machines.

To configure SELinux on Eucalyptus 4.4:

1. On each Storage Controller (SC) host machine, run the following command:

   ```
   setsebool -P eucalyptus_storage_controller 1
   ```

   This allows Eucalyptus to manage EBS volumes.

2. On each Management Console host machine, run the following command:

   ```
   setsebool -P httpd_can_network_connect 1
   ```

   This allows the Management Console's HTTP proxy to access the back end.

   Tip: If you can't access the console after starting it, this KB article might help: Cannot access eucaconsole when SELinux set to enforcing.

3. If your cloud uses VPCMIDO networking mode, on the Cloud Controller (CLC), run the following command:

   ```
   setsebool -P httpd_can_network_connect 1
   ```

   This allows the CLC's HTTP proxy to access the back end.

SELinux is now configured and ready to use with your Eucalyptus 4.4 cloud.

Configure Network Modes

This section provides configuration instructions for Eucalyptus networking modes.

Eucalyptus overlays a virtual network on top of your existing network. In order to do this, Eucalyptus supports these networking modes: EDGE (AWS EC2 Classic compatible) and VPCMIDO (AWS VPC compatible).

Configure EDGE Network Mode

This topic provides configuration instructions for Eucalyptus EDGE network mode. Eucalyptus requires network connectivity between its clients (end-users) and the cloud components (e.g., CC, CLC, and Walrus).

Note: If you are not using EDGE mode with Eucalyptus, you can skip this topic. See Configure VPCMIDO Network Mode.

To configure Eucalyptus for EDGE mode, most networking configuration is handled through settings in a global Cloud Controller (CLC) property file.

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

Option	Description	Component
VNET_BRIDGE	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`.	Node Controller
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`.	Node Controller
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`	All CCs and NCs
VNET_PRIVINTERFACE	The name of the network interface that is on the same network as the NCs. Default: `eth0`	Node Controller
VNET_PUBINTERFACE	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`	Node Controller

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 Network JSON File

To configure the rest of the EDGE mode parameters, you must create a `network.json` configuration file. Later in the installation process you will Upload the Network JSON Configuration File to the CLC.

1. Create the network 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 `network.json` 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"
                        ]
                    }
                ]
}
```

Configure VPCMIDO Network Mode

This topic provides configuration instructions for Eucalyptus VPCMIDO network mode. Eucalyptus requires network connectivity between its clients (end-users) and the cloud components (e.g., CC, CLC, and storage).

Note: If you are not using VPCMIDO mode with Eucalyptus, you can skip this topic. See Configure EDGE Network Mode.

To configure VPCMIDO mode parameters, you must create a `network.json` configuration file. Later in the installation process you will Upload the Network JSON Configuration File to the CLC.

1. Create the network JSON file.

   a) Open a text editor.
   b) Create a file similar to the following structure. This example demonstrates two gateways and two BGP peers (sections relevant to VPCMIDO are shown here).

```
{
    "Mido": {
        "BgpAsn": "64512",
        "Gateways": [
            {
                "Ip": "10.111.5.11",
                "ExternalDevice": "em1.116",
                "ExternalCidr": "10.116.128.0/17",
                "ExternalIp": "10.116.133.11",
                "BgpPeerIp": "10.116.133.173",
                "BgpPeerAsn": "65000",
                "BgpAdRoutes": [
                    "10.116.150.0/24"
                ]
            },
            {
                "Ip": "10.111.5.22",
                "ExternalDevice": "em1.117",
                "ExternalCidr": "10.117.128.0/17",
                "ExternalIp": "10.117.133.22",
                "BgpPeerIp": "10.117.133.173",
                "BgpPeerAsn": "65001",
                "BgpAdRoutes": [
                    "10.117.150.0/24"
                ]
            }
        ],
        "Mode": "VPCMIDO",
        "PublicIps": [
            "10.116.150.10-10.116.150.254",
            "10.117.150.10-10.117.150.254"
        ]
    }
}
```

2. Save the `network.json` file.

The following example demonstrates a gateway with static routing configuration. This configuration might be used for a Proof of Concept (POC) environment; however, it is not recommended for production.

```
{
    "Mido": {
        "Gateways": [
            {
                "Ip": "10.111.5.11",
                "ExternalDevice": "em1.116",
                "ExternalCidr": "10.116.128.0/17",
                "ExternalIp": "10.116.133.11",
                "ExternalRouterIp": "10.116.133.173"
            }
        ]
    },
    "Mode": "VPCMIDO",
    "PublicIps": [
        "10.116.150.10-10.116.150.254"
    ]
}
```

VPCMIDO Gateway Configuration Parameters

This topic provides detailed configuration parameter information for Eucalyptus VPCMIDO network mode.

Eucalyptus VPCMIDO Gateway Configuration

The following table provides a list of VPCMIDO parameters.

Parameter	Description	Validation
BgpAsn	(Optional) *Global* BGP configuration * BGP Autonomous System Number assigned (to be decided by Eucalyptus administrator/installer) for this Eucalyptus VPCMIDO deployment. Private ASN range should be used: • 16-bit: 64512 - 65534 • 32-bit: 131072 - 4199999999 (RFC6996)	• Private use blocks recommended, but owners of public ASNs can use public ASNs or other blocks if they wish. • Valid range is 1 - 4294967295.
Gateways	(The VPCMIDO gateway parameters are below.)	Per MidoNet/BGP limitation, a maximum of 6 MidoGateways can be used.
Ip	Internal IP address of Mido Gateway (not to be confused with the IP address of the gateway interface used in external communications). **Note**: Replaces 4.3 GatewayHost parameter.	• Must be a valid IP address. • Must be a live IP address configured on the machine.

Parameter	Description	Validation
ExternalDevice	Device name of Mido Gateway interface that is physically connected to the external network (i.e., has L2 connectivity to the infrastructure router or BGP peer). This interface is dedicated for MidoNet use (Mido Gateway Operating System should not have control of this device). **Note**: Replaces 4.3 *GatewayInterface* parameter.	Must be a valid network interface connected to the network where L2 communication with BgpPeerIp (or ExternalRouterIp) can be established.
ExternalCidr	CIDR block used in the external routing. **Note**: Replaces 4.3 PublicNetworkCidr parameter.	Must be a valid CIDR block.
ExternalIp	IP address to be configured on ExternalDevice by eucanetd. Its subnet is as specified in ExternalCidr (ExternalCidr must contain ExternalIp). **Note**: Replaces 4.3 GatewayIP parameter.	• Must be a valid and unused IP address. • Must be within ExternalCidr. • Must not be a network or broadcast address.
ExternalRouterIp	IP address of an external router (for static configuration). This is the router IP address used in default routes for traffic originating from MidoNet. **Note**: Partially replaces 4.3 PublicGatewayIp parameter.	• Must be a valid and unused IP address. • Must be within ExternalCidr. • Must not be a network or broadcast address. • Either ExternalRouterIp or BgpPeerIp is required.
BgpPeerIp	(Optional) BGP configuration * IP address of a BGP peer. This is the IP address to where MidoNet router will attempt to establish a BGP session. **Note**: Partially replaces 4.3 PublicGatewayIp parameter.	• Must be a valid and unused IP address. • Must be within ExternalCidr. • Must not be a network or broadcast address. • Either ExternalRouterIp or BgpPeerIp is required.
BgpPeerAsn	(Optional) BGP configuration * BGP peer ASN for this MidoGateway.	Valid range is 1 - 4294967295.

Parameter	Description	Validation
BgpAdRoutes	(Optional) BGP configuration * A list of CIDR blocks delegated to this Eucalyptus VPCMIDO deployment. VPCMIDO BGP will be configured to advertise these routes. Eucalyptus public IPs must be within these CIDR blocks. The same list can be used for all MidoGateways. The advantage of having a separate list per MidoGateway is that it allows different MidoGateways to be responsible for different CIDR blocks. If the same list of CIDR blocks is used for all MidoGateways, MidoNet built-in load sharing/balancing mechanism is used.	Each entry must be a valid CIDR block.

PublicIps	The public IP address ranges associated with VPCMIDO.	• With BGP:
		
		   • Each public IP must be within one of the CIDR blocks in the union of all BgpAdRoutes entries.
		   • Must be a valid IP address range.
		   • Must not contain network or broadcast address of the CIDR blocks in the union of all BgpAdRoutes.
		 • Without BGP: On-premise infrastructure must route all PublicIps to one of the MidoGateways.

* BGP parameters are optional but recommended for production. Static configuration can be used for a Proof of Concept (POC) environment; however, it is not recommended for production.

Gateways with BGP require `BgpPeerAsn`, `BgpAdRoutes`, and `BgpAsn`. If all gateways are static (no BGP), `BgpAsn` is optional. A gateway with BGP has `BgpPeerAsn` and `BgpAdRoutes` parameters; a static gateway does not.

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

Option	Description
	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.

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.

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: If you are upgrading and you just restored your cloud data, do not initialize the CLC; skip this step.

Important: Make sure that the `eucalyptus-cloud` process is *not* running before 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:

```
systemctl enable eucalyptus-cloud.service
```

4. Enter the following command to start the CLC:

```
systemctl start eucalyptus-cloud.service
```

5. If you are running in VPCMIDO networking mode:

a) If you want the eucanetd service to start at each boot-time, run this command:

```
systemctl enable eucanetd.service
```

b) Start the eucanetd service:

```
systemctl start eucanetd.service
```

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:

```
systemctl enable eucalyptus-cloud.service
```
3. Enter the following command to start the UFS:
```
systemctl start eucalyptus-cloud.service
```
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:
```
systemctl enable eucalyptus-cloud.service
```
2. Log in to the Walrus Backend host machine and enter the following command:
```
systemctl start eucalyptus-cloud.service
```

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:
```
systemctl enable eucalyptus-cluster.service
```
3. Enter the following command to start the CC:
```
systemctl start eucalyptus-cluster.service
```
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.

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

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:
```
systemctl enable eucalyptus-cloud.service
```
3. If you want the tgtd service to start at each boot-time, run this command:
```
systemctl enable tgtd.service
```
Note: Eucalyptus depends on tgtd to create and manage Eucalyptus block storage volumes when the storage provider is either DAS or Overlay.

4. Enter the following commands to start the SC:

```
systemctl start tgtd.service
```

```
systemctl start eucalyptus-cloud.service
```

5. 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:

```
systemctl enable eucalyptus-node.service
```

3. Enter the following command to start the NC:

```
systemctl start eucalyptus-node.service
```

4. If you are running in EDGE networking mode:

a) If you want the eucanetd service to start at each boot-time, run this command:

```
systemctl enable eucanetd.service
```

b) Start the eucanetd service:

```
systemctl start eucanetd.service
```

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 multiple Management Console host machines, we recommend turning off the default memcached in your console.ini file. See Configure Memcached for details.

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:

```
systemctl enable eucaconsole.service
```

3. Enter the following command to start the console:

```
systemctl start eucaconsole.service
```

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 port 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.

Note: If you are upgrading, proceed to the Register the Node Controllers section. (You don't need to register the rest (e.g., UFS, Walrus, etc.) during the non-NC upgrade, because those registrations are already listed in the cloud database, which you recovered before getting here.)

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 shorthand 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.
- If you are upgrading, you should understand that:

  - If you're upgrading an NC, just register that NC (on the CC that had it registered before).
  - If you're upgrading the set of non-NC host machines, register all the NCs (on each CC that had NCs registered).

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 Eucalyptus 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 `192.0.2.5`, and the IP address of Walrus is `192.0.2.6`, the host `compute.mycloud.example.com` resolves to `192.0.2.5` and `objectstorage.mycloud.example.com` resolves to `192.0.2.6`.

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`.

Optional: Configure Eucalyptus DNS to Spoof AWS Endpoints

You can configure instances to use AWS region FQDNs for service endpoints by enabling DNS spoofing.

1. Set up a Eucalyptus cloud with Eucalyptus DNS and HTTPS endpoints. When creating CSR, make sure and add Subject Alternative Names for all the supported AWS services for the given region that's being tested. For example:

```
$ openssl req -in wildcard.c-06.autoqa.qa1.eucalyptus-systems.com.csr
       -noout -text | less X509v3 Subject Alternative Name:
```

```
      DNS:ec2.us-east-1.amazonaws.com, DNS:autoscaling.us-
east-1.amazonaws.com,
      DNS:cloudformation.us-east-1.amazonaws.com, DNS:monitoring.us-
east-1.amazonaws.com,
      DNS:elasticloadbalancing.us-east-1.amazonaws.com,
 DNS:s3.amazonaws.com,
      DNS:sts.us-east-1.amazonaws.com
```

2. Set DNS spoofing:

```
[root@d-17 ~]#  euctl dns.spoof_regions --region euca-admin@future
dns.spoof_regions.enabled = true
dns.spoof_regions.region_name =
dns.spoof_regions.spoof_aws_default_regions = true
dns.spoof_regions.spoof_aws_regions = true
```

3. Launch an instance, and allow SSH access.

4. SSH into the instance and install AWS CLI.

```
ubuntu@euca-172-31-12-59:~$ sudo apt-get install -y python-pip
ubuntu@euca-172-31-12-59:~$ sudo -H pip install --upgrade pip
ubuntu@euca-172-31-12-59:~$ sudo -H pip install --upgrade awscli
```

5. Run `aws configure` and set access and secret key information if not using instance profile.

6. Confirm AWS CLI works with HTTPS Eucalyptus service endpoint:

```
ubuntu@euca-172-31-12-59:~$ aws --ca-bundle euca-ca-0.crt
--endpoint-url https://ec2.c-06.autoqa.qa1.eucalyptus-systems.com/ ec2
 describe-key-pairs
{
    "KeyPairs": [
        {
            "KeyName": "devops-admin",
            "KeyFingerprint":
 "ee:4f:93:a8:87:8d:80:8d:2c:d6:d5:60:20:a3:2d:b2"
        }
    ]
}
```

7. Test against AWS FQDN service endpoint that matches one of the SANs in the signed certificate:

```
ubuntu@euca-172-31-12-59:~$ aws --ca-bundle euca-ca-0.crt
--endpoint-url https://ec2.us-east-1.amazonaws.com ec2 describe-key-pairs{
    "KeyPairs": [
        {
            "KeyName": "devops-admin",
            "KeyFingerprint":
 "ee:4f:93:a8:87:8d:80:8d:2c:d6:d5:60:20:a3:2d:b2"
        }
    ]
}
```

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.

Tip: This is where you would begin using the admin role, if you want to use that feature.

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 Eucalyptus 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.

Note: This creates an admin user in the built-in 'eucalyptus' account. The admin user has full control of all aspects of the cloud. For additional security, you might instead want to create a new account and grant it access to a more limited administration role. See the Identity and Access Management (IAM) Guide for more information.

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 Eucalyptus DNS.

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

What to do next

The remainder of this guide assumes you have completed the above steps.

Use these credentials after this point.

Upload the Network JSON Configuration File

This topic describes how to upload the JSON network configuration file you created 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/
network_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.

Note: Ceph RGW is a Controlled Availability feature. For information about Controlled Availability features in Eucalyptus, see Special Status Release Features.

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 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.

Note: Ceph RGW is a Controlled Availability feature. For information about Controlled Availability features in Eucalyptus, see Special Status Release Features.

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.

Note: Storage SANs are deprecated as of Eucalyptus 4.4; to be removed in a future release.

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 to 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; this backend provider includes snapshot delta support

Note: EBS snapshot delta support is a Technical Preview feature. For information about Technical Preview features in Eucalyptus, see Special Status Release Features.

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 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. If you want to enable snapshot deltas for your Ceph backend:

   Note: Snapshot deltas are supported only on Ceph-RBD.

   Note: EBS snapshot delta support is a Technical Preview feature. For information about Technical Preview features in Eucalyptus, see Special Status Release Features.

   a) Verify that snapshots are enabled:

   ```
   euctl ZONE.storage.shouldtransfersnapshots=true
   ```

   b) Set the maximum number of deltas to be created before creating a new full snapshot:

   ```
   euctl ZONE.storage.maxsnapshotdeltas=NON_ZERO_INTEGER
   ```

   Note: This variable applies to all Ceph volumes.

   Tip: If you need to create a non-Ceph volume from a Ceph snapshot, this value would need to be set to zero (at least temporarily).

7. 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-node` service is running, terminate/stop all instances. After all instances are terminated, stop the eucalyptus-node service.

```
systemctl stop eucalyptus-node.service
```

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-ev and qemu-img-ev 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. If you are *not* using the RHEV package procedure to install the `qemu-kvm-ev` and `qemu-img-ev` packages, install Eucalyptus-built RHEV packages: `qemu-kvm-ev` and `qemu-img-ev`, which can be found in the same yum repository as other Eucalyptus packages.

```
yum install qemu-kvm-ev qemu-img-ev
```

6. Start the `libvirtd` service.

```
systemctl start libvirtd.service
```

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-node service is started.

```
systemctl start eucalyptus-node.service
```

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).

Note: Storage SANs are deprecated as of Eucalyptus 4.4; to be removed in a future release.

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.
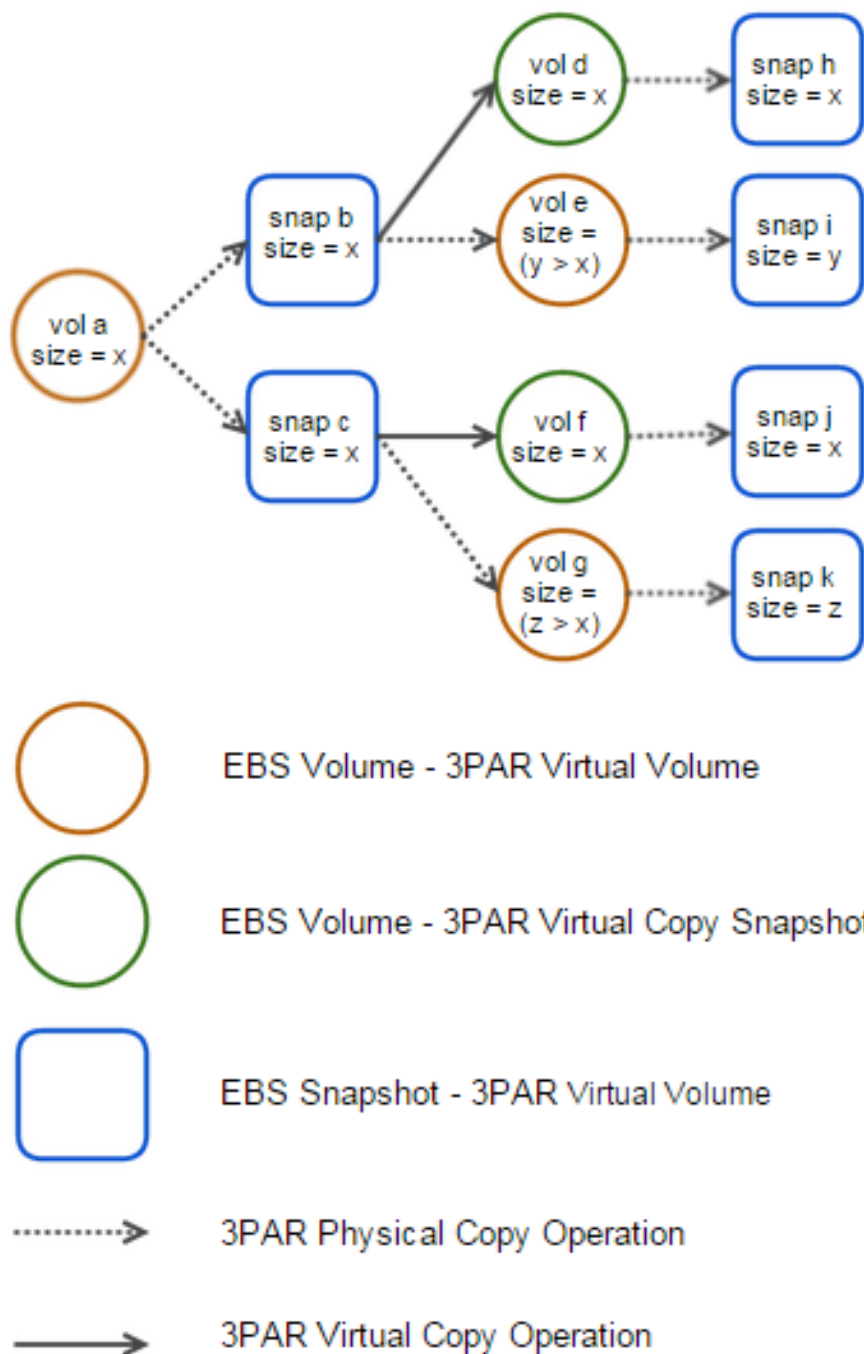
Snapshot to volume optimization

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

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
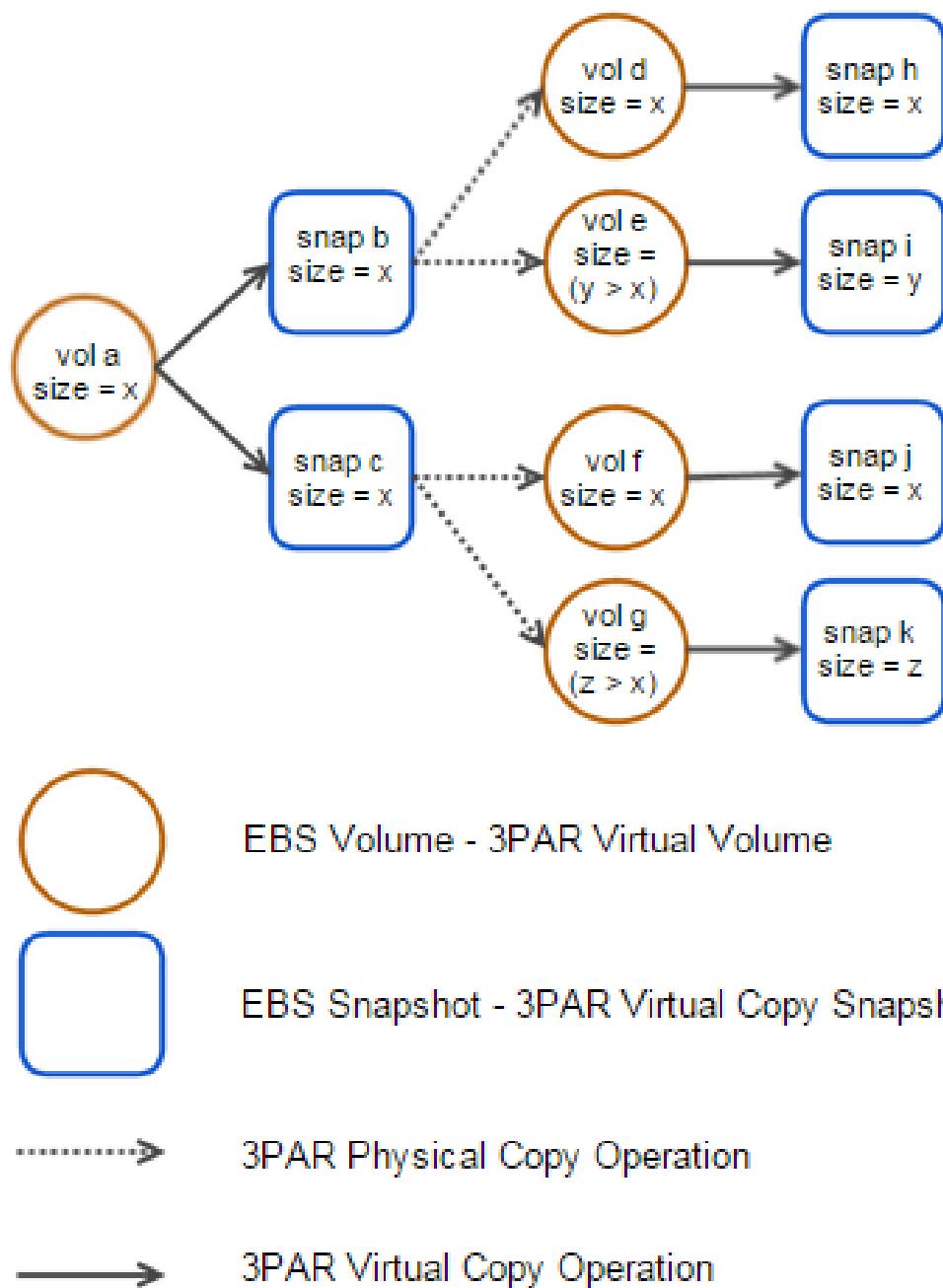
Use a NetApp SAN

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

Note: Storage SANs are deprecated as of Eucalyptus 4.4; to be removed in a future release.

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 con#gured.

    ```
    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 con#gured.

```
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 con#guring 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).

Note: Storage SANs are deprecated as of Eucalyptus 4.4; to be removed in a future release.

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 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.

6. (Optional) These properties are available for EqualLogic thin provisioning, which can save significant space on your SAN.

These properties take effect only during new EBS volume / snapshot creation. EBS volumes created from previously existing snapshots inherit the provisioning settings of the parent.

```
euctl -s ZONE.storage.eqlthin
ZONE.storage.eqlthinprovision = Thin provision space on Equallogic SAN.
 Value must be true to enable and false to disable thin provisioning
ZONE.storage.eqlthinminreserve = Minimum volume reserve for a thin-
provisioned volume expressed as a % of the volume size. This is the amount
 of immediately writable, guaranteed space reserved for the volume
```

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.

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.

   ```
   systemctl stop eucalyptus-cloud.service
   ```
3. Edit all the config files on NC and CC for Edge networking mode. For more information, see Configure EDGE Network Mode.
4. Install eucanetd on all NCs.

   ```
   yum install eucanetd
   ```
5. Start eucanetd on all NCs

   ```
   systemctl start eucanetd.service
   ```
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 Network JSON File.

Your Edge networking mode is now properly configured.

Eucalyptus Upgrade

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

You can upgrade to Eucalyptus 4.4.5 from 4.4.4 (or 4.3.1.1). If your current version is earlier than 4.4.4, see the prescribed paths below. Follow the directions in that version's Installation Guide in the documentation archive, and then upgrade to 4.4.5 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.

Prescribed upgrade paths

The following are the prescribed upgrade paths for Eucalyptus versions prior to 4.4:

* Upgrade from 3.1.2 to 3.2.2
* Upgrade from 3.2.2 to 3.3.2
* Upgrade from 3.3.2 to 3.4.3
* Upgrade from 3.4.3 to 4.0.2
* Upgrade from 4.0.2 to 4.1.2
* Upgrade from 4.1.2 to 4.2.2
* Upgrade from 4.2.2 to 4.3.1 (see note below)

Note: You must have completed the upgrade to 4.3 on RHEL 7 before you can upgrade to 4.4.

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 4.4. See the Compatibility Matrix in the Release Notes for supported versions.
* Verified the health of your current Eucalyptus deployment, as described in Inspect System Health in the Administration Guide.
* Followed the prescribed upgrade paths 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. See also Back Up Eucalyptus Cloud Data in the Administration Guide.
* 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 4.4.5 and a Walrus at 4.4.4.
* 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-4.4.5) 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:

* Perform the upgrade steps on *every* Eucalyptus host machine, as described in this section.

- We recommend that you also install the new version of the Management Console, although this is not required. If you don't install the new version, you will not be able to use new features from the console.
- We recommend that you install the current 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 preview 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.cloud/software/eucalyptus/4.4/
rhel/7/x86_64/eucalyptus-release-4.4-2.8.as.el7.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.cloud/software/euca2ools/3.4/rhel/7/
x86_64/euca2ools-release-3.4-2.2.as.el7.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-4.4-1.el7.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:

```
systemctl stop eucalyptus-cloud.service
```

2. If you have separate SC host machines, log in to each host and shut down the SC services:

```
systemctl stop eucalyptus-cloud.service
```

3. If you have a separate Walrus host machine, log in and shut down the Walrus backend services:

```
systemctl stop eucalyptus-cloud.service
```

4. If you have separate UFS host machines, log in to each host and shut down the UFS services:

```
systemctl stop eucalyptus-cloud.service
```

5. 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:

```
systemctl stop eucalyptus-cloud.service
```

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

```
systemctl stop eucalyptus-cluster.service
```

7. (Optional) If you are not sure which hosts are *running* eucanetd, check the status on each CLC, CC, and NC:

```
systemctl status eucanetd.service
```

   Important: Depending on your network configuration, eucanetd might be running on any of these hosts. Also note that it can be *installed* and not *running*. See About Eucanetd for more information.

   Make note of all hosts that have eucanetd *running*; you'll need this when you start up the services again.

8. Log in to each host machine running eucanetd and shut it down:

```
systemctl stop eucanetd.service
```

9. Log in to each NC host machine and shut down the NC service:

```
systemctl stop eucalyptus-node.service
```

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

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

```
systemctl stop eucaconsole.service
```

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 4.4.5. 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.cloud/software/euca2ools/3.4/rhel/7/
x86_64/euca2ools-release-3.4-2.2.as.el7.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 expire-cache
```

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 4.4.5.

Tip: It's a good idea to upgrade all the non-NC's before the NCs.

To upgrade Eucalyptus

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

```
yum install http://downloads.eucalyptus.cloud/software/eucalyptus/4.4/
rhel/7/x86_64/eucalyptus-release-4.4-2.8.as.el7.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-4.4-1.el7.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 203.0.113.$host;
ssh 203.0.113.$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-ev
```

7. Review the Java dependency for any changes you might need to make after upgrading Eucalyptus packages.

Java 8 is required as of Eucalyptus 4.3.0; previous releases of Eucalyptus used Java 7.

See Configure Java.

You are now ready to Start Eucalyptus Services.

Start Eucalyptus Services

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

Prerequisites

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

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

To start Eucalyptus services after the upgrade

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

   ```
   systemctl start eucalyptus-cloud.service
   ```

2. If you have separate SC host machines, log in to each host and start the services:

   ```
   systemctl start eucalyptus-cloud.service
   ```

3. If you have a separate Walrus host machine, log in and start the services:

   ```
   systemctl start eucalyptus-cloud.service
   ```

4. If you have separate UFS host machines, log in to each host and start the services:

   ```
   systemctl start eucalyptus-cloud.service
   ```

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

   ```
   systemctl start eucalyptus-cloud.service
   ```

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

   ```
   systemctl start eucalyptus-cluster.service
   ```

7. Log in to each NC server and start the service:

   ```
   systemctl start eucalyptus-node.service
   ```

8. Log in to each host machine that was running eucanetd prior to upgrade and start the service again:

   Important: Depending on your network configuration, eucanetd might have been running on any of these hosts: CLC, CC, NC. Also note that it might have been *installed* and not *running*. See About Eucanetd for more information.

   ```
   systemctl start eucanetd.service
   ```

   Note: Use the list of *running* eucanetd hosts, which you created in Shutdown Services.

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 service is enabled.

To verify that all services are enabled:

1. Verify the Eucalyptus versions. For example:

   ```
   # euca-version
   euca2ools 3.4.1
   ```

```
eucalyptus 4.4.5
```

2. If you are using the Walrus backend for object storage, 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
```

3. 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
```

4. 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
```

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

```
euca-describe-instance-types --show-capacity --by-zone
```

The returned output should a non-zero number in the `Total` column, as in the following example.

```
AVAILABILITYZONE        test00
INSTANCETYPE Name       CPUs  Memory (MiB)  Disk (GiB)   NICs   Used / Total
  Used %
INSTANCETYPE t1.micro     1        256           5        2      0 /     4
      0%
INSTANCETYPE m1.small     1        256          10        2      0 /     4
      0%
INSTANCETYPE m1.medium    1        512          10        2      0 /     4
      0%
INSTANCETYPE c1.medium    2        512          10        2      0 /     2
      0%
INSTANCETYPE m1.large     2        512          10        3      0 /     2
      0%
INSTANCETYPE m1.xlarge    2       1024          10        4      0 /     2
      0%
INSTANCETYPE c1.xlarge    2       2048          10        4      0 /     2
      0%
AVAILABILITYZONE        test01
INSTANCETYPE Name       CPUs  Memory (MiB)  Disk (GiB)   NICs   Used / Total
  Used %
INSTANCETYPE t1.micro     1        256           5        2      0 /     4
      0%
INSTANCETYPE m1.small     1        256          10        2      0 /     4
      0%
INSTANCETYPE m1.medium    1        512          10        2      0 /     4
      0%
INSTANCETYPE c1.medium    2        512          10        2      0 /     2
      0%
INSTANCETYPE m1.large     2        512          10        3      0 /     2
      0%
INSTANCETYPE m1.xlarge    2       1024          10        4      0 /     2
      0%
INSTANCETYPE c1.xlarge    2       2048          10        4      0 /     2
      0%
```

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.

• 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
```

Note: If you use the `--region` parameter on this command, do not specify localhost (as in user@localhost); the command will fail with an error. Use the region name as shown in a Euca2ools configuration file such as `/root/.euca/euca-admin.ini`. For example, if these lines are shown in that file:

```
region machine1.example.com
user = 000805022146:admin
```

Add the following, if you use the `--region` parameter:

```
--region 000805022146:admin@machine1.example.com
```

Your Eucalyptus 4.4 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.

Downgrade Overview

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. Note that downgrade instructions are different, depending on whether your Eucalyptus services are co-located or each hosted 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 4.4.4 release package on each host machine:

```
yum downgrade http://downloads.eucalyptus.com/software/eucalyptus/4.4/
rhel/7/x86_64/eucalyptus-release-4.4-1.el7.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 4.4.4:

```
yum downgrade http://downloads.eucalyptus.com/software/subscription/
eucalyptus-enterprise-release-4.4-1.el7.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-node
 eucanetd
```

Note: The `eucalyptus-nc` service name changed to `eucalyptus-node` in Eucalyptus 4.3.

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-cluster
downgrade eucalyptus-cloud
downgrade eucalyptus-common-java
downgrade eucalyptus-common-java-libs
downgrade eucalyptus-sc
downgrade eucalyptus-service-image
downgrade eucalyptus-walrus
downgrade eucanetd
```

Note: The `eucalyptus-cc` service name changed to `eucalyptus-cluster` in Eucalyptus 4.3.

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
```

CC transaction commands:

```
downgrade eucalyptus
downgrade eucalyptus-admin-tools
downgrade eucalyptus-cluster
downgrade eucanetd
```

Note: The `eucalyptus-cc` service name changed to `eucalyptus-cluster` in Eucalyptus 4.3.

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 3PAR transaction commands:

```
downgrade eucalyptus-enterprise-storage-san-threepar
downgrade eucalyptus-enterprise-storage-san-threepar-libs
```

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 4.4 class file artifacts; they will be regenerated as needed for your downgraded cloud.

Downgrade Euca2ools

In general, there is no need to downgrade Euca2ools.

Whenever you install Euca2ools, it will always give you the latest patch of a release (for example, you'd get 3.4.1 over 3.4.0). If, for example, you have 3.4.1 installed, then performing the following steps would downgrade to 3.4.0.

Note: If Euca2ools is not the source of upgrade failure, there is no reason to downgrade Euca2ools.

1. Expire the cache for the yum repositories on each host machine:

```
yum clean expire-cache
```

2. Downgrade to Euca2ools 3.3.3 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 3.3.3
eucalyptus 4.4.4
```

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/4.4/rhel/7/x86_64/ \
-P /tmp/eucalyptus
```

2. Download Euca2ools:

```
wget -r --no-parent \
http://downloads.eucalyptus.com/software/euca2ools/3.4/rhel/7/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/4.4/rhel/7/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/3.4/rhel/7/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-7.noarch.rpm
```

2. Configure the Euca2ools package repository:

```
yum install http://downloads.eucalyptus.cloud/software/euca2ools/3.4/rhel/7/
x86_64/euca2ools-release-3.4-2.2.as.el7.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
Use Ceph-RBD	Added snapshot delta functionality.	March 2017
Eucalyptus Upgrade	Updated upgrade for 4.4.0 release. Removed RHEL 6 / 7 upgrade info, which applied only to the 4.3.x releases.	March 2017
Networking	Removed Managed modes.	March 2017
Networking	Updates and corrections to VPCMIDO mode.	March 2017