Eucalyptus 4.1.2 Administration Guide

2016-01-26 Eucalyptus Systems

Contents

Management Overview

The section shows you how to access Eucalyptus with a web-based console and with command line tools. This section also describes how to perform common management tasks.

This document is intended to be a reference. You do not need to read it in order, unless you are following the directions for a particular task.

Document version: Build 3029 (2016-01-26 12:00:31)

Overview of Eucalyptus

Eucalyptus is a Linux-based software architecture that implements scalable, efficiency-enhancing private and hybrid clouds within an enterprise's existing IT infrastructure. Because Eucalyptus provides Infrastructure as a Service (IaaS), you can provision your own resources (hardware, storage, and network) through Eucalyptus on an as-needed basis.

A Eucalyptus cloud is deployed across your enterprise's on-premise data center. As a result, your organization has a full control of the cloud infrastructure. You can implement and enforce various level of security. Sensitive data managed by the cloud does not have to leave your enterprise boundaries, keeping data completely protected from external access by your enterprise firewall.

Eucalyptus was designed from the ground up to be easy to install and non-intrusive. The software framework is modular, with industry-standard, language-agnostic communication. Eucalyptus is also unique in that it provides a virtual network overlay that isolates network traffic of different users as well as allows two or more clusters to appear to belong to the same Local Area Network (LAN).

Eucalyptus also is compatible with Amazon's EC2, S3, and IAM services. This offers you hybrid cloud capability.

Command Line Interface

Eucalyptus supports two command line interfaces (CLIs): the administration CLI and the user CLI.

The administration CLI is installed when you install Eucalyptus server-side components. The administration CLI is for maintaining and modifying Eucalyptus.

The other user CLI, called Euca2ools, can be downloaded and installed on clients. Euca2ools is a set of commands for end users and can be used with both Eucalyptus and Amazon Web Services (AWS).

The commands used in this guide assume that the environment variables exported by a eucarc file for an administrative Eucalyptus user have been set. For more information, see the *Eucalyptus Installation Guide*.

Manage Your Cloud

After you install and initially configure Eucalyptus, there are some common administration tasks you can perform. This section describes these tasks and associated concepts.

Tip: The **System Management** section of the **Quick Links** area allows you to go to the **Start Guide** or the **Service Components** page.

Cloud Overview

This topic presents an overview of the components in Eucalyptus.

Eucalyptus is comprised of several components: Cloud Controller, Walrus, Cluster Controller, Storage Controller, and Node Controller. Each component is a stand-alone web service. This architecture allows Eucalyptus both to expose each web service as a well-defined, language-agnostic API, and to support existing web service standards for secure communication between its components.

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

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

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

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

Node Controller The Node Controller (NC) executes on any machine that hosts VM instances. The NC controls VM activities, including the execution, inspection, and termination of VM instances. It also fetches and maintains a local cache of instance images, and it queries and controls the system software (host OS and the hypervisor) in response to queries and control requests from the CC. The NC is also responsible for the management of the virtual network endpoint.

Networking Configuration Options

All network-related options specified in /etc/eucalyptus/eucalyptus.conf use the prefix VNET_. The most commonly used VNET options are described in the following table.

Important: If you change the value of in the `eucalyptus.conf` file, you must restart the Cluster Controller.

Option	Description	Modes

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

`VNET_BRIDGE`	On an NC, this is the name of the bridge interface to which instances' network interfaces should attach. A physical interface that can reach the CC must be attached to this bridge. Common setting for KVM is `br0`.	Edge (on NC) Managed (No VLAN)
`VNET_DHCPDAEMON`	The ISC DHCP executable to use. This is set to a distro-dependent value by packaging. The internal default is `/usr/sbin/dhcpd3`.	Edge (on NC) Managed Managed (No VLAN)
`VNET_DHCPUSER`	The user the DHCP daemon runs as on your distribution. For CentOS 6 and RHEL 6, this is typically `root`. Default: `dhcpd`	Managed Managed (No VLAN)

Option	Description	Modes
`VNET_DNS`	The address of the DNS server to supply to instances in DHCP responses. Example: `VNET_DNS="173.205.188.129"`	Managed Managed (No VLAN)
`VNET_LOCALIP`	By default the CC automatically determines which IP address to use when setting up tunnels to other CCs. Set this to the IP address that other CCs can use to reach this CC if tunneling does not work.	Managed Managed (No VLAN)
`VNET_MACPREFIX`	This option is used to specify a prefix for MAC addresses generated by Eucalyptus for VM instances. The prefix has to be in the form `HH:HH` where H is a hexadecimal digit. Example: `VNET_MACPREFIX="D0:D0"`	Managed, Managed (No VLAN)
`VNET_MODE`	The networking mode in which to run. The same mode must be specified on all CCs and NCs in your cloud. Valid values: `EDGE` `MANAGED`, `MANAGED-NOVLAN`,	All
`VNET_PRIVINTERFACE`	The name of the network interface that is on the same network as the NCs. In Managed and Managed (No VLAN) modes this must be a bridge for instances in different clusters but in the same security group to be able to reach one another with their private addresses. Default: `eth0`	Edge (on NC) Managed
`VNET_PUBINTERFACE`	**On a CC**, this is the name of the network interface that is connected to the "public" network. **On an NC**, this is the name of the network interface that is connected to the same network as the CC. Depending on the hypervisor's configuration this may be a bridge or a physical interface that is attached to the bridge. Default: `eth0`	Edge (on NC) Managed Managed (No VLAN)

`VNET_PUBLICIPS`	A space-separated list of individual and/or hyphenated ranges of public IP addresses to assign to instances. If you do not set a value for this option, all instances will receive only private IP addresses.
	Example:
	`VNET_PUBLICIPS=`
	`"173.205.188.140-173.205.188.254"`
	Tip: To offer more public IPs, you can span subnets. However you must list each subnet range separately. For example: `"10.133.82.50-10.133.82.254`
	`10.133.83.0-10.133.83.254"`
		Managed
		Managed (No VLAN)

Option	Description	Modes
`VNET_SUBNET, VNET_NETMASK`	These options control the internal private network used by instances within Eucalyptus. Eucalyptus assigns a distinct subnet of private IP addresses to each security group. This setting dictates how many addresses each of these subnets should contain. Specify a power of 2 between 16 and 2048. This is directly related, though not equal, to the number of instances that can reside in each security group. Eucalyptus reserves eleven addresses per security group.	Managed, Managed (No VLAN)

Cloud Best Practices

This section details Eucalyptus best practices for your private cloud.

Synchronize Clocks

Eucalyptus checks message timestamps across components in the cloud infrastructure. This assures command integrity and provides better security.

Eucalyptus components receive and exchange messages using either Query or SOAP interfaces (or both). Messages received over these interfaces are required to have some form of a time stamp (as defined by AWS specification) to prevent message replay attacks. Because Eucalyptus enforces strict policies when checking timestamps in the received messages, for the correct functioning of the cloud infrastructure, it is crucial to have clocks constantly synchronized (for example, with ntpd) on all machines hosting Eucalyptus components. To prevent user command failures, it is also important to have clocks synchronized on the client machines.

Following the AWS specification, all Query interface requests containing the Timestamp element are rejected as expired after 15 minutes of the timestamp. Requests containing the Expires element expire at the time specified by the element. SOAP interface requests using WS-Security expire as specified by the WS-Security Timestamp element.

When checking the timestamps for expiration, Eucalyptus allows up to 20 seconds of clock drift between the machines. This is a default setting. You can change this value for the CLC at runtime by setting the `bootstrap.webservices.clock_skew_sec` property as follows:

```
euca-modify-property -p
bootstrap.webservices.clock_skew_sec=<new_value_in_seconds>
```

For additional protection from the message replay attacks, the CLC implements a replay detection algorithm and rejects messages with the same signatures received within 15 minutes. Replay detection parameters can be tuned as described in *Configure Replay Protection*.

Configure SSL

In order to connect to Eucalyptus using SSL, you must have a valid certificate for the Cloud Controller (CLC). You must also be running the Cloud Controller and Cluster Controller (CC) on separate machines.

Create a keystore

Eucalyptus uses a PKCS12-format keystore. If you are using a certificate signed by a trusted root CA, use the following command to convert your trusted certificate and key into an appropriate format:

```
openssl pkcs12 -export -in [YOURCERT.crt] -inkey [YOURKEY.key] \
 -out tmp.p12 -name [key_alias]
```

Note: this command will request an export password, which is used in the following steps.

Save a backup of the Eucalyptus keystore, at `/var/lib/eucalyptus/keys/euca.p12`, and then import your keystore into the Eucalyptus keystore as follows:

```
keytool -importkeystore \
-srckeystore tmp.p12 -srcstoretype pkcs12 -srcstorepass [export_password] \
-destkeystore /var/lib/eucalyptus/keys/euca.p12 -deststoretype pkcs12 \
-deststorepass eucalyptus -alias [key_alias] \
-srckeypass [export_password]
```

Enable the Cloud Controller to use this keystore

Run the following commands on the Cloud Controller (CLC):

```
euca-modify-property -p bootstrap.webservices.ssl.server_alias=[key_alias]
euca-modify-property -p \
bootstrap.webservices.ssl.server_password=[export_password]
```

Restart the CLC by running `service eucalyptus-cloud restart` or `/etc/init.d/eucalyptus-cloud restart`

Optional: Configure the Cloud Controller to redirect requests on port 443 to port 8773

The Cloud Controller listens for both SSL and non-SSL connections on port 8773. If you have other tools that expect to speak SSL on port 443, you should forward requests on that port to port 8773. For example, the following iptables command can be used:

```
iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-ports 8773
```

High Availability

This topic explains recommendations for high availability deployments.

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

- Detection of service faults and monitoring of system health: gather service status, determine current service topology, admit requests which can be satisfied using only healthy services in that topology
- Tools for interrogating the system's health: access to service state information
- Error gathering to aid in determining the cause: access to fault information as it impacts service function
- Automated failover when redundant services are configured: removal of faulty services and enabling of healthy services
- Service state control: ability to remove individual component-services (when configured with HA pair) from operation without disrupting service
- Replacement/restoration of component-services: procedures for restoring/replacing a component service after a total-loss failure (e.g., disk failure, host combustion, etc.)

In addition to previously detailed deployment recommendations, delivering highly available services with Eucalyptus depends on appropriate operational and maintenance support. The following sections detail the related system functionality and procedures.

Understanding Service State

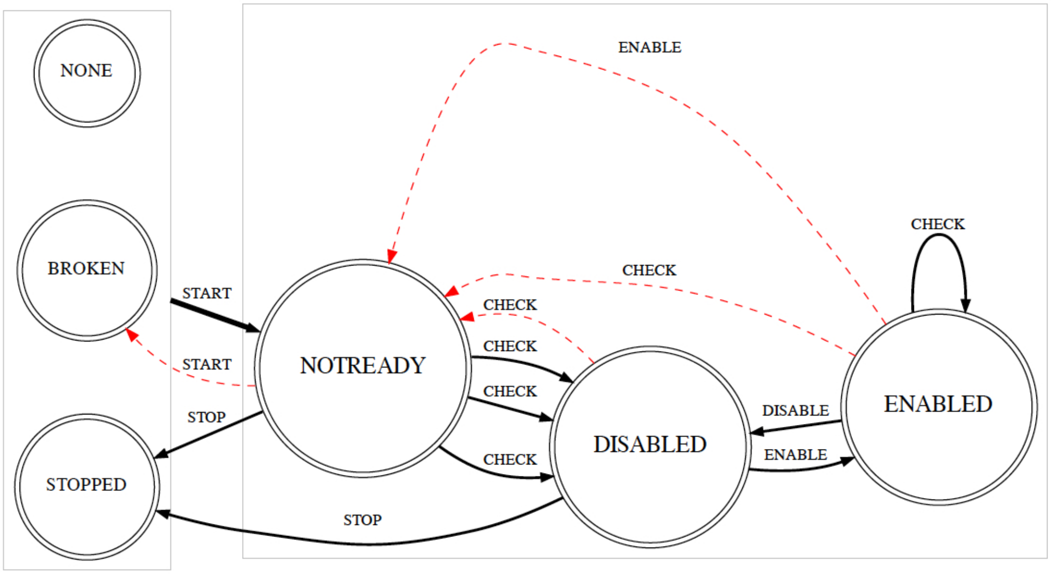
The system monitors service health and enables healthy services to process user requests while marking faulty services as being NOTREADY. Each component service is interrogated by the system to determine its current state. Faults are detected either:

- The service reporting a fault has been detected (for example, due to misconfiguration, dependency service failure, environmental fault, etc.)
- Failure to contact the service

The following table overviews the relevant states

State	Operational	In use by system	Description
ENABLED	Yes	Yes	Service is operating correctly and is selected for processing requests
DISABLED	Yes	No	Service is operating correctly but is not selected for processing requests
NOTREADY	No	No	Service is failing to operate correctly
BROKEN	No	No	Service is not contactable by the system
STOPPED	N/A	No	Service has been stopped by an administrator

The following diagram indicates the set of relevant states and transitions between them. Black arrows indicate a transition between states that is initiated by the system or an administrator request. Red errors indicate a failure to transition into the originating state that results in a transition to the destination error state.



Based on the collected service state, the system will:

- Attempt to advance previously non-functioning services to a functional state
- Determine whether any functioning services can be `ENABLED` and added to the set used for serving requests

On the Cloud Controller host, with eucalyptus admin credentials loaded, run `euca-describe-services` to see up-to-date service information including the state of each service as described in the above table.

Understanding System Availability

The impact of a service fault on the system's availability depends upon the deployment and configuration of the system. The following table details the scope a service fault can have on system availability for each component type.

Component	Scope (Fault Region)	Description
Cloud Controller	Cloud	The CLC is a cloud-wide service and must have at least one operation service.
Walrus	Cloud	Walrus is a cloud-wide service and must have at least one operation service.
Cluster Controller	Availability Zone	CCs are associated with a partition and service requests specific to an availability zone. Should an availability zone not have an operational CC, instance requests will be rejected for the corresponding zone.
Storage Controller	Availability Zone	Storage controllers are associated with a partition and service requests specific to an availability zone. Should an availability zone not have an operational storage controller volume and snapshot creation requests will be rejected for the corresponding zone
Arbitrators	User-facing Service Host	Arbitrators are associated with a host that runs user-facing component services (CLC, Walrus). Each host must have an operational Arbitrator. Should a component service host have a configured but faulty Arbitrator, a fail-stop condition occurs and locally hosted services report a `NOTREADY` error.
Node Controller	Compute Host	NCs are associated with each node and interact with the hypervisor to service node-specific requests.

A quick way to evaluate system availability is to determine whether:

- The cloud has an enabled CLC
- The cloud has an enabled Walrus
- The availability zone has an enabled CC
- The availability zone has an enabled SC
- The user-facing service host has one reachable Arbitrator per host (if you configure an Arbitrator)

Storage Volumes

Eucalyptus manages storage volumes for your private cloud. Volume management strategies are application specific, but this topic includes some general guidelines.

When setting up your Storage Controller, consider whether performance (bandwidth and latency of read/write operations) or availability is more important for your application. For example, using several smaller volumes will allow snapshots to be taken on a rolling basis, decreasing each snapshot creation time and potentially making restore operations faster if the restore can be isolated to a single volume. However, a single larger volume allows for faster read/write operations from the VM to the storage volume.

An appropriate network configuration is an important part of optimizing the performance of your storage volumes. For best performance, each Node Controller should be connected to a distinct storage network that enables the NC to communicate with the SC or SAN, without interfering with normal NC/VM-instance network traffic.

Eucalyptus includes configurable limits on the size of a single volume, as well as the aggregate size of all volumes on an SC. The SC can push snapshots from the SAN device, where the volumes reside, to Walrus, where the snapshots become available across multiple clusters. Smaller volumes will be much faster to snapshot and transfer, whereas large volumes will take longer. However, if many concurrent snapshot requests are sent to the SC, operations may take longer to complete.

Although an SC can manage an arbitrary number of volumes, intermittent issues have been reported with some hypervisors when attaching more than 16 volumes to a single NC. Where possible, limiting the number of volumes to no more than 16 per NC is advisable.

EBS volumes are created from snapshots on the SC or SAN, after the snapshot has been downloaded from Walrus to the device. Creating an EBS volume from a snapshot on the same cluster as the source volume of the snapshot will reduce delays caused by having to transfer snapshots from Walrus.

Cloud Tasks

This section contains a listing of your Eucalyptus cloud-related tasks.

Inspect System Health

Eucalyptus provides access to the current view of service state and the ability to manipulate the state. You can inspect the service state to either ensure system health or to identify faulty services. You can modify a service state to maintain activities and apply external service placement policies.

View Service State

Use the `euca-describe-services` command to view the service state. The output indicates:

- Component type of the service
- Partition in which the service is registered
- Unique name of the service
- Current view of service state
- Last reported epoch (this can be safely ignored)
- Service URI
- Fully qualified name of the service (This is needed for manipulating services that did not get unique names during registration. For example: internal services like reporting or DNS)

The default output includes the services that are registered during configuration, as well as information about the DNS service, if present. You can obtain additional service state information, such as internal services, by providing the `-system-internal` flag.

You can also make requests to retrieve service information that is filtered by either:

- current state (for example, `NOTREADY`)
- host where service is registered
- partition where service is registered
- type of service (for example, CC or Walrus)

When you investigate service failures, you can specify `-events` to return a summary of the last fault. You can retrieve extended information (primarily useful for debugging) by specifying `-events -events-verbose`.

Heartbeat Service

`http://CLCIPADDRESS:8773/services/Heartbeat` provides a list of components and their respective statuses. This allows you to find out if a service is enabled without requiring cloud credentials.

Modify Service State

To modify a service:

Enter the following command on the CLC, Walrus, or SC machines:

```
eucalyptus-cloud stop
```

On the CC, use the following command:

```
eucalyptus-cc stop
```

If, for example, you have SCs that are correctly configured and operating in HA mode. However, you want to shut down the primary SC for maintenance. The primary SC is `SC00` and the secondary SC is `SC01`. `SC00` is `ENABLED` and `SC01` is `DISABLED`.

To stop `SC00` and cause `SC01` to take over, you would enter the following command on SC00:

```
eucalyptus-cloud stop
```

To check status of services, you would enter:

```
euca-describe-services
```

When `SC01` starts, the eucalyptus-cloud process on the host that SC00 is shutdown and maintenance tasks can be performed. When maintenance is complete, you can start the eucalyptus-cloud process on `SC00`. `SC00` will enter the `DISABLED` state by default. You can chose to let `SC01` continue to be the primary and `SC00` will be the secondary.

If you want to designate `SC00` as the primary, make sure no volumes or snapshots are being created and that no volumes are being attached or detached, and then enter on SC01:

```
eucalyptus-cloud stop
```

Monitor the state of services using `euca-describe-services` until `SC01` is marked `DISABLED` and `SC00` is `ENABLED`.

View User Resources

To see resource use by your cloud users, Eucalyptus provides the following commands with the `-verbose` flag.

- `euca-describe-groups verbose`: Returns information about security groups in your account, including output type identifier, security group ID, security group name, security group description, output type identifier, account ID of the group owner, name of group granting permission, type of rule, protocol to allow, start of port range, end of port range, source (for ingress rules) or destination (for egress rules), and any tags assigned to the security group.
- `euca-describe-instances verbose`: Returns information about your instances, including output type identifier, reservation ID, name of each security group the instance is in, output type identifier, instance ID for each running instance, EMI ID of the image on which the instance is based, public DNS name associated with the instance (for instances in the running state), private DNS name associated with the instance (for instances in running state), instance state, key name, launch index, instance type, launch time, availability zone, kernel ID, ramdisk ID, monitoring state, public IP address, private IP address, type of root device (ebs or instance-store), placement group the cluster instance is in, virtualization type (paravirtual or hvm), any tags assigned to the instance, hypervisor type, block device identifier for each EBS volume the instance is using, along with the device name, the volume ID, and the timestamp.
- `euca-describe-keypairs verbose`: Returns information about key pairs available to you, including keypair identifier, keypair name, and private key fingerprint.
- `euca-describe-snapshots verbose`: Returns information about EBS snapshots available to you, including snapshot identifier, ID of the snapshot, ID of the volume, snapshot state (pending, completed, error), timestamp when snapshot initiated, percentage of completion, ID of the owner, volume sized, description, and any tags assigned to the snapshot.

- `euca-describe-volumes verbose`: Describes your EBS volumes, including volume identifier, volume ID, size of the volume in GiBs, snapshot from which the volume was created, availability zone, volume state (creating, available, in-use, deleting, deleted, error), timestamp of the volume creation, and any tags assigned to the volume.

List Arbitrators

To see a list a arbitrators running on your cloud, perform the steps listed in this topic.

- Enter the following command to display Arbitrators for the current CLC or Walrus:

```
/usr/sbin/euca-describe-services --system-internal
```

- Enter the following command to display Arbitrators on both primary and secondary CLCs or Walruses:

```
/usr/sbin/euca_conf --list-arbitrators
```

Change Network Configuration

You might want to change the original network configuration of your cloud. To change your network configuration, perform the tasks listed in this topic.

1. Log in to the CLC and open the `/etc/eucalyptus/eucalyptus.conf` file.
2. Navigate to the Networking Configuration section and make your edits.
3. Save the file.
4. Restart the Cluster Controller.

```
service eucalyptus-cc restart
```

Add a Node Controller

If you want to increase your system's capacity, you'll want to add more Node Controllers (NCs).

To add an NC, perform the following tasks:

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

```
/usr/sbin/euca_conf --register-nodes \ "[Node1_IP]; ...
[NodeN_IP];"
```

2. When prompted, enter the password to log into each node.

   Eucalyptus requires this password to propagate the cryptographic keys.

Migrate Instances Between Node Controllers

In order to ensure optimal system performance, or to perform system maintenance, it is sometimes necessary to move running instances between Node Controllers (NCs). You can migrate instances individually, or migrate all instances from a given NC.

Important: For migrations to succeed, you must have `INSTANCE_PATH` set to the same value in the `eucalyptus.conf` file on each Node Controller.

- To migrate a single instance to another NC, enter the following command:

```
euca-migrate-instances -i [instance_id]
```

  You can also optionally specify `--dest=[destination NC IP]` or `--exclude-dest=[excluded NC IP]`, to ensure that the instance is migrated to one of the specified Node Controllers, or to avoid migrating the instance to any of the specified Node Controllers. These flags may be used more than once to specify multiple Node Controllers.

- To migrate all instances away from a Node Controller, enter the following command:

```
euca-migrate-instances --source=[NC IP]
```

You can also optionally specify `--stop-source`, to stop the specified Node Controller and ensure that no new instances are started on that NC while the migration occurs. This allows you to safely remove the NC without interrupting running instances. The NC will remain in the DISABLED state until it is explicitly enabled using `euca-modify-service -s start [NC IP]`.

- In some cases, timeouts may cause a migration to initially fail. Run the command again to complete the migration.

Remove a Node Controller

Describes how to delete NCs in your system.

If you want to decrease your system's capacity, you'll need to decrease NC servers. To delete an NC, perform the following tasks.

Log in to the CC and enter the following command:

```
/usr/sbin/euca_conf --deregister-nodes "<nodeName1> ... <nodeNameN>"
```

Restart Eucalyptus

Describes the recommended processes to restart Eucalyptus, including terminating instances and restarting Eucalyptus components.

You must restart Eucalyptus whenever you make a physical change (e.g., switch out routers), or edit the eucalyptus.conf file. To restart Eucalyptus, perform the following tasks in the order presented.

Tip: Before you restart Eucalyptus, we recommend that you notify all users that you are terminating all instances.

Shut Down All Instances

To terminate all instances on all NCs perform the steps listed in this topic.

To terminate all instances on all NCs:

Enter the following command:

```
euca-terminate-instances <instance_id>
```

Restart the CLC

Log in to the CLC and enter the following command:

```
service eucalyptus-cloud restart
```

All Eucalyptus components on this server will restart.

Restart Walrus

Log in to Walrus and enter the following command:

```
service eucalyptus-cloud restart
```

Restart the CC

Log in to the CC and enter the following command:

```
service eucalyptus-cc restart
```

Restart the SC

Log in to the SC and enter the following command:

```
service eucalyptus-cloud restart
```

Restart an NC

To restart an NC perform the steps listed in this topic.

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

```
service eucalyptus-nc restart
```

2. Repeat for each NC.

> You can automate the restart command for all of your NCs. Store a list of your NCs in a file called `nc-hosts` that looks like:
>
> ```
> nc-host-00
> nc-host-01
> ...
> nc-host-nn
> ```
>
> To restart all of your NCs, run the following command:
>
> ```
> cat nc-hosts | xargs -i ssh root@{} service eucalyptus-nc restart
> ```

Shut Down Eucalyptus

Describes the recommended processes to shut down Eucalyptus.

There may be times when you need to shut down Eucalyptus. This might be because of a physical failure, topological change, backing up, or making an upgrade. We recommend that you shut down Eucalyptus components in the reverse order of how you started them. To stop the system, shut down the components in the order listed.

Tip: Before you shut Eucalyptus down, we recommend that you notify all users that you are terminating all instances.

Shut Down All Instances

To terminate all instances on all NCs perform the steps listed in this topic.

To terminate all instances on all NCs:

Enter the following command:

```
euca-terminate-instances <instance_id>
```

Shut Down the NCs

To shut down the NCs perform the steps listed in this topic.

To shut down the NCs:

1. Log in as root to a machine hosting an NC.
2. Enter the following command:

```
service eucalyptus-nc stop
```

3. Repeat for each machine hosting an NC.

Shut Down the CCs

To shut down the CCs:

1. Log in as root to a machine hosting a CC.
2. Enter the following command:

```
service eucalyptus-cc stop
```

3. Repeat for each machine hosting a CC.

Shut Down the SCs

To shut down the SC:

1. Log in as root to the physical machine that hosts the SC.
2. Enter the following command:

```
service eucalyptus-cloud stop
```

3. Repeat for any other machine hosting an SC.

Shut Down Walrus

To shut down Walrus:

1. Log in as root to the physical machine that hosts Walrus.
2. Enter the following command:

```
service eucalyptus-cloud stop
```

Shut Down the CLC

To shut down the CLC:

1. Log in as root to the physical machine that hosts the CLC.
2. Enter the following command:

```
service eucalyptus-cloud stop
```

Back Up the Database

To back up the cloud database follow the steps listed in this topic.

1. Extract the Eucalyptus PostgreSQL database cluster into a script file.

```
pg_dumpall --oids -c -h/var/lib/eucalyptus/db/data -p8777 -Uroot
-f~/eucalyptus_pg_dumpall-backup.sql
```

2. Back up the keys directory.

```
tar -czvf ~/eucalyptus-keydir.tgz /var/lib/eucalyptus/keys
```

Disable CloudWatch

To disable CloudWatch, run the following command.

```
euca-modify-property -p
               <partition>.cloudwatch.disable_cloudwatch_service=true
```

Operations

This section contains concepts and tasks associated with operating your Eucalyptus cloud.

Operations Overview

This section is for architects and cloud administrators who plan to deploy Eucalyptus in a production environment. It is not intended for end users or proof-of-concept installations.

To run Eucalyptus in a production environment, you must be aware of your hardware and network resources. This guide is to help you make decisions about deploying Eucalyptus. It is also meant to help you keep Eucalyptus running smoothly.

Planning Your Deployment

To decide on your deployment's scope, determine the use case for your cloud. For example, will this be a small dev-test environment, or a large and scalable web services environment?

To help with scoping your deployment, we recommend you go to the *Eucalyptus Reference Architectures* page. There you will find the most popular use cases and the physical resources required.

Testing Your Deployment

This topic details what you should test when you want to make sure your deployment is working. The following suggested test plan contains tasks that ensure DNS, imaging, and storage are working.

DNS

- Verify that instances can ping their:
    - Private DNS name
    - Public DNS name
- Verify that instances are pingable on their public DNS names from:
    - Outside the cloud
    - An instance inside the cloud

Imaging

- Verify that an EBS-backed image boots successfully
- Verify that you can create an image from a running EBS-backed instance
- Verify that you can install a new Ubuntu image
- Verify that you can deregister an image
- Verify that you can import an instance
- Verify that you can import a volume

Walrus

- Verify that you can make a basic s3cmd request
- Verify that you can successfully perform a multi-part upload (use a 1G+ file)

Customizing Your Deployment

For most production deployments, we recommend that you use a configuration management tool. Customers have been successful deploying using the following:

- Chef
- Puppet F-Secure
- Anisible

This section describes the most commonly applied post-install customizations and the issues they pose:

- Over-subscription
- Networking changes (Edge and managed modes)
- Reporting / CloudWatch tweaks/customizations
- Capacity changes

Over-subscription

Over-subscription refers to the practice of expanding your computer beyond its limits. Over-subscription applies only to node controllers. You may modify disks and cores to allow enough usage buffer for your instance.

1. Navigate to `/etc/eucalyptus/` and locate the `eucalyptus.conf` file.
2. Edit the following values to define the appropriate size buffers for your instances:

Option	Description
`NC_WORK_SIZE`	Defines the amount of disk space available for instances to be run.
`NC_CACHE_SIZE`	Defines how much disk space is needed for images to be cached.
`MAX_CORES`	Defines the number of cores that are available for VMs.

3. In order for these changes to take effect, you must restart the NC.

Networking Changes (EDGE and Managed Modes)

You can modify the default by adding network IPs to your cloud or changing your network from managed to EDGE network. Changing these values do not require turning down the whole system.

Add Network IPs

To add network IPs, perform one of the following:

1. In Edge network mode, adding or changing the IP involves creating a JSON file and uploading it the Cloud Controller (CLC). See *Configure for Edge Mode* for more details.
   No restart needed, changes apply automatically.
2. In managed mode, navigate to `/etc/eucalyptus/` and locate the `eucalyptus.conf` file.
   a) Add more IPs by specifying them in the `VNET_PUBLICIPS` parameter.
   b) Restart the CC and CLC in order to apply the changes.

Change Modes

You can modify the default network from managed to Edge network.

See *Eucalyptus Migration to Edge Networking Mode* for more details.

Change Reporting/CloudWatch Properties

You can change the following reporting and CloudWatch properties:

Reporting Property	Description
`cloud.monitor.default_poll_interval_mins`	If set to 0 = no reporting. The more often you poll, the more hit on the performance.
`reporting.default_write_interval_mins`	How often polled data is written to the database.
`cloud.monitor.history_size`	How many data points per poll interval will be collected or how many samples per poll interval.
`cloudwatch.disable_cloudwatch_service`	Disables cloudwatch when set to true.

Change Capacity

Capacity changes refer to adding another cluster or more nodes.

1. To add another cluster, *install*, *start*, and *register*.
2. To add more nodes, see *Add a Node Controller*.

Managing Policies

This topic details best practices for managing your cloud policies.

- Establish a workflow for account creation, including the initial request for a cloud account and the email containing credentials.
- Limit your use of individual policies. Focus your policies on groups and add individuals to the group.
- Use groups to assign permissions to individual users. Limit the use of policies for individual users.

For more information about policy best practices, see *IAM Best Practices*.

Networking

This topic addresses networking in the Eucalyptus cloud.

Networking Modes

Eucalyptus offers different modes to provide you with a cloud that will fit in your current network. For information what each networking mode has to offer, see *Plan Networking Modes*.

EC2-Classic Networking

Eucalyptus supports EC2-Classic networking. Your instances run in a single, flat network that you share with others. For more information about EC2-Classic networking, go to *Differences Between Instances in EC2-Classic and EC2-VPC*.

More Information

For more information about networking, go to the following resources:

- *Next Generation Network Driver* (introductory for how Eucalyptus is using networking)
- *Midokura and Eucalyptus*
- *Edge Networking Mode*
- *Standard Topology Overview* (this PDF is high-level and good for introductory material but not for troubleshooting)

Monitoring

This topic includes details about which resources you should monitor.

Component	Open Ports	Running Processes
Cloud Controller (CLC)	8773 (web services), 8777 (PostgreSQL)	eucalyptus-cloud, postgres
User-facing services (UFS)		eucalyptus-cloud
Walrus		eucalyptus-cloud
Cluster Controller (CC)		eucalyptus-cloud
Storage Controller (SC)		eucalyptus-cloud, tgtd (for DAS and Overlay)
Node Controller (NC)		httpd, dhcpd, eucanetd (edge modes), qemu-kvm / 1 per instance
Management Console	8888	eucaconsole

Backup and Recovery

This section details how to backup your data, as well as steps to take if things go wrong.

Back Up Your Cloud

This section explains what you need to back up to protect your cloud data.

We recommend that you back up the following data:

- The cloud database: see *Back Up the Database*
- Object storage. For objects in Walrus, the frequency depends on current load. Use your own discretion to determine backup plan and strategy. You must have Walrus running. For information about backing up Riak CS, go to *Backing Up Riak*.
- Volumes in each cluster (DAS and Overlay)
- Configuration files for each Eucalyptus component (`/etc/eucalyptus/eucalyptus.conf`)
- Eucalyptus and LVM snapshots
- SAN technologies vary, so see the backup documentation for your SAN.

Users are responsible for volume backups using EBS snapshots on their defined schedules.

Back Up the Database

To back up the cloud database follow the steps listed in this topic.

1. Extract the Eucalyptus PostgreSQL database cluster into a script file.

```
pg_dumpall --oids -c -h/var/lib/eucalyptus/db/data -p8777 -Uroot
-f~/eucalyptus_pg_dumpall-backup.sql
```

2. Back up the keys directory.

```
tar -czvf ~/eucalyptus-keydir.tgz /var/lib/eucalyptus/keys
```

Recover Cloud Data

This topic explains what steps to take to bring your backed-up data to your cloud.

We recommend that you back up the following data:

- The cloud database: see *Restore the Database*
- Objects in Walrus: The frequency depends on current load. Use your own discretion to determine backup plan and strategy You must have Walrus running.
- Volumes in each cluster (DAS and Overlay)

- Configuration files for each Eucalyptus component (`/etc/eucalyptus/eucalyptus.conf`)
- Eucalyptus and LVM snapshots
- SAN technologies vary, so see the backup documentation for your SAN.

Users are responsible for volume backups using EBS snapshots on their defined schedules.

Restore the Database

To restore the cloud database follow the steps listed in this topic.

1. Stop the CLC service.

```
/etc/init.d/eucalyptus-cloud stop
```

2. Remove traces of the old database.

```
rm -rf /var/lib/eucalyptus/db
```

3. Re-initialize the database structure.

```
euca_conf --initialize
```

4. Start the database manually.

```
su eucalyptus -c "/usr/pgsql-9.1/bin/pg_ctl start -w \
-s -D/var/lib/eucalyptus/db/data -o '-h0.0.0.0/0 -p8777 -i'"
```

5. Restore the backup.

```
psql -U root -d postgres -p 8777 -h /var/lib/eucalyptus/db/data -f
~/eucalyptus_pg_dumpall-backup.sql
```

6. Restore the keys.

```
tar -xvf ~/eucalyptus-keysdir.tgz -C /
```

7. Stop the database manually.

```
su eucalyptus -c "/usr/pgsql-9.1/bin/pg_ctl stop -D/var/lib/eucalyptus/db/data"
```

8. Start CLC service

```
/etc/init.d/eucalyptus-cloud start
```

Recovering from a Failure: Walrus

Some sample scenarios in which we offer solutions.

In these examples, we will assume that Walrus `WS00` is the primary and `WS01` is the secondary Walrus server.

Software Failure Example

In this scenario, `WS01` refuses to go to `DISABLED` state. DRBD complains that it is in split brain mode. `drbdadm cstate r0` shows that DRBD is in WFConnection state.

If you are sure that data on `WS01` is out of date and can be discarded, execute the following commands to restore HA mode.

1. Shut down the eucalyptus-cloud process on WS01.
2. Ensure that the DRBD connection is down by typing "drbdadm disconnect r0" on any of the two Walrus hosts.
3. On the primary Walrus, WS00, set drbd as the primary by executing "drbdadm primary r0"
4. On the secondary Walrus, WS01, execute the following command:

```
drbdadm -- --discard-my-data connect
```

Warning: This command will discard all data on WS01 and synchronize data from WS00.

5. Monitor the state of DRBD by running:

   ```
   watch -n 2 cat /proc/drbd
   ```

6. When the data on WS01 is synced, start the eucalyptus-cloud process on WS01.

Hardware Failure Example

In this example, the primary `WS00` needs to be taken out of service due to a hardware failure, such as a failed disk.

1. Shut down the eucalyptus-cloud process on WS00 if it is still running.
2. Monitor service status by running `euca-describe-services` on WS01 and ensure that WS01 has taken over as the new primary (state: ENABLED).
3. Shut down the host running WS00.
4. If the host running WS00 is to be replaced entirely or the OS reinstalled:

   - On the primary CLC, enter the following to deregister WS00:

     ```
     euca_conf --deregister-walrusbackend --component WS00 partition <name of
     partition>
           --host <WS00 host>
     ```

   - After Linux has been installed on the new WS00 host and it is ready for use, please reinstall the "eucalyptus-walrus" package.
   - Synchronize the DRBD configuration (/etc/drbd.conf and /etc/eucalyptus/drbd*) from the WS01 host.
   - On WS00, re-configure DRBD by following the Configure DRBD section of the Installation Guide and performing the steps that are relevant to the secondary Walrus server (WS00 is the new secondary Walrus server, in this example).
   - Re-register WS00 with a new host name if necessary. This will synchronize keys.

5. On WS00, execute the following command:

   ```
   drbdadm -- --discard-my-data connect
   ```

   Warning: This command will discard all data on WS00 and synchronize data from WS01.

6. Monitor the state of DRBD by entering:

   ```
   watch -n 2 cat /proc/drbd
   ```

   WS01 should be marked as the primary and WS00 is the new secondary. Wait until data is synchronized.

7. When the data on WS00 is synced from WS01, start the eucalyptus-cloud process on WS00.
8. Monitor service status by running "euca-describe-services" on the primary CLC and ensure that WS00 is DISABLED and WS01 is ENABLED.

At this point, the Walrus service is back in HA mode.

Troubleshooting

This topic details how to find information you need to troubleshoot most problems in your cloud.

To troubleshoot Eucalyptus, you must have the following:

- a knowledge about which machines each Eucalyptus component is installed on
- root access to each machine hosting Eucalyptus components:

  - Cloud Controller (CLC)
  - User-facing services (UFS)

    - Walrus
    - Storage Controller (SC)
    - Cluster Controller (CC)
    - Node Controller (NC)
- an understanding of the network configuration connecting the Eucalyptus components

For most problems, the procedure for tracing problems is the same: start at the bottom to verify the bottom-most component, and then work your way up. If you do this, you can be assured that the base is solid. This applies to virtually all Eucalyptus components and also works for proactive, targeted monitoring.

For more information about troubleshooting, go to Tips to Troubleshooting Eucalyptus *Part 1* and *Part 2*.

Eucalyptus Log Files

Usually when an issue arises in Eucalyptus, you can find information that points to the nature of the problem either in the Eucalyptus log files or in the system log files. This topic details log file message meanings, location, configuration, and fault log information.

Log File Location and Content

By default, the Eucalyptus log files are stored in `/var/log/eucalyptus/` on each machine that hosts a Eucalyptus component. If Eucalyptus is installed somewhere other than the filesystem root (`/`), log files are stored in `$EUCALYPTUS/var/log/eucalyptus/`.

CLC, Walrus, SC, and UFS Log Files

Cloud controller (CLC), Walrus, Storage controller (SC), and User-Facing Services (UFS) log files are as follows:

Log File	Description
`cloud-cluster.log`	This log contains information about your clusters as the CLC sees things: current status, current capacity, and any problems. These logs can help you detect if there is a capacity or communication issue associated with your clusters.
`cloud-fault.log`	This file is reserved for issues with known error codes and known resolutions.
`cloud-output.log`	This file contains all info-level logs for the Java component itself. If there are multiple Java components on a single host (for example, CLC and Walrus), the info-level logs for all of the components will go here.
`cloud-debug.log`	This file contains all messages generated from debug-level logging.
`cloud-error.log`	This file contains is enabled by default alongside info. Along with `cloud-output.log`, the `cloud-error.log` is one of the first places you should look.
`cloud-exhaust.log`	This file is full of errors and warnings.
`cloud-extreme.log`	This is legitimately a setting for developers only, because production usage would fill up the hard drive with log files very quickly.

Log File	Description
`startup.log`	STDOUT and STDERR are redirected to this log file for system startup. This file contains system JVM startup output including system bootstrap information, component bootstrap configuration, local service discovery, and network interfaces.
`upgrade.log`	This file records the output from upgrade process. Same as seen on the command line when upgrading.
`cloud-requests.log`	This file is only located on the UFS component and logs the system requests made to the different services: ec2, autoscaling, cloudformation, etc.
`jetty-request-[date].log`	This file is only located on the CLC component and tracks access information (credentials) associated with users and their accounts.
`/var/log/messages`	This file contains any general host problems. For example: networking issues, disk space, hardware failures.

CC Log Files

For the Cluster controller (CC), the general types of errors to look for are errors with node orchestration, communication issues between CC and NCs, and tunneling issues with multi-cluster configurations (to span security groups across AZs). Log files are as follows:

Log File	Description
`cc.log`	This is the canonical place for CC error messages, and is the common log for all info and warning messages as well. In the C code, we mostly follow syslog practices. You can change the CC logging level on the fly with a restart. Log messages tend to be readable and informative.
`cc-fault.log`	This file contains issues with known error codes and known resolutions.
`axis2c.log`	This file is for the web services stack on the CC. Web services calls get translated here. It is not too "user-friendly" for parsing, but you normally do not need to go through it. Most issues appearing in this file have to do with credential errors or OpenSSL issues.
`httpd-cc_error_log`	This file generally contains information about events around the web services stack. For example: component start or stop, IP tables, or networking errors.
`/var/log/messages`	This file contains DHCP bridge issues and general network-related issues.

NC Log Files

For the Node controller (NC) log files generally detail instance tasks, instance lifecycle, and instance operations. NC log files are as follows:

Log File	Description
`nc.log`	This file is the common file for all info, error, and warning messages. It is a good starting place for all issues on an NC.
`axis2c.log`	This file is for the web services stack on the NC. Web services calls get translated here. It is not too "user-friendly" for parsing, but you normally do not need to go through it. Most issues appearing in this file have to do with credential errors or OpenSSL issues.
`httpd-nc_error_log`	This file generally contains information about events around the web services stack. For example: component start or stop, IP tables, or networking errors.
`nc-fault.log`	This file contains issues that have known error codes and known resolutions.
`euca_test_nc.log`	When the NC starts up, it runs through a self-test. This file contains log message from that process. It is useful to review if you have a fresh NC and you're seeing issues.
`/var/log/messages`	This file contains high-level KVM, libvirt, and general hypervisor issues. It also contains iSCSI/EBS issues (usually connecting instances to storage), and networking issues in certain Eucalyptus networking modes (most useful in Edge, Managed, and Managed-NoVLAN).
`/var/log/libvirt/*`	Various low-level libvirt errors and low-level QEMU and KVM errors.

API Services Running As Instances

The following log files are relevant to cloud administrators who have access to instances directly:

Log File	Description
`worker.log`	Logs on the Image Worker image are stored in `/var/log/eucalyptus-imaging-worker.`
`servo.log`	Logs on a given Elastic Load Balancer (ELB) are stored in `/var/log/load-balancer-servo.`

System Log Files

You might also find helpful information about the nature of an issue in the system logs. In particular, the following logs may be relevant:

- `/var/log/messages`
- `/var/log/libvirt/`

Log File Levels

All messages that show up as `FAULT`, `FATAL`, or `ERROR` require an action by the administrator.

FAULT Anything identified in a fault log has an identifiable cause and an identifiable solution, but one that Eucalyptus cannot fix by itself. The administrator needs to act immediately.

FATAL Any condition that indicates that Eucalyptus has failed (for example, OOM).

ERROR Any condition for which an operator must take immediate action to identify and/or remedy.

WARN An indication that the system could not perform a task, but does not necessarily indicate that immediate action by the operator is required. For example, when a user tries to allocate a bucket when their quota is exceeded, or when an action is being retried unsuccessfully, with the final timeout possibly giving ERROR instead of WARN.

INFO This is the default recommended log level. Any log message that contains useful information to see “what is happening” and generally indicates healthy activity. For example, anytime a user runs `euca-describe-instances` (that is, User A does Action B at Time T with Correlation-id I, and it succeeded or failed--grep for Correlation-id I in various logs for more info). This is useful for troubleshooting, but not necessarily for monitoring.

DEBUG Detailed debug data is only available when the cloud is set to debug mode, and unlike INFO, it does not seek to aggregate messages. Instead, it writes them out the second they’re generated. For example, entering or leaving a particular function. These messages are generally incomprehensible to administrators, but are useful to Eucalyptus engineers for debugging.

TRACE (backend) or EXTREME (frontend) These are useful for engineers only in development. Unlike DEBUG, installations running in TRACE or EXTREME mode can actually degrade the system as a result of the monitoring activity, and could actually create failures. We recommend that you don't

Log File Configuration

For the CC and the NC, you can configure the log level using the `LOGLEVEL` parameter in `eucalyptus.conf`. This parameter will be picked up dynamically when the value is changed in the config file, without requiring a restart of the component.

For all other components, you can configure the log level by passing an appropriate `--log-level` argument in the init script. You can also dynamically change the level using `euca-modify-property` and set an appropriate value for `cloud.euca_log_level`. This takes precedence over the value specified in the init script.

Valid log levels are as follows, from most to least verbose:

- ALL
- EXTREME
- TRACE
- DEBUG
- INFO
- WARN
- ERROR
- FATAL
- OFF

If no value is specified, the default `INFO` is used.

Log File Format

Eucalyptus logs now have a standard format, which varies slightly per log level.

For log levels FATAL, ERROR, WARN and INFO:

```
YYYY-MM-DD HH:MM:SS LEVEL | message
```

For log levels DEBUG and TRACE:

```
YYYY-MM-DD HH:MM:SS LEVEL PROCESS:THREAD loggingMethodOrClass     | message
```

For log level EXTREME and ALL:

```
YYYY-MM-DD HH:MM:SS LEVEL PROCESS:THREAD loggingMethodOrClass
FILENAME:LineNumber | message
```

Fault Logs

Eucalyptus includes fault logs for easy identification of conditions outside of Eucalyptus's control that may cause it to fail. These messages are logged per component, and each fault is logged only once per component, in `/var/log/eucalyptus/[component]-fault.log`. The messages include a suggested resolution, and can be customized. Where they have been translated, Eucalyptus will use the system-configured `LOCALE` variable to serve appropriate messages.

Fault messages are based on XML-formatted templates, stored in a per-locale directory structure, with one file per fault message, and one file storing common strings. Default templates are shipped with Eucalyptus. These are stored in `/usr/share/eucalyptus/faults/` as follows:

```
/usr/share/eucalyptus/faults/en_US/0001.xml
...
/usr/share/eucalyptus/faults/en_US/1234.xml
/usr/share/eucalyptus/faults/en_US/common.xml
```

Using Localized Fault Logs

Localized messages are located in a per-locale directory under `/usr/share/eucalyptus/faults/`. If localized messages are available matching the system LOCALE, Eucalyptus will use those messages. If no LOCALE is set, Eucalyptus defaults to `en_US`.

Set the system LOCALE in `/etc/sysconfig/i18n` as follows:

```
LOCALE=ru_RU
```

Using Customized Fault Logs

To use your own customized messages, copy the message files to the appropriate directory under `/etc/eucalyptus/faults/` and edit them. Do not change the filenames. To test the fault template, run `euca-generate-fault`, providing the component name, fault ID, and any relevant parameters along with their values.

```
euca-generate-fault -c component_name fault_id [param] [value]
```

For example

```
euca-generate-fault -c nc 1008 daemon ntpd
```

The test fault should be logged in the appropriate component fault log (in this case, `/var/log/eucalyptus/nc-fault.log`

Eucalyptus uses customized messages where they are available, preferring a non-localized custom message over a localized default message. Localized messages should be in a per-locale directory under `/etc/eucalyptus/faults/`, with a directory name that matches the system LOCALE. If no LOCALE is set, Eucalyptus defaults to `en_US`.

Network Information

When you have to troubleshoot, it's important to understand the elements of the network on your system.

Here are some ideas for finding out information about your network:

- It is also important to understand the elements of the network on your system. For example, you might want to list bridges to see which devices are enslaved by the bridge. To do this, use the `brctl` command.
- You might also want to list network devices and evaluate existing configurations. To do this, use these commands: `ip`, `ifconfig`, and `route`.
- If you are running Eucalyptus in Managed networking mode, you can also use `vconfig` to evaluate VLAN configuration.
- You can get further information if you use the `euca-describe` commands with the `verbose` options. For example, `euca-describe-instances verbose` returns all instances running by all users on the system. Other describe commands are:

- `euca-describe-volumes verbose`
- `euca-describe-snapshots verbose`
- `euca-describe-groups verbose`
- `euca-describe-keypairs verbose`

Common Problems

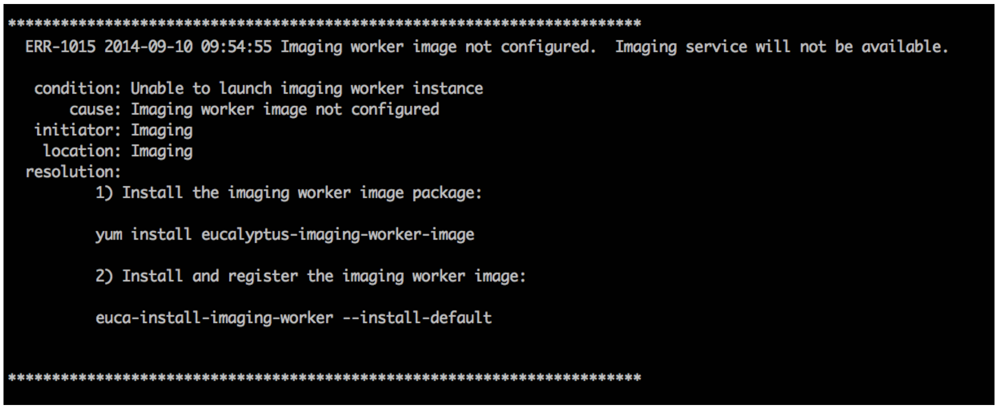
This section describes common problems and workarounds.

Problem: install-time checks

Eucalyptus offers installation checks for any Eucalyptus component or service (CLC, Walrus, SC, NC, SC, services, and more). When Eucalyptus encounters an error, it presents the problem to the operator. These checks are used for install-time problems. They provide resolutions to some of the fault conditions.

Each problematic condition contains the following information:

Heading	Description
Condition	The fault found by Eucalyptus
Cause	The cause of the condition
Initiator	What is at fault
Location	Where to go to fix the fault
Resolution	The steps to take to resolve the fault



```
************************************************************************
  ERR-1015 2014-09-10 09:54:55 Imaging worker image not configured.  Imaging service will not be available.

   condition: Unable to launch imaging worker instance
       cause: Imaging worker image not configured
   initiator: Imaging
    location: Imaging
  resolution:
          1) Install the imaging worker image package:

          yum install eucalyptus-imaging-worker-image

          2) Install and register the imaging worker image:

          euca-install-imaging-worker --install-default


************************************************************************
```

For more information about all the faults we support, go to *https://github.com/eucalyptus/eucalyptus/tree/master/util/faults/en_US*.

Problem: instance runs but fails

Run `euca-describe-nodes` to verify if instance is there. Is the instance there?

- Yes:

  a) Go to the *NC log* for that NC and grep your instance ID. Did you find the instance?

  - Yes:

    Is there an error message?

    - Yes:

      This clues you in to some helpful information

        - No:

          Go to *CC log* and grep the instance ID.

  b) No:

     Go to the *CC log* and grep the instance ID. Is it there error message?

     - Yes:

       The error message should give you some helpful information.

     - No:

       grep the instance ID in *cloud-output.log*. Is there error message?

       - Yes:

         The error message should give you some helpful information.

       - No:

         grep volume ID in *SC log*.

- No:

  Log in as admin and run `euca-describe-instance`. Is the instance there?

  - Yes:

    - Note your AZ.
    - Run `euca-describe-az verbose`.
    - Note the CC IP
    - Go to the *CC log* and grep the instance ID.

  - No:

    Start over and run a new instance, recreate failure, and start these steps over.

Problem: can't communicate with instance

Use ping from a client (not the CLC). Can you ping it?

- Yes:

  Check the open ports on security groups and retry connection using SSH or HTTP. Can you connect now?

  a) Yes. Okay, then. You're work is done.

  b) No:

     Try the same procedure as if you can't ping it up front.

- No:

  Is your cloud running in Edge networking mode?

  - Yes:

    Run `euca-describe-nodes`. Is your instance there?

    - Yes:

      Ping the instance's public IP from the NC. Can you ping it? Check network between client and NC (this indicates that the problem is not the Eucalyptus network).

    - No:

Check `eucanetd.log` and IP tables rules. Make sure the IP address has visible public IPs and that the IP tables have expected ports opened.

- No, it is not in Edge networking mode:
  1. Run `euca-describe-instances`
  2. Note the AZ name.
  3. Run `euca-describe-AZ verbose`.
  4. Note the IP for the CC.
  5. Ping the instance's private IP from the CC.

     Are there error messages?

     - Yes:

       Check the network connection between the client and the CC.
     - No:

       Check `eucanetd.log` and the IP tables rules. Make sure the IP address has visible public IPs and that the IP tables have expected ports opened.

Problem: volume creation failed

Symptom: Went from available to fail. This is typically caused by the CLC and the SC.

On the SC, use `df` or `lvdisplay` to check the disk space. Is there enough space?

- Yes:

  Check the *SC log* and grep the volume ID. Is there error message?

  a) Yes. This provides clues to helpful information.

  b) No:

  Check *cloud-output.log* a volume ID error.
- No:

  Delete volumes or add disk space.

Problem: snapshot creation failed

On the SC, use `df` or `lvdisplay` to check the disk space in `var/lib/eucalyptus/volumes`. Is there enough space?

- Yes:

  Use `df` or `lvdisplay` to check the disk space in `var/lib/eucalyptus/bukkits`. Is there enough space?

  a) Yes.

  - Use `euca-describe-services` and note the IP addresses for the OSG and SC.
  - SSH to SC and ping the OSG.

    Are there error messages?

    - Yes:

      Check *the SC and the OSG logs* for the snapshot ID.
    - No:

      Check the network connection between the SC and the OSG.

  b) No:

Delete volumes or add disk space.

- No:

  Delete volumes or add disk space.

Component Workarounds

This section contains troubleshooting information for Eucalyptus components and services.

Walrus and Storage

This topic contains information about Walrus-related problems and solutions.

Walrus decryption failed.

On Ubuntu 10.04 LTS, kernel version 2.6.32-31 includes a bug that prevents Walrus from decrypting images. This can be determined from the following line in cloud-output.log

```
javax.crypto.
BadPaddingException: pad block corrupted
```

If you are running this kernel:

1. Update to kernel version 2.6.32-33 or higher.
2. De-register the failed image (`euca-deregister`).
3. Re-register the bundle that you uploaded (`euca-register <bucket>/<manifest>`).

Walrus physical disk is not large enough.

1. Stop the CLC.
2. Add a disk.
3. Migrate your data.

Make sure you use LVM with your new disk drive(s).

Access and Identities

This topic contains information about access-related problems and solutions.

Need to verify an existing LIC file.

1. Enter the following command:

   ```
   /usr/sbin/euca-describe-properties | grep ldap
   ```

   The output from the example above shows the name of the LIC file and status of the synchronization (set to false).

   ```
   PROPERTY  authentication.ldap_integration_configuration
   { 'sync': { 'enable':'false' } }
   ```

Windows Images

This topic contains information to help you troubleshoot your Windows images.

Properties

A typical size of Windows images is large and Eucalyptus has a set of properties that limit the size of various storage components. The first step in troubleshooting is to make sure that the values are large enough to store your Windows images. You can modify a property using

```
/usb/sbin/euca-modify-property -p <property>=<value>
```

The properties that might affect registering Windows images are:

- `walrus.storagemaxbucketsizeinmb`: max bucket size enforced by Walrus; should be larger than a Windows image
- `walrus.storagemaxcachesizeinmb`: total size of all images that is cached in Walrus; should be larger than the sum of all images (Windows/Linux) in Walrus

- walrus.storagemaxtotalsnapshotsizeingb: if a Windows image is a type of EBS-backed EMI, this should be large enough to store all EBS backed images
- `{PARTITION}.storage.maxvolumesizeingb`: if a Windows image is a type of EBS-backed EMI, this should be large enough to store the image

In addition, during the `euca-run-instances`, the CLC may time-out an instance while a large windows image (images in both Walrus and EBS) is being launched. We recommend that you raise the values of the following properties.

- `cloud.vmstate.instance_timeout`: maximum wait time, in minutes, before the instance becomes running. Am instance cannot stay in pending longer than this. Default: 60
- `cloud.vmstate.ebs_volume_creation_timeout`: maximum wait time, in minutes, before a volume backing a boot from EBS image is created. Default: 30
- `cloud.addresses.maxkillorphans`: The public IP assigned to an instance will be expired after the time limit. The exact time-out is {maxkillorphans*5} seconds (by default it's 50 seconds). If the volume backing an EBS image is not created in time, the public IP will be released from the instance.

Image Preparation

`euca-bundle-image` hangs

The time to bundle an image is proportional to the image size. Because the typical size of Windows image is big, give enough time until bundling is complete. As a rule of thumb, it may take up to 20 min. for bundling a 10 GB Windows image.

`euca-upload-bundle` fails

Make sure ‘walrus.storagemaxbucketsizeinmb’ is large enough. If not, ask your administrator.

Instance Launch and Login

Instance stays in pending

Typically, it takes longer to launch Windows images than Linux images as the delay is proportional to the image size. This can be especially long when the image is seeded on NCs the first time (images are cached in NCs and run within few seconds thereafter). As a rule of thumb, 10 GB Windows images may take up to 10 minutes to become ‘running’ when it is not cached in NCs.

Instance stay in pending and goes to shutdown

An instance may time-out if the Windows image is too big. Review and adjust the relevant properties.

Instance is running, but not accessible using Remote Desktop.

after instances become running, you should wait until Windows is fully booted. If the image is sysprepped, the booting time may take up to 10 min. Also you should make sure the followings are cleared:

- The port 3389 is authorized in the security group
- If the instance is attached to your active directory domain, the domain GPO shouldn’t block the RDP port (3389)
- The username should be authorized to log-in using Remote Desktop (refer to User guide: Windows integration service)

Finding the login username and password

Use `Administrator` and the password retrieved by `euca-get-password`. If the instance is attached to a domain, you may use your domain username and password (make sure the username is prepended with domain name, such as `YOUR_DOMAIN\Alice`).

Can’t retrieve windows password using `euca-get-password`

Make sure the platform field of your windows EMI is set to ‘windows’, not ‘linux’ (use `euca-describe-images`). If not, the most likely reason is that the image name does not begin with ‘windows’. You should bundle/upload/register the image with a proper name.

Instance is not attached to an Active Directory domain

- Make sure the parameters set in Windows integration service are correct. One way to verify them is to log in the instance using Administrator password and manually attach the instance to the domain (System Properties -> Computer Name) using the same parameters.

- Make sure `VNET_DNS` in eucalyptus.conf is set to the domain controller (refer to User Guide: Configure Active Directory).

Disk and Volume

Ephemeral disks are not visible in the Windows

Open Disk Management console (**All Programs**->**Administrative Tools**->**Server Manager**->**Storage**) and find the uninitialized disks. You should create a partition on the disk and format it.

EBS volume is attached, but not visible in the Windows

Open Disk Management console (**All Programs**->**Administrative Tools**->**Server Manager**->**Storage**) and find the uninitialized disks. You should create a partition on the disk and format it. You don't have to repeat it when the volume is reattached later.

EBS volume is detached, but the disk drive (for example, `E:\`) is still visible in the Windows

For KVM hypervisor, you should perform "remove hardware safely" before detaching the volume.

`euca-bundle-instance` fails

Make sure the bucket specified with '-b' option doesn't already exist and the property 'walrus.storagemaxbucketsizeinmb' is large enough to store the image.

Instances

This topic contains information to help you troubleshoot your instances.

Inaccurate IP addresses display in the output of euca-describe-addresses.

This can occur if you add IPs from the wrong subnet into your public IP pool, do a restart on the CC, swap out the wrong ones for the right ones, and do another restart on the CC. To resolve this issue, run the following commands.

Note: A restart should only be performed when no instances are running, or when instance service interruption can be tolerated. A restart causes the CC to reset its networking configuration, regardless of whether or not it is in use. A restart of a CC in Managed and Managed (NoVLAN) modes that is managing active VMs can cause a temporary loss of network connectivity until the CC relearns the network topology and rebuilds the IP table entries.

```
/etc/init.d/eucalyptus-cloud stop
/etc/init.d/eucalyptus-cc stop
iptables -F
/etc/init.d/eucalyptus-cc restart
/etc/init.d/eucalyptus-cloud start
```

NC does not recalculate disk size correctly

This can occur when trying to add extra disk space for instance ephemeral storage. To resolve this, you need to delete the instance cache and restart the NC.

For example:

```
rm -rf /var/lib/eucalyptus/instances/*
service eucalyptus-nc restart
```

Elastic Load Balancing

This topic explains suggestions for problems you might have with Elastic Load Balancing (ELB).

Can't synchronize with time server

Eucalyptus sets up NTP automatically for any instance that has an internet connection to a public network. If an instance doesn't have such a connection, set the cloud property `loadbalancing.loadbalancer_vm_ntp_server` to a valid NTP server IP address. For example:

```
euca-modify-property -p
loadbalancing.loadbalancer_vm_ntp_server=169.254.169.254
```

```
PROPERTY loadbalancing.loadbalancer_vm_ntp_server
169.254.169.254 was {}
```

Need to debug an ELB instance

To debug an ELB instance, set the `loadbalancing.loadbalancer_vm_keyname` cloud property to the keypair of the instance you want to debug. For example:

```
# euca-modify-property -p
loadbalancing.loadbalancer_vm_keyname=sshlogin
PROPERTY loadbalancing.loadbalancer_vm_keyname sshlogin was
{}
```

High Availability

This topic contains information to help you troubleshoot your high availability deployment.

In the event that incorrect keys for a secondary CLC are used, Eucalyptus behaves as if that CLC no longer exists. The current primary CLC continues to operate as expected. In order to bring back the secondary CLC, perform the following tasks.

1. Stop the secondary CLC.

   ```
   service eucalyptus-cloud stop
   ```

2. On the secondary CLC, delete all files from `/var/lib/eucalyptus/db`.
3. On the secondary CLC, delete all .pem and vtunpass files from `var/lib/eucalyptus/keys`.
4. Start the secondary CLC.

   ```
   service eucalyptus-cloud start
   ```

5. Re-register the secondary CLC with the primary CLC.

   ```
   /usr/sbin/euca_conf --register-cloud --partition eucalyptus
   --host [Secondary_CLC_IP] --component [CLC_Name]
   ```

Imaging Worker

This topic contains troubleshooting tips for the Imaging Worker.

Some requests that require the Imaging Worker might remain in pending for a long time. For example: an import task or a paravirtual instance run. If request remains in pending, the Imaging Worker instance might not able to run because of a lack of resources (for example, instance slots or IP addresses).

You can check for this scenario by listing latest AutoScaling activities:

```
euscale-describe-scaling-activities -g asg-euca-internal-imaging-worker-01
```

Check for failures that indicate inadequate resources such as:

```
ACTIVITY       1950c4e5-0db9-4b80-ad3b-5c7c59d9c82e    2014-08-12T21:05:32.699Z
        asg-euca-internal-imaging-worker-01    Failed   Not enough resources
available: addresses; please stop or terminate unwanted instances or release
unassociated elastic IPs and try again, or run with private addressing only
```

Manage Users and Groups

You can also perform user authentication by integrating Eucalyptus with an existing LDAP or Active Directory. This information cannot be changed from Eucalyptus side when LDAP/AD integration is turned on. However, other Eucalyptus-specific information about user, group and account is still stored within the local database of Eucalyptus, including certificates, secret keys and attached policies.

For more information about synchronizing an existing LDAP or Active Directory with Eucalyptus, see *LDAP/AD Integration*.

Access Overview

The Eucalyptus design of user identity and access management provides layers in the organization of user identities. This gives you refined control over resource access. Though compatible with the AWS IAM, there are also a few Eucalyptus-specific extensions that meet the needs of enterprise customers.

Access Concepts

This section describes what Eucalyptus access is and what you need to know about it so that you can configure access to your cloud.

User Identities

In Eucalyptus, user identities are organized into accounts. An account is the unit of resource usage accounting, and also a separate namespace for many resources (security groups, key pairs, users, etc.).

Accounts are identified by either a unique ID (UUID) or a unique name. The account name is like IAM’s account alias. It is used to manipulate accounts. However, for AWS compatibility, the EC2 commands often use account ID to display resource ownership.

There are command line tools to discover the correspondence of account ID and account name. For example, euare-accountlist lists all the accounts with both their IDs and names.

An account can have multiple users, but a user can only be in one account. Within an account, users can be associated with Groups. Group is used to attach access permissions to multiple users. A user can be associated with multiple groups. Because an account is a separate name space, user names and group names have to be unique only within an account. Therefore, user X in account A and user X in account B are two different identities.

Both users and groups are identified by their names, which are unique within an account (they also have UUIDs, but are rarely used).

Special Identities

Eucalyptus has two special identities for the convenience of administration and use of the system.

- The **eucalyptus** account: Each user in the eucalyptus account has unrestricted access to all of the cloud's resources, similar to the superuser on a typical Linux system. These users are often referred to as system administrators or cloud administrators. This account is automatically created when the system starts for the first time. You cannot remove the eucalyptus account from the system.
- The **admin** user of an account: Each account, including the eucalyptus account, has a user named admin. This user is created automatically by the system when an account is created. The admin of an account has full access to the resources owned by the account. You can not remove the admin user from an account. The admin can delegate resource access to other users in the account by using policies.

Credentials

This topic describes the different types of credentials used by Eucalyptus.

Each user has a unique set of credentials. These credentials are used to authenticate access to resources. There are three types of credentials:

- An **X.509 certificate** is used to authenticate requests to the SOAP API service.
- A **secret access key** is used to authenticate requests to the REST API service.

You can manage credentials using the command line tools (the `euare-` commands). For more information about the command line tools, see the *Euca2ools Reference Guide*.

In IAM, each account has its own credentials. In Eucalyptus, the equivalent of account credentials are the credentials of admin user of the account.

You can download the full set of credentials for a user or an account, including X509 certificate and secret access key, by:

```
/usr/sbin/euca_conf --get-credentials
```

or:

```
euca-get-credentials
```

Eucalyptus returns the following:

- An arbitrary existing active secret access key
- A newly generated X509 certificate

Account Creation

This topic describes the process for creating an account.

You can create accounts using the command line tool. You must be a cloud administrator to use this command. Accounts created are available for immediate access.

To create an account, run the following command:

```
euare-accountcreate -a account_name
```

To get the account registration status, run the following command:

```
euare-usergetattributes --as-account account_name -u admin --show-extra
```

Where the `--show-extra` option displays extra information of a user in the following order:

- Enabled status
- Registration status
- Password expires

The account registration status has the following values based on the status of registration process: `REGISTERED`, `APPROVED`, or `CONFIRMED`. An account that is not confirmed cannot be used or accessed. The system administrator can manipulate the account registration status in by running the following command:

```
euare-usermod --as-account account_name -u admin --reg-status=status
```

The command line manipulation of the registration status does not send the notification emails.

Special User Attributes

Eucalyptus extends the IAM model by providing the following extra attributes for a user.

- **Registration status:** This is only meaningful for the account administrator (that is, the account level).
- **Enabled status:** . Use this attribute to temporarily disable a user.
- **Password expiration date**
- **Custom information:** Add any name-value pair to a user's custom information attribute. This is useful for attaching pure text information, like an address, phone number, or department. This is especially helpful with external LDAP or Active Directory services.

You can retrieve and modify the registration status, enabled status, and password expiration date using the `euare-usergetattributes` and `euare-usermod` commands. You can retrieve and modify custom information using `euare-usergetinfo` and `euare-userupdateinfo` commands. For more information, see the *Euca2ools Reference Guide* for details about these commands.

Roles

A *role* A role is a mechanism that enables the delegation of access to users or applications.

A role is associated with an account, and has a set of permissions associated with it that are defined in the form of an IAM *policy*. A policy specifies a set of actions and resources within that account that the role is allowed to access.

Note: For more information on policies, see *policies*Policy Overview.

By assuming a role, a user or an applications gets a set of permissions associated with that role. When a role is assumed, the Eucalyptus STS service returns a set of temporary security credentials that can then be used to make programmatic requests to resources in your account. This eliminates the need to share or hardcode security credentials with applications that need access to resources in your cloud.

Eucalyptus roles are managed through the Eucalyptus Euare service, which is compatible with Amazon's Identity and Access Management service. For more information on IAM and roles, please see the *Amazon IAM User Guide*.

Usage Scenarios for Roles

There are several scenarios in which roles can be useful, including:

Applications

Applications running on instances in your Eucalyptus cloud will often need access to other resources in your cloud. Instead of creating AWS credentials for each application, or distributing your own credentials,, you can use roles to enable you to delegate permission to the application to make API requests. For more information, see *Launch an Instance with a Role*.

Account Delegation

You can use roles to allow one account to access resources owned by another account. IAM Roles under the 'eucalyptus' account can be assumed by users under 'non-eucalyptus' account(s). For example, if you had an 'infrastructure auditing' account, and an audit was needed to be performed on all the cloud resources used on the cloud, a user could assume the 'Resource Administrator' role and audit all the resources used by all the accounts on the cloud. For more information on IAM account delegation, see *Using Roles to Delegate Permissions and Federate Identities*. Also, go to the walkthrough provided in the*AWS Identity and Access Management* section of the AWS documentation.

Pre-Defined Roles

Eucalyptus offers a number of pre-defined privileged roles. These roles are associated with the `eucalyptus` account, and have privileges to manage resources across the cloud and non-privileged accounts. Only the eucalyptus account can manage or modify these roles.

To see the pre-defined roles, use `euare-rolelistbypath` with the credentials of a user that is part of the `eucalyptus` account. For example:

```
# euare-rolelistbypath
arn:aws:iam::944786667073:role/eucalyptus/AccountAdministrator
arn:aws:iam::944786667073:role/eucalyptus/InfrastructureAdministrator
arn:aws:iam::944786667073:role/eucalyptus/ResourceAdministrator
```

Account Administrator

The Account Administrator (AA) can manage Eucalyptus accounts. To view the policy associated with the Account Administrator role, use `euare-rolelistpolicies` with the credentials of a user that is part of the `eucalyptus` account. For example:

```
# euare-rolelistpolicies --role-name AccountAdministrator --verbose
AccountAdministrator
{
  "Statement":[ {
    "Effect": "Allow",
    "Action": [
      "iam:*"
    ],
    "NotResource": "arn:aws:iam::eucalyptus:*",
    "Condition": {
      "Bool": { "iam:SystemAccount": "false" }
    }
  } ]
}
IsTruncated: false
```

Resource Administrator

The Resource Administrator (RA) can manage AWS-defined resources (such as S3 objects, instances, users, etc) across accounts. To view the policy associated with the Resource Administrator role, use `euare-rolelistpolicies` with the credentials of a user that is part of the `eucalyptus` account. For example:

```
# euare-rolelistpolicies --role-name ResourceAdministrator --verbose
ResourceAdministrator
{
  "Statement":[ {
    "Effect": "Allow",
    "Action": [
      "autoscaling:*",
      "cloudwatch:*",
      "ec2:DescribeInstanceAttribute",
      "ec2:DescribeInstances",
      "ec2:DescribeInstanceStatus",
      "ec2:DescribeInstanceTypes",
      "ec2:GetConsoleOutput",
      "ec2:GetPasswordData",
      "ec2:ImportInstance",
      "ec2:ModifyInstanceAttribute",
      "ec2:MonitorInstances",
      "ec2:RebootInstances",
      "ec2:ReportInstanceStatus",
      "ec2:ResetInstanceAttribute",
      "ec2:RunInstances",
      "ec2:StartInstances",
      "ec2:StopInstances",
      "ec2:TerminateInstances",
      "ec2:UnmonitorInstances",
      "ec2:*AccountAttributes*",
      "ec2:*Address*",
      "ec2:*AvailabilityZones*",
      "ec2:*Bundle*",
      "ec2:*ConversionTask*",
      "ec2:*CustomerGateway*",
      "ec2:*DhcpOptions*",
      "ec2:*ExportTask*",
      "ec2:*Image*",
      "ec2:*InternetGateway*",
      "ec2:*KeyPair*",
      "ec2:*NetworkAcl*",
      "ec2:*NetworkInterface*",
      "ec2:*PlacementGroup*",
      "ec2:*ProductInstance*",
      "ec2:*Region*",
      "ec2:*ReservedInstance*",
      "ec2:*Route*",
      "ec2:*SecurityGroup*",
      "ec2:*Snapshot*",
      "ec2:*SpotDatafeedSubscription*",
      "ec2:*SpotInstance*",
      "ec2:*SpotPrice*",
      "ec2:*Subnet*",
      "ec2:*Tag*",
      "ec2:*Volume*",
      "ec2:*Vpc*",
      "ec2:*Vpn*",
      "ec2:*VpnGateway*",
      "elasticloadbalancing:*",
      "s3:*"
    ],
    "Resource": "*"
  }, {
    "Effect": "Allow",
    "Action": [
      "iam:Get*",
      "iam:List*"
    ],
    "NotResource": "arn:aws:iam::eucalyptus:*"
  } ]
}
IsTruncated: false
```

Infrastructure Administrator

The Infrastructre Administrator (IA) can perform operations related to cloud setup and management. Typical responibilities include:

- Installation and configuration (prepare environment, install Eucalyptus, configure Eucalyptus)
- Monitoring and maintenance (infrastructure supporting the cloud, cloud management layer, upgrades, security patches, diagnostics and troubleshooting)
- Backup and restoration

To view the policy associated with the Infrastructure Administrator role, use `euare-rolelistpolicies` with the credentials of a user that is part of the `eucalyptus` account. For example:

```
# euare-rolelistpolicies --role-name InfrastructureAdministrator --verbose
InfrastructureAdministrator
{
  "Statement":[ {
    "Effect": "Allow",
    "Action": [
      "euprop:*",
      "euserv:*",
      "euconfig:*",
      "ec2:MigrateInstances"
    ],
    "Resource": "*"
  } ]
}
IsTruncated: false
```

Policy Overview

Eucalyptus uses the policy language to specify user level permissions as AWS IAM. Policies are written in JSON. Each policy file can contain multiple statements, each specifying a permission.

A permission statement specifies whether to allow or deny a list of actions to be performed on a list of resources, under specific conditions.

A permission statement has the following components:

- **Effect:** Begins the decision that applies to all following components. Either: `“Allow”` or `“Deny”`
- **Action or NotAction:** Indicates service-specific and case-sensitive commands. For example: `“ec2:RunInstances”`

- **Resource or NotResource:** Indicates selected resources, each specified as an Amazon resource name (ARN). For example: `“arn:aws:s3:::acme_bucket/blob”`
- **Condition:** Indicates additional constraints of the permission. For example: `"DateGreaterThan"`

The following policy example contains a statement that gives a user with full permission. This is the same access as the account administrator:

```
{
  "Version":"2011-04-01",
  "Statement":[{
    "Sid":"1",
    "Effect":"Allow",
    "Action":"*",
    "Resource":"*"
  }]
}
```

For more information about policy language, go to the *IAM User Guide*.

Policy Notes

You can combine IAM policies with account level permissions. For example, the admin of account A can give users in account B permission to launch one of account A’s images by changing the image attributes. Then the admin of account B can use IAM policy to designate the users who can actually use the shared images.

You can attach IAM policies to both users and groups. When attached to groups, a policy is equivalent to attaching the same policy to the users within that group. Therefore, a user might have multiple policies attached, both policies attached to the user, and policies attached to the group that the user belongs to.

Do not attach IAM policies (except quota policies, a Eucalyptus extension) to account admins. At this point, doing so won’t result in a failure but may have unexpected consequences.

Policy Extensions

Eucalyptus extends the IAM policy in order to meet the needs of enterprise customers.

EC2 Resource

In IAM, you cannot specify EC2 resources in a policy statement except a wildcard,`“*”`. So, you can't restrict a permission to specific EC2 entities. For example, you can't restrict a user to run instances on a specific image or VM type. To solve that, Eucalyptus created the EC2 resource for the policy language. The following example shows the ARN of an EC2 resource.

```
arn:aws:ec2::<account_id>:<resource_type>/<resource_id>
```

Where account id is optional.

Eucalyptus supports the following resource types for EC2:

- image
- securitygroup
- address (either an address or address range: 192.168.7.1-192.168.7.255)
- availabilityzone
- instance
- keypair
- volume
- snapshot
- vmtype

The following example specifies permission to launch instances with only an m1.small VM type:

```
{
  "Version":"2011-04-01",
  "Statement":[{
    "Sid":"2",
    "Effect":"Allow",
    "Action":"ec2:RunInstances",

    "Resource": [
    "arn:aws:ec2:::vmtype/m1.small",
    "arn:aws:ec2:::image/*",
    "arn:aws:ec2:::securitygroup/*",
    "arn:aws:ec2:::keypair/*",
    "arn:aws:ec2:::availabilityzone/*",
    "arn:aws:ec2:::instance/*"
]

  }]
}
```

Policy Key

Eucalyptus implements the following AWS policy keys:

- aws:CurrentTime
- aws:SourceIp

Eucalyptus extends the policy keys by adding the following to the lifetime of an instance:

- ec2:KeepAlive: specifies the length of time (in minutes) that an instance can run
- ec2:ExpirationTime: specifies the expiration time (in minutes) for an instance

The following example restricts an instance running time to 24 hours:

```
{
  "Version":"2011-04-01",
  "Statement":[{
    "Sid":"3",
    "Effect":"Allow",
    "Action":"ec2:RunInstances",
    "Resource":"*",
    "Condition":{
      “NumericEquals”:{
        “ec2:KeepAlive”:”1440”
      }
    }
  }]
}
```

If there are multiple `ec2:KeepAlive` or `ec2:ExpirationTime` keys that match a request, Eucalyptus chooses the longer lifetime for the instance to run.

Default Permissions

Different identities have different default access permissions. When no policy is associated with them, these identities have the permission listed in the following table.

Identity	Permission
System admin	Access to all resources in the system
Account admin	Access to all account resources, including those shared resources from other accounts like public images and shared snapshots
Regular user	No access to any resource

Quotas

Eucalyptus adds quota enforcement to resource usage. To avoid introducing another configuration language into Eucalyptus, and simplify the management, we extend the IAM policy language to support quotas.

The only addition added to the language is the new `limit` effect. If a policy statement’s `effect` is `limit`, it is a quota statement.

A quota statement also has action and resource fields. You can use these fields to match specific requests, for example, quota only being checked on matched requests. The actual quota type and value are specified using special quota keys, and listed in the `condition` part of the statement. Only condition type `NumericLessThanEquals` can be used with quota keys.

Important: An account can only have a quota policy. Accounts can only accept IAM policies where Effect is "Deny" or "Limit". If you attach an IAM policy to an account where the Effect is "Allow", you will get unexpected results.

The following quota policy statement limits the attached user to only launch a maximum of 16 instances in an account.

```
{
 "Version":"2011-04-01",
 "Statement":[{
   "Sid":"4",
   "Effect":"Limit",
   "Action":"ec2:RunInstances",
   "Resource":"*",
   "Condition":{
     “NumericLessThanEquals”:{
       “ec2:quota-vminstancenumber”:”16”
     }
   }
 }]
}
```

You can attach quotas to both users and accounts, although some of the quotas only apply to accounts. Quota attached to groups will take no effect.

When a quota policy is attached to an account, it actually is attached to the account administrator user. Since only system administrator can specify account quotas, the account administrator can only inspect quotas but can't change the quotas attached to herself.

The following is all the quota keys implemented in Eucalyptus:

Quota Key	Description	Applies to
`s3:quota-bucketnumber`	Number of S3 buckets	account and user
`s3:quota-bucketobjectnumber`	Number of objects in each bucket	account and user
`s3:quota-bucketsize`	Size of bucket, in MB	account and user
`s3:quota-buckettotalsize`	total size of all buckets, in MB	account and user
`ec2:quota-addressnumber`	Number of elastic IPs	account and user
`ec2:quota-imagenumber`	Number of EC2 images	account and user
`ec2:quota-securitygroupnumber`	Number of EC2 security groups	account and user
`ec2:quota-snapshotnumber`	Number of EC2 snapshots	account and user
`ec2:quota-vminstancenumber`	Number of EC2 instances	account and user
`ec2:quota-volumenumber`	Number of EC2 volumes	account and user
`ec2:quota-volumetotalsize`	Number of total volume size, in GB	account and user

Quota Key	Description	Applies to
`cloudformation:quota-stacknumber`	Number of Cloudformation stacks allowed to create	account
`iam:quota-groupnumber`	Number of IAM groups	account
`iam:quota-usernumber`	Number of IAM users	account

Default Quota

Contrary to IAM policies, by default, there is no quota limits (except the hard system limit) on any resource allocations for a user or an account. Also, system administrators are not constrained by any quota. Account administrators are only be constrained by account quota.

Algorithms

This topic describes the algorithms used by Eucalyptus to determine access.

Policy Evaluation Algorithm

You can associated multiple policies and permission statements with a user. The way these are combined together to control the access to resources in an account is defined by the policy evaluation algorithm. Eucalyptus implements the *same policy evaluation algorithm as AWS IAM*:

1. If the request user is account admin, access is allowed.
2. Otherwise, collect all the policy statements associated with the request user (attached to the user and all the groups the user belongs to), which match the incoming request (i.e. based on the API being invoked and the resources it is going to access).

    a. If there is no matched policy statement, access is denied (default implicit deny).
    b. Otherwise, evaluate each policy statement that matches.

        - If there is a statement that explicitly denies the access, the request is denied.
        - If there is no explicit deny, which means there is at least one explicit allow, access is allowed.

Access Evaluation Algorithm

Now we give the overall access evaluation combining both account level permissions and IAM permissions, which decides whether a request is accepted by Eucalyptus:

1. If the request user is sys admin, access is allowed.
2. Otherwise, check account level permissions, e.g. image launch permission, to see if the request user's account has access to the specific resources.

    a. If not, the access is denied.
    b. Otherwise, invoke the policy evaluation algorithm to check if the request user has access to the resources based on IAM policies.

Quota Evaluation Algorithm

Like the normal IAM policies, a user may be associated with multiple quota policies (and multiple quota statements). How all the quota policies are combined to take effect is defined by the quota evaluation algorithm:

1. If the request user is sys admin, there is no limit on resource usage.
2. Otherwise, collect all the quotas associated with the request user, including those attached to the request user's account and those attached to the request user himself/herself (for account admin, we only need collect account quotas).

3. Evaluate each quota one by one. Reject the request as long as there is one quota being exceeded by the request. Otherwise, accept the request.

Note: The hard limits on some resources override quota limits. For example, `walrus.storagemaxbucketsizeinmb` (system property) overrides the `s3:quota-bucketsize` (quota key).

Sample Policies

A few example use cases and associated policies.

Here are some example use cases and associated polices. You can edit these polices for your use, or use them as examples of JSON syntax and form.

Tip: For more information about JSON syntax used with AWS resources, go to *Using AWS Identity and Access Management*.

Examples: Allowing Specific Actions

The following policy allows a user to only run instances and describe things.

```
{
   "Statement":[{
      "Effect":"Allow",
      "Action":["ec2:*Describe*","ec2:*Run*"],
      "Resource":"*",
      }
   ]
   }
```

The following policy allows a user to only list things:

```
{
  "Statement": [
    {
      "Sid": "Stmt1313686153864",
      "Action": [
        "iam:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
  }
```

The following policy grants a generic basic user permission for running instances and describing things.

```
{
  "Statement": [
    {
      "Sid": "Stmt1313605116084",
      "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AttachVolume",
        "ec2:Authorize*",
        "ec2:CreateKeyPair",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSnapshot",
        "ec2:CreateVolume",
        "ec2:DeleteKeyPair",
        "ec2:DeleteSecurityGroup",
        "ec2:DeleteSnapshot",
        "ec2:DeleteVolume",
        "ec2:Describe*",
        "ec2:DetachVolume",
        "ec2:DisassociateAddress",
        "ec2:GetConsoleOutput",
        "ec2:RunInstances",
        "ec2:TerminateInstances"
        "ec2:ReleaseAddress"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Examples: Denying Specific Actions

The following policy allows a user to do anything but delete.

```
{
  "Statement": [
    {
      "Action": [
        "ec2:Delete*"
      ],
      "Effect": "Deny",
      "Resource": "*"
    }
  ]
  }
```

The following policy denies a user from creating other users.

```
{
  "Statement": [
    {
      "Sid": "Stmt1313686153864",
      "Action": [
        "iam:CreateUser"
      ],
      "Effect": "Deny",
      "Resource": "*"
    }
  ]
}
```

Examples: Specifying Time Limits

The following policy allows a user to run instances within a specific time.

```
{
  "Statement": [
    {
      "Sid": "Stmt1313453084396",
      "Action": [
        "ec2:RunInstances"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "DateLessThanEquals": {
          "aws:CurrentTime": "2011-08-16T00:00:00Z"
        }
      }
    }
  ]
}
```

The following policy blocks users from running instances at a specific time.

```
{
  "Statement": [
    {
      "Sid": "Stmt1313453084396",
      "Action": [
        "ec2:RunInstances"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "DateLessThanEquals": {
          "aws:CurrentTime": "2011-08-16T00:00:00Z"
        }
      }
    }
  ]
  }
```

The following policy keeps alive an instance for 1,000 hours (60,000 minutes).

```
{
  "Statement": [
    {
      "Action": ["ec2:RunInstances" ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": { "NumericEquals":{"ec2:KeepAlive":"60000"}}
    }
  ]
  }
```

The following policy sets an expiration date on running instances.

```
{
  "Statement": [
    {
      "Action": ["ec2:RunInstances" ],
      "Effect": "Allow",
      "Resource": "*",
     "Condition": { "DateEquals":{"ec2:ExpirationTime":"2011-08-16T00:00:00Z"}}

    }
  ]
}
```

Examples: Restricting Resources

The following policy allows users to only launch instances with a large image type.

```
{
  "Statement": [
    {
      "Action": [
        "ec2:RunInstances"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ec2:::vmtype/m1.xlarge"
    }
  ]
  }
```

The following policy restricts users from launching instances with a specific image ID.

```
{
  "Statement": [
    {
      "Action": [
        "ec2:RunInstances"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:ec2:::image/emi-0FFF1874"
    }
  ]
  }
```

The following policy restricts users from allocating addresses to a specific elastic IP address.

```
{
  "Statement": [
    {
      "Sid": "Stmt1313626078249",
      "Action": "*",
      "Effect": "Deny",
      "Resource": "arn:aws:ec2:::address/192.168.10.140"
    }
  ]
  }
```

The following policy denies volume access.

```
{
  "Statement": [
    {
      "Action": [
        "ec2:*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:ec2:::volume/*"
    }
  ]

}
```

Note: For policies attached to an account, quota limits can be specified. See the Quotas section for further details.

LDAP/AD Integration

You can use the Eucalyptus LDAP/Active Directory (AD) integration to synchronize existing LDAP/AD user and group information with Eucalyptus.

When you enable LDAP/AD synchronization, Eucalyptus imports specified user and group information from LDAP or AD and maps them into a predefined two-tier account/group/user structure

Note that Eucalyptus only imports the identities and some related information. Any Eucalyptus-specific attributes are still managed from Eucalyptus. These include:

- User credentials: secret access keys and X.509 certificates.
- Policies: IAM policies and quotas. Policies are associated with identities within Eucalyptus, and stored in internal database.

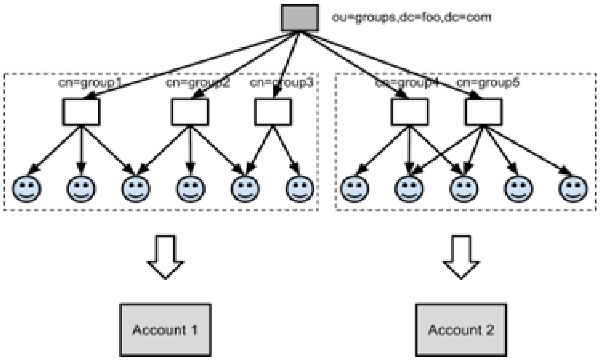
Also note that special identities, including system administrators and account administrators, are created in Eucalyptus and not imported from LDAP/AD. Only normal user identities are imported.

Important: If you integrate LDAP/AD, you do not need to create IAM user login profiles for your users.

Identity Mapping

Identities in LDAP/AD are organized differently from the identity structure in Eucalyptus. So a transformation is required to map LDAP/AD identities into Eucalyptus.

The following image shows a simple scheme of how the mapping works. In this scheme, the user groups in LDAP tree are partitioned into two sets. Each set is mapped into one separate account. Group 1, 2 and 3 are mapped to Account 1 and Group 4 and 5 are mapped to Account 2. As the result, all users in Group 1, 2 and 3 will be in Account 1, and all users in Group 4 and 5 will be in Account 2.



To summarize the mapping method:

1. Pick user groups from LDAP/AD and combine them into different accounts. There are two ways of doing this:
   - Use something called accounting groups. Account groups are essentially groups of groups. Accounting groups rely on a key understanding of object class types in LDAP. In short, accounting groups are mapped to STRUCTURAL object classes in LDAP. For more information about object class types, refer to the *LDAP Models RFC* under the "2.4. Object Classes". Each accounting group contains multiple user groups in LDAP/AD. Then each accounting group maps to an account in Eucalyptus.
   - Manually partition groups into accounts. Each group partition maps to an account.
2. Once the accounts are defined (by accounting groups or group partitions), all the LDAP/AD user groups will be mapped into Eucalyptus groups within specific accounts; and LDAP/AD users will be mapped into Eucalyptus users. Using the options to filter the groups and users to be imported into Eucalyptus allows granular control.
3. Groups are group object types in LDAP. The group object type in LDAP/AD needs to have the attribute type determining membership where the value is the Fully Distinguished Name (FDN) of the user(s). Some examples of group object types for LDAP/AD are as follows:
   - *groupOfNames*
   - *groupOfUniqueNames*
   - *Group-Of-Names*
   - *groupOfUniqueNames*
   - *Group*

Note that each group can be mapped into multiple accounts. But understand that Eucalyptus accounts are separate name spaces. So for groups and users that are mapped into different accounts, their information (name, attributes, etc) will be duplicated in different accounts. And duplicated users will have separate credentials in different accounts. For example, Group 1 may map to both Account 1 and Account 2. Say user A belongs to Group 1. Then Account 1 will have user A and Account 2 will also have user A. User A in Account 1 and user A in Account 2 will have different credentials, policies, etc., but the same user information.

Note: Currently, there is not a way to map individual users into an account. The mapping unit is LDAP user group. What maps where groups and users end up regarding accounts DEPENDS upon the accounting-groups or groups-partition definitions.

LDAP/AD Integration Configuration

The LDAP/AD Integration Configuration (LIC) is a JSON format file. This file specifies everything Eucalyptus needs to know about how to synchronize with an LDAP or AD service.

You can find a LIC template at `/usr/share/eucalyptus/lic_template`. This template shows all the fields of the LIC, and provides detailed documentation and example values for each field.

To start a LIC file, use the LIC command line tool.

```
/usr/sbin/euca-lictool --password <password> --out example.lic
```

The above command invokes the LIC tool to create a template LIC and fill in the encrypted password for authenticating to LDAP/AD service (i.e. the password of the administrative user for accessing the LDAP/AD during synchronization). The LIC tool’s primary functions are to encrypt the LDAP/AD password and to generate the starting LIC template. The usage of the LIC tool shows different ways to invoke the command.

Once you have the LIC template, you can fill in the details by editing the “*.lic” file using your favorite editor as it is a simple text file. As we said above, the LIC file is in JSON format. Each top level entity specifies one aspect of the LDAP/AD synchronization. The following shows one possible example of a LIC file.

```
{
"ldap-service":{
  "server-url":"ldap://localhost:7733",
  "auth-method":"simple",
  "user-auth-method":"simple",
  "auth-principal":"cn=ldapadmin,dc=foo,dc=com",
  "auth-credentials": "{RSA/ECB/PKCS1Padding}EAXRnvwnKtCZOxSrD/F3ng/yHH3J4jMxNUS

  kJJf6oqNMsUihjUerZ20e5iyXImPgjK1ELAPnppEfJvhCs7woS7jtFsedunsp5DJCNhgmOb2CR/MnH

  11V3FNY7bBWoew5A8Wwy6x7YrPMS0j7dJkwM7yfp1Z6AbKOo2688I9uIvJUQwEKS4dOp7RVdA0izlJ

  BDPAxiFZ2qa40VjFI/1mggbiWDNlgxiVtZXAEK7x9SRHJytLS8nrNPpIvPuTg3djKiWPVOLZ6vpSgP

  cVEliP261qdUfnf3GDKi3jqbPpRRQ6n8yI6aHw0gAtq8/qPyqjkkDP8JsGBgmXMxiCNPogbWg==",

  "use-ssl":"false",
  "ignore-ssl-cert-validation":"false",
  "krb5-conf":"/path/to/krb5.conf",
},

"sync":{
  "enable":"true",
  "auto":"true",
  "interval":"900000",
  "clean-deletion":"false",
},

"accounting-groups":{
  "base-dn":"ou=groups,dc=foo,dc=com",
  "id-attribute":"cn",
  "member-attribute":"member",
  "selection":{
      "filter":"objectClass=accountingGroup",
      "select":["cn=accountingToSelect,ou=Groups,dc=foo,dc=com"],
      "not-select":["cn=accountingToIgnore,ou=Groups,dc=foo,dc=com"],
  }
},

"groups":{
  "base-dn":"ou=groups,dc=foo,dc=com",
  "id-attribute":"cn",
  "member-attribute":"member",
  "selection":{
      "filter":"objectClass=groupOfNames",
      "select":["cn=groupToSelect,ou=Groups,dc=foo,dc=com"],
      "not-select":["cn=groupToIgnore,ou=Groups,dc=foo,dc=com"],
  }
},

"users":{
  "base-dn":"ou=people,dc=foo,dc=com",
  "id-attribute":"uid",
  "user-info-attributes":{
      "fullName":"Full Name",
      "email":"Email"
  },
  "selection":{
      "filter":"objectClass=inetOrgPerson",
      "select":["uid=john,ou=People,dc=foo,dc=com",
"uid=jack,ou=People,dc=foo,dc=com"],
      "not-select":["uid=tom,ou=People,dc=foo,dc=com"],
  }
},
```

In the following sections explain each field of LIC in detail.

ldap-service

The `ldap-service` element contains everything related to the LDAP/AD service.

Element	Description
server-url	The LDAP/AD server URL, starting with ldap://.
auth-method	The LDAP/AD authentication method to perform synchronization.
auth-principal	The ID of the administrative user for synchronization.
auth-credentials	The credentials for LDAP/AD authentication, like a password. We recommend that you encrypt this using `/usr/sbin/euca-lictool`.
user-auth-method	The LDAP/AD authentication method for normal users to perform Management Console login. • *simple*: for clear text user/password authentication. • *DIGEST-MD5*: for SASL authentication using MD5 • *GSSAPI*: SASL authentication using Kerberos V5.
use-ssl	Specifies whether to use SSL for connecting to LDAP/AD service. If this option is enabled, make sure the SSL port for LDAP is defined as part of the server-url. The default port for LDAP+SSL is port 636.
ignore-ssl-cert-validation	Specifies whether to ignore self-signed SSL certs. This is useful when you only have self-signed SSL certs for your LDAP/AD services.
krb5-conf	The file path for krb5.conf, if you use GSSAPI authentication method.

sync

The `sync` element contains elements for controlling synchronization.

Element	Description
enable	Set to "true" to enable LDAP synchronization. When this is “false”, all other fields can be ignored. Default value: false
auto	Set to true to turn on automatic synchronization. Set to false to turn off synchronization.
interval	The length in milliseconds of the automatic synchronization interval.
clean-deletion	Parameter denoting whether to remove identity entities from Eucalyptus when they are deleted from LDAP. Set to true if you want Eucalyptus to remove any identities once their counterparts in LDAP are deleted. Set to false if you want these identities kept without being purged.

accounting-groups

This section uses a special group in LDAP/AD to designate accounts in the Eucalyptus “accounting group.” The accounting group takes normal LDAP/AD groups as members, i.e., they are groups of groups.

The accounting group’s name becomes the account name in Eucalyptus. The member groups become Eucalyptus groups in that account. And the users of all those groups become Eucalyptus users within that account and corresponding Eucalyptus groups.

Important: If you use `accounting-groups`, remove the `groups-partition` section. These two sections are mutually exclusive.

Element	Description
base-dn	The base DN of accounting groups in the LDAP/AD tree.
id-attribute	The ID attribute name of the accounting group entry in LDAP/AD tree.
member-attribute	The LDAP/AD attribute name for members of the accounting group.
selection	The accounting groups you want to map to. This contains the following elements: • *filter*: The LDAP/AD searching filter used for the LDAP/AD search to get the relevant LDAP/AD entities, e.g. the users to be synchronized. (Example: objectClass=groupOfNames). This element works the same as the filter option that is found in ldapsearch, therefore when doing more advanced searching using compound filters, use boolean operators - AND (&), OR (|), and/or NOT (!). (Example: `(&(ou=Sales)(objectClass=groupOfNames))` • *select*: Explicitly gives the full DN of entities to be synchronized, in case they can not be specified by the search filter. (Example: cn=groupToSelect,ou=Groups,dc=foo,dc=com) • *not-select:* Explicitly gives the full DN of entities NOT to be synchronized, in case this can not be specified by the search filter. (Example: cn=groupToIgnore,ou=Groups,dc=foo,dc=com)

groups-partition

Like accounting-groups, groups-partition specifies how to map LDAP/AD groups to Eucalyptus accounts. However, in this section you to manually specify which LDAP/AD groups you want to map to Eucalyptus accounts.

Important: If you use `groups-partition`, remove the `accounting-groups` section. These two sections are mutually exclusive.

The Eucalyptus accounts are created by partitioning LDAP/AD groups. Each partition composes an Eucalyptus account. So all the groups within the partition become Eucalyptus groups within that account. All the users of those groups will become Eucalyptus users within that account and the corresponding Eucalyptus groups.

This section requires that you specify one partition at a time, using a list of JSON key-value pairs. For each entry, the key is the account name to be mapped and the value is a list of names of LDAP/AD groups to be mapped into the account. For example:

```
"groups-partition": {
          "salesmarketing": ["sales", "marketing"],
          "devsupport": ["engineering", "support"],
}
```

Here salesmarketing and devsupport are names for the groups partition and are used as the corresponding Eucalyptus account names.

Tip: If you use groups-partition, remove the accounting-groups section. These two sections are mutually exclusive.

groups

The `groups` element specifies how to map LDAP/AD groups to Eucalyptus groups. It contains the elements listed in the following table. The meanings are similar to those in accounting-groups element.

Element	Description
base-dn	The base DN for searching groups.
id-attribute	The ID attribute name of the LDAP group.
member-attribute	The name of the attribute for group members. Usually, it is member in modern LDAP implementation, which lists full user DN.
selection	The specific LDAP/AD groups you want to map to. This contains the following elements: • *filter*: The LDAP/AD searching filter used for the LDAP/AD search to get the relevant LDAP/AD entities, e.g. the users to be synchronized. (Example: objectClass=groupOfNames). This element works the same as the filter option that is found in ldapsearch, therefore when doing more advanced searching using compound filters, use boolean operators - AND (&), OR (|), and/or NOT (!). (Example: `(&(ou=Sales)(objectClass=groupOfNames))` • *select*: The LDAP/AD searching filter used for the LDAP/AD search to get the relevant LDAP/AD entities, e.g. the users to be synchronized. (Example: objectClass=groupOfNames) • *not-select:* Explicitly gives the full DN of entities NOT to be synchronized, in case this can not be specified by the search filter. (Example: cn=groupToIgnore,ou=Groups,dc=foo,dc=com)

users

Explicitly gives the full DN of entities NOT to be synchronized, in case this can not be specified by the search filter.

Element	Description
base-dn	The base DN for searching users.
id-attribute	The attribute ID of the LDAP user.
selection	The specific LDAP/AD users you want to map to. This contains the following elements: • *filter*: The LDAP/AD searching filter used for the LDAP/AD search to get the relevant LDAP/AD entities, e.g. the users to be synchronized. (Example: objectClass=organizationalPerson). This element works the same as the filter option that is found in ldapsearch, therefore when doing more advanced searching using compound filters, use boolean operators - AND (&), OR (|), and/or NOT (!). (Example: `(&(ou=Sales)(objectClass=organizationalPerson)))` • *select*: Explicitly gives the full DN of entities to be synchronized, in case they can not be specified by the search filter. (Example: cn=userToSelect,ou=People,dc=foo,dc=com) • *not-select:* Explicitly gives the full DN of entities NOT to be synchronized, in case this can not be specified by the search filter. (Example: cn=userToIgnore,ou=People,dc=foo,dc=com)

Synchronization Process

This topic explains what happens to start the synchronization process and what the synchronization process does.

The synchronization always starts when the following happens:

- You manually upload a LDAP/AD Integration Configuration (LIC) file. Every new or updated LIC upload triggers a new synchronization.
- If the automatic synchronization is enabled, a synchronization is started when the timer goes off.

Note: Eucalyptus does not allow concurrent synchronization. If you trigger synchronization more than once within a short time period, Eucalyptus only allows the first one.

During a synchronization, everything specified by an LIC in the LDAP/AD tree will be downloaded into Eucalyptus' internal database. Each synchronization is a merging process of the information already in the database and the information from LDAP/AD. There are three cases for each entity: user, group or account:

- If an entity from LDAP/AD is not in Eucalyptus, a new one is created in the database.
- If an entity from LDAP/AD is already in Eucalyptus, the Eucalyptus version is updated. For example, if a user's info attributes are changed, those changes are downloaded and updated.
- If an entity in Eucalyptus is missing from LDAP/AD, it will be removed from the database if the clean-deletion option in LIC is set to true. Otherwise, it will be left in the database.

Important: If clean-deletion is set to true, the removed entities in Eucalyptus will be lost forever, along with all its permissions and credentials. The resources associated with the entity will be left untouched. It is system administrator's job to recycle these resources.

Access Tasks

This section provides details about the tasks you perform using policies and identities. The tasks you can perform are divided up into tasks for users, tasks for groups, and tasks for policies.

The following use cases detail work flows for common processes:

- *Use Case: Create an Administrator*
- *Use Case: Create a User*

You can perform the following access-related tasks listed in the following sections:

- Accounts:
  - *Add an Account*
  - *Approve an Account*
  - *Reject an Account*
  - *Rename an Account*
  - *List Accounts*
  - *Delete an Account*
- Groups:
  - *Create a Group*
  - *Add a Group Policy*
  - *Modify a Group*
  - *Add a User to a Group*
  - *Remove a User from a Group*
  - *List Groups*
  - *List Policies for a Group*
  - *Delete a Group*
- Users:
  - *Add a User*
  - *Add a User to a Group*
  - *Create a Login Profile*
  - *Modify a User*
  - *List Users*
  - *Delete a User*
- Credentials:
  - *Generating User Credentials*
  - *Retrieving Existing User Credentials*
  - *Uploading a Certificate*
  - *Working with Administrator Credentials*
- Roles:
  - *Launch an Instance with a Role*

Note: To distinguish an IAM operation, run commands with --help and look for *iam* described in the --url option, which indicates the IAM service that the command is talking to.

Use Case: Create an Administrator

This use case details tasks for creating an administrator. These tasks require that you have your account credentials for sending requests to Eucalyptus using the command line interface (CLI) tools.

To create an administrator account, perform the tasks that follows.

Create an Admin Group

Eucalyptus recommends using account credentials as little as possible. You can avoid using account credentials by creating a group of users with administrative privileges.

1. Create a group called `administrators`.

   ```
   euare-groupcreate -g administrators
   ```

2. Verify that the group was created.

   ```
   euare-grouplistbypath
   ```

   Eucalyptus returns a listing of the groups that have been created, as in the following example.

   ```
   arn:aws:iam::123456789012:group/administrators
   ```

Add a Policy to the Group

Add a policy to the administrators group that allows its members to perform all actions in Eucalyptus.

Enter the following command to create a policy called `admin-root` that grants all actions on all resources to all users in the administrators group:

```
euare-groupaddpolicy -p admin-root -g administrators -e Allow -a "*" -r "*"
-o
```

Create an Administrative User

Create a user for day-to-day administrative work and add that user to the administrators group.

1. Enter the following command to create an administrative user named `alice`:

   ```
   euare-usercreate -u alice
   ```

2. Add the new administrative user to the administrators group.

   ```
   euare-groupadduser -g administrators -u alice
   ```

Generate Administrative Credentials

To start running commands as the new administrative user, you must create an access key for that user.

1. Enter the following command to generate an access key for the administrative user:

   ```
   euare-useraddkey -u alice
   ```

   Eucalyptus returns the access key ID and the user's secret key.

2. Open the `~/.eucarc` file and replace your account credentials you just created, as in this example:

   ```
   export EC2_ACCESS_KEY='WOKSEQRNM1LVIR702XVX1'
   export EC2_SECRET_KEY='0SmLCQ8DAZPKoaC7oJYcRMfeDUgGbiSVv1ip5WaH'
   ```

3. Save and close the file.
4. Open the `~/.iamrc` file and replace your account credentials, as in this example:

   ```
   AWSAccessKeyId=WOKSEQRNM1LVIR702XVX1
   AWSSecretKey=0SmLCQ8DAZPKoaC7oJYcRMfeDUgGbiSVv1ip5WaH
   ```

5. Save and close the file.
6. Switch euca2ools over to using the new credentials.

   ```
   source ~/.eucarc
   ```

Use Case: Create a User

This use case details tasks needed to create a user with limited access.

Create a Group

We recommend that you apply permissions to groups, not users. In this example, we will create a group for users with limited access.

1. Enter the following command to create a group for users who will be allowed create snapshots of volumes in Eucalyptus.

```
euare-groupcreate -g ebs-backup
```

2. Open an editor and enter the following JSON policy:

```
{
  "Statement": [
    {
      "Action": [
        "ec2:CreateSnapshot"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

3. Save and close the file.
4. Enter the following to add the new policy name `allow-snapshot` and the JSON policy file to the `ebs-backup` group:

```
euare-groupuploadpolicy -g ebs-backup -p allow-snapshot -f allow-snapshot.json
```

Create the User

Create the user for the group with limited access.

Enter the following command to create the user `sam` in the group `ebs-backup` and generate a new key pair for the user:

```
euare-usercreate -u sam -g ebs-backup -k
```

Eucalyptus responds with the access key ID and the secret key, as in the following example:

```
AKIAJ25S6IJ5K53Y5GCA
QLKyiCpfjWAvlo9pWqWCbuGB9L3T61w7nYYF057l
```

Accounts

Accounts are the primary unit for resource usage accounting. Each account is a separate name space and is identified by its UUID (Universal Unique Identifier).

Tasks performed at the account level can only be done by the users in the **eucalyptus** account.

Add an Account

To add an account perform the steps listed in this topic.

To add a new account:

Enter the following command:

```
euare-accountcreate -a <account_name>
```

Eucalyptus returns the account name and its ID, as in this example:

```
account01  592459037010
```

Rename an Account

To rename an account perform the steps listed in this topic.

To change an account's name:

Enter the following command:

```
uare-accountaliascreate -a <new_name>
```

List Accounts

To list accounts perform the steps in this topic.

Use the `euare-accountlist` command to list all the accounts in an account or to list all the users with a particular path prefix. The output lists the ARN for each resulting user.

```
euare-userlistbypath -p <path>
```

Delete an Account

To delete an account perform the steps listed in this topic.

Tip: If there are resources tied to the account that you delete, the resources remain. We recommend that you delete these resources first.

Enter the following command:

```
euare-accountdel -a <account_name> -r true
```

Use the `-r` option set to `true` to delete the account recursively. You don't have to use this option if have already deleted users, keys, and passwords in this account.

Eucalyptus does not return any message.

Groups

Groups are used to share resource access authorizations among a set of users within an account. Users can belong to multiple groups.

Important: A group in the context of access is not the same as a security group.

This section details tasks that can be performed on groups.

Create a Group

To create a group perform the steps listed in this topic.

Enter the following command:

```
euare-groupcreate -g <group_name>
```

Eucalyptus does not return anything.

Add a Group Policy

To add a group policy perform the steps listed in this topic.

Enter the following command:

```
euare-groupaddpolicy -g <group_name> -p <policy_name> -e <effect> -a
       <actions> -o
```

The optional `-o` parameter tells Eucalyptus to return the JSON policy, as in this example:

```
{"Version":"2008-10-17","Statement":[{"Effect":"Allow",
"Action":["ec2:RunInstances"], "Resource":["*"]}]}
```

Modify a Group

To modify a group perform the steps listed in this topic.

Modifying a group is similar to a "move" operation. Whoever wants to modify the group must have permission to do it on both sides of the move. That is, you need permission to remove the group from its current path or name, and put that group in the new path or name.

For example, if a group changes from one area in a company to another, you can change the group's path from `/area_abc/` to `/area_efg/`. You need permission to remove the group from `/area_abc/`. You also need permission to put the group into `/area_efg/`. This means you need permission to call `UpdateGroup` on both `arn:aws:iam::123456789012:group/area_abc/*` and `arn:aws:iam::123456789012:group/area_efg/*`.

1. Enter the following command to modify the group's name:

   ```
   euare-groupmod -g <group_name> --new-group-name <new_name>
   ```

   Eucalyptus does not return a message.
2. Enter the following command to modify a group's path:

   ```
   euare-groupmod -g <group_name> -p <new_path>
   ```

   Eucalyptus does not return a message.

Remove a User from a Group

To remove a user from a group perform the steps listed in this topic.

Enter the following command:

```
euare-groupremoveuser -g <group_name> -u <user-name>
```

List Groups

To list groups perform the steps listed in this topic.

Enter the following command:

```
euare-grouplistbypath
```

Eucalyptus returns a list of paths followed by the ARNs for the groups in each path. For example:

```
arn:aws:iam::eucalyptus:group/groupa
```

List Policies for a Group

To list policies for a group perform the steps listed in this topic.

Enter the following command:

```
euare-grouplistpolicies -g <group_name>
```

Eucalyptus returns a listing of all policies associated with the group.

Delete a Group

To delete a group perform the steps listed in this topic.

When you delete a group, you have to remove users from the group and delete any policies from the group. You can do this with one command, using the `euare-groupdel` command with the `-r` option. Or you can follow the following steps to specify who and what you want to delete.

1. Individually remove all users from the group.

   ```
   euare-groupremoveuser -g <group_name> -u <user_name>
   ```

2. Delete the policies attached to the group.

   ```
   euare-groupdelpolicy -g <group_name> -p <policy_name>
   ```

3. Delete the group.

   ```
   euare-groupdel -g <group_name>
   ```

The group is now deleted.

Users

Users are subsets of accounts and are added to accounts by an appropriately credentialed administrator. While the term **user** typically refers to a specific person, in Eucalyptus, a **user** is defined by a specific set of credentials generated to enable access to a given account. Each set of user credentials is valid for accessing only the account for which they were created. Thus a user only has access to one account within a Eucalyptus system. If an individual person wishes to have access to more than one account within a Eucalyptus system, a separate set of credentials must be generated (in effect a new ‘user’) for each account (though the same username and password can be used for different accounts).

When you need to add a new user to your Eucalyptus cloud, you'll go through the following process:

1	*Create a user*
2	*Add user to a group*
3	*Give user a login profile*

Add a User

To add a user, perform the steps in this topic.

Enter the following command

```
euare-usercreate -u <user_name> -g <group_name> -k
```

Eucalyptus does not return a response.

Tip: If you include the `-v` parameter, Eucalyptus returns a response that includes the user's ARN and GUID.

Add a User to a Group

To add a user to a group perform the steps listed in this topic.

Enter the following command:

```
euare-groupadduser -g <group_name> -u <user-name>
```

Create a Login Profile

To create a login profile, perform the tasks in this topic.

Enter the following command:

```
euare-useraddloginprofile -u <user_name> -p <password>
```

Eucalyptus does not return a response.

Modify a User

Modifying a user is similar to a "move" operation. To modify a user, you need permission to remove the user from the current path or name, and put that user in the new path or name.

For example, if a user changes from one team in a company to another, you can change the user's path from `/team_abc/` to `/team_efg/`. You need permission to remove the user from `/team_abc/`. You also need permission to put the user into `/team_efg/`. This means you need permission to call UpdateUser on both `arn:aws:iam::123456789012:user/team_abc/*` and `arn:aws:iam::123456789012:user/team_efg/*`.

To rename a user:

1. Enter the following command to rename a user:

   ```
   euare-usermod -u <user_name> --new-user-name <new_name>
   ```

   Eucalyptus does not return a message.
2. Enter the following command:

   ```
   euare-groupmod -u <user_name> -p <new_path>
   ```

   Eucalyptus does not return a message.

List Users

To list users within a path, perform the steps in this topic.

Use the `euare-userlistbypath` command to list all the users in an account or to list all the users with a particular path prefix. The output lists the ARN for each resulting user.

```
euare-userlistbypath -p <path>
```

Delete a User

To delete a user, perform the tasks in this topic.

Enter the following command

```
euare-userdel -u <user_name>
```

Eucalyptus does not return a response.

Credentials

Eucalyptus uses different types of credentials for both user and administrative functions. Besides the login and password used for accessing the Eucalyptus Administrator Console, Eucalyptus uses an SSH keypair and an X.509 certificate to control access to instances and to Eucalyptus system functions using the command line tools. This section discusses the various types of credentials and how to use them.

- *Working with User Credentials*
- *Working with Administrator Credentials*

Working with User Credentials

This section describes how to create new user credentials and retrieve existing user credentials.

- *Generating User Credentials*
- *Retrieving Existing User Credentials*
- *Uploading a Certificate*

Generating User Credentials

The first time you get credentials using the `euca_conf` command, a new secret access key is generated. On each subsequent request to get credentials, an existing active secret key is returned. You can also generate new keys using the `euare-useraddkey` command.

Tip: Eucalyptus creates a new private key and X.509 certificate pair each time you request a user's credentials using `euca_conf`.

- To generate a new key for a user by an account administrator, enter the following

  ```
  euare-useraddkey -u <user_name>
  ```

- To generate a private key and an X.509 certificate pair, enter the following:

```
euare-usercreatecert -u <user_name>
```

Retrieving Existing User Credentials

When you retrieve existing credentials, Eucalyptus returns a zip file that contains keys, certificates, a bash script, and several other required files. To use these credentials with such CLI tools as euca2ools or ec2-tools, unzip the zip file to a directory of your choice.

- An administrator with a root access to the machine on which CLC is installed can get credentials using `euca_conf` CLI tool on that machine.

  ```
  /usr/sbin/euca_conf --cred-account <account> --cred-user <user_name>
   --get-credentials <filename>.zip
  ```

  Where <account> and <user_name> are the names of the account and the user whose credentials are retrieved.

  Tip: You can omit the `--cred-account` and `--cred-user` options when you get credentials for the **admin** user of the **eucalyptus** account.

- A user can get his or her credentials by logging in into the Eucalyptus Administrator Console and clicking **Download new credentials** in the drop-down menu at the top of the screen. This will result in a download of a zip file.

In the following example we download the credentials zip file to `~/.euca`, then change access permissions, as shown:

```
mkdir ~/.euca
cd ~/.euca
unzip <filepath>/<creds_zipfile>.zip
chmod 0700 ~/.euca
chmod 0600 *
```

Important: The zip file with credentials contains security-sensitive information. We recommend that you remove or read- and write-protect the file from other users after unzipping.

Alternatively, you can view and copy your access keys and X.509 certificates from the Eucalyptus Administrator Console after logging in, using the Navigation menu.

Uploading a Certificate

To upload a certificate provided by a user:

Enter the following command:

```
euare-useraddcert -u <user_name> -f <cert_file>
```

Working with Administrator Credentials

Important: When you run the following command, you are requesting a new X.509 and a corresponding private key. You cannot retrieve an existing private key.

To generate a set of credentials:

1. Log in to the CLC.
2. Get administrator credentials and source eucarc:

```
/usr/sbin/euca_conf --get-credentials admin.zip
unzip admin.zip
chmod 0600 *
source eucarc
```

Synchronize LDAP/AD

To start an LDAP/AD synchronization:

1. Create an LDAP/AD Integration Configuration (LIC) file to specify all the details about the LDAP/AD synchronization.
2. Upload the LIC file to Eucalyptus using `euca-modify-property`.

Start a LIC File

To start a LIC file perform the steps listed in this topic.

The LIC is a file in JSON format, specifying everything Eucalyptus needs to know about how to synchronize with an LDAP or AD service. Eucalyptus provides a LIC template at `${EUCALYPTUS}/usr/share/eucalyptus/lic_template`. This template shows all the fields of the LIC, and provides detailed documentation and example values for each field.

To start a LIC file:

Enter the following command:

```
/usr/sbin/euca-lictool --password secret --out example.lic
```

The above command invokes the LIC tool to create a template LIC and fill in the encrypted password for authenticating to LDAP/AD service (i.e. the password of the administrative user for accessing the LDAP/AD during synchronization). The LIC tool’s primary functions are to encrypt the LDAP/AD password and to generate the starting LIC template. The usage of the LIC tool shows different ways to invoke the command.

Once you have the LIC template, you can fill in the details by editing the `*.lic` file using a text editor. Each top level entity specifies one aspect of the LDAP/AD synchronization.

Upload a New LIC File

To upload a new LIC file perform the steps listed in this topic.

To upload a new LIC file:

Enter the following:

```
/usr/sbin/euca-modify-property -f
            authentication.ldap_integration_configuration=<lic_filename.lic>
```

This triggers a new synchronization using the uploaded LIC file.

Roles

A *role* is a mechanism that allows applications to request temporary security credentials on a user or application's behalf.

Note: Eucalyptus roles are managed through the Eucalyptus Euare service, which is compatible with Amazon's Identity and Access Management service. For more information on IAM and roles, see the *Amazon IAM User Guide*.

Launch an Instance with a Role

To create a role for a Eucalyptus instance, you must first create a trust policy that you can use for it.

Create Trust Policy

You can create trust policies in two ways:

- a file method
- a command line method

Create trust policy using a file

1. Create a trust policy file with the contents below and save it in a text file called `role-trust-policy.json`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Create the role using the `euare-rolecreate` command, specifying the trust policy file that was previously created:

   ```
   # euare-rolecreate --role-name describe-instances -f role-trust-policy.json
   # euare-rolelistbypath
   arn:aws:iam::408396244283:role/describe-instances
   ```

3. Proceed with applying an access policy to a role.

Create trust policy using the command line

The other way to create the role is to use the command line options to specify the trust policy:

1. Issue the following string on the command line:

   ```
   # euare-rolecreate --role-name describe-instances --service
   http://compute.acme.eucalyptus-systems.com:8773/
   # euare-rolelistbypath
   arn:aws:iam::408396244283:role/describe-instances
   ```

2. Proceed with applying an access policy to a role.

Create and apply an access policy to a role

1. Create a policy and save it in a text file with a `.json` extension. The following example shows a policy that allows listing the contents of an S3 bucket called "mybucket":

   ```
   {
     "Statement": [
       {
         "Action": [
           "s3:ListBucket"
         ],
         "Effect": "Allow",
         "Resource": "arn:aws:s3:::mybucket"
       }
     ]
   }
   ```

   Note: For more information on policies, see *Policy Overview*.

2. Apply the access policy to the role using the `euare-roleuploadpolicy` command, passing in the filename of the policy you created in the previous step:

   ```
   euare-roleuploadpolicy --role-name mytestrole --policy-name s3-list-bucket
   --policy-document my-test-policy.json
   ```

Use a Role with an Instance Application

You can use the AWS Java SDK to programmatically perform IAM role-related operations in your Eucalyptus cloud. This example shows how to use the AWS SDK to retrieve the credentials for the IAM role associated with the Eucalyptus instance.

1. The following program lists the contents of the bucket "my-test-bucket" using the credentials stored in the Java system properties:

```
import com.amazonaws.auth.*;
import com.amazonaws.AmazonClientException;
import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.AWSCredentialsProvider;
import com.amazonaws.auth.ClasspathPropertiesFileCredentialsProvider;
import com.amazonaws.services.ec2.AmazonEC2;
import com.amazonaws.services.ec2.AmazonEC2Client;
import com.amazonaws.services.s3.*;
import com.amazonaws.services.s3.model.*;

public class MyTestApp {

    static AmazonEC2 ec2;
    static AmazonS3 s3;

    private static void init() throws Exception {

        AWSCredentialsProvider credentials = new 
ClasspathPropertiesFileCredentialsProvider();

        s3 = new AmazonS3Client(credentials);
        s3.setEndpoint("http://128.0.0.1:8773/services/Walrus");
    }


    public static void main(String[] args) throws Exception {

        init();

        try {

            String bucketName = "my-test-bucket";
            System.out.println("Listing bucket " + bucketName + ":");
            ListObjectsRequest listObjectsRequest = new 
ListObjectsRequest(bucketName, "", "", "", 200);
            ObjectListing bucketList;
            do {
                bucketList = s3.listObjects(listObjectsRequest);
                for (S3ObjectSummary objectInfo :
                        bucketList.getObjectSummaries()) {
                    System.out.println(" - " + objectInfo.getKey() + "  " +
                            "(size = " + objectInfo.getSize() + 
                            ")");
                }
                listObjectsRequest.setMarker(bucketList.getNextMarker());
            } while (bucketList.isTruncated());

        } catch (AmazonServiceException eucaServiceException ) {
            System.out.println("Exception: " + 
eucaServiceException.getMessage());
            System.out.println("Status Code: " + 
eucaServiceException.getStatusCode());
            System.out.println("Error Code: " + 
eucaServiceException.getErrorCode());
            System.out.println("Request ID: " + 
eucaServiceException.getRequestId());
        } catch (AmazonClientException eucaClientException) {
            System.out.println("Error Message: " + 
eucaClientException.getMessage());
        }
        System.out.println("===== FINISHED =====");
    }
}
```

This application produces output similar to the following:

```
Listing bucket my-test-bucket:
 - precise-server-cloudimg-amd64-vmlinuz-virtual.manifest.xml  (size = 3553)
 - precise-server-cloudimg-amd64-vmlinuz-virtual.part.0  (size = 4904032)
 - precise-server-cloudimg-amd64.img.manifest.xml  (size = 7014)
 - precise-server-cloudimg-amd64.img.part.0  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.1  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.10  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.11  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.12  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.13  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.14  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.15  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.16  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.17  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.18  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.19  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.2  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.20  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.21  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.22  (size = 2570400)
 - precise-server-cloudimg-amd64.img.part.3  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.4  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.5  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.6  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.7  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.8  (size = 10485760)
 - precise-server-cloudimg-amd64.img.part.9  (size = 10485760)
===== FINISHED =====
```

The problem with this approach is that the credentials are hardcoded into the application - this makes them less secure, and makes the application more difficult to maintain. Using IAM roles is a more secure and easier way to manage credentials for applications that run on Eucalyptus cloud instances.

2. Create a role with a policy that allows an instance to list the contents of a specific bucket, and then launch an instance with that role (for an example, see *Launch an Instance with a Role*. An example policy that allows listing of a specific bucket will look similar to the following:

```
{
  "Statement": [
    {
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::my-test-bucket"
    }
  ]
}
```

3. The following line of code retrieves the credentials that are stored in the application's credentials profile:

```
AWSCredentialsProvider credentials = new
ClasspathPropertiesFileCredentialsProvider();
```

To use the role-based credentials associated with the instance, replace that line of code with the following:

```
AWSCredentialsProvider credentials = new InstanceProfileCredentialsProvider();
```

The program now looks like this:

```
import com.amazonaws.auth.*;
import com.amazonaws.AmazonClientException;
import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.AWSCredentialsProvider;
import com.amazonaws.auth.ClasspathPropertiesFileCredentialsProvider;
import com.amazonaws.services.ec2.AmazonEC2;
import com.amazonaws.services.ec2.AmazonEC2Client;
import com.amazonaws.services.s3.*;
import com.amazonaws.services.s3.model.*;

public class MyTestApp {

    static AmazonEC2 ec2;
    static AmazonS3 s3;

    private static void init() throws Exception {

        AWSCredentialsProvider credentials = new
InstanceProfileCredentialsProvider();

        s3 = new AmazonS3Client(credentials);
        s3.setEndpoint("http://128.0.0.1:8773/services/Walrus");
    }


    public static void main(String[] args) throws Exception {

        init();

        try {

            String bucketName = "my-test-bucket";

            System.out.println("Listing bucket " + bucketName + ":");
            ListObjectsRequest listObjectsRequest = new
ListObjectsRequest(bucketName, "", "", "", 200);
            ObjectListing bucketList;
            do {
                bucketList = s3.listObjects(listObjectsRequest);
                for (S3ObjectSummary objectInfo :
                        bucketList.getObjectSummaries()) {
                    System.out.println(" - " + objectInfo.getKey() + "  " +
                            "(size = " + objectInfo.getSize() +
                            ")");
                }
                listObjectsRequest.setMarker(bucketList.getNextMarker());
            } while (bucketList.isTruncated());

        } catch (AmazonServiceException eucaServiceException ) {
            System.out.println("Exception: " +
eucaServiceException.getMessage());
            System.out.println("Status Code: " +
eucaServiceException.getStatusCode());
            System.out.println("Error Code: " +
eucaServiceException.getErrorCode());
            System.out.println("Request ID: " +
eucaServiceException.getRequestId());
        } catch (AmazonClientException eucaClientException) {
            System.out.println("Error Message: " +
eucaClientException.getMessage());
        }
        System.out.println("===== FINISHED =====");
    }

}
```

NOTE: Running this code outside of an instance will result in the following error message:

```
Listing bucket my-test-bucket:
Error Message: Unable to load credentials from Amazon EC2 metadata service
```

When the application is running on an instance that was launched with the role you created, the credentials for the role assigned to the instance will be retrieved from the Instance Metadata Service.

Manage Resources

This section includes tasks to help you manage your users' cloud resources.

Manage Compute Resources

To manage compute resources on a Eucalyptus cloud, use the `verbose` option in any `euca-describe-*` command.

The following are some examples you can use to view various compute resources. For more information about compute commands, see *EC2-Compatible Commands*.

- To see all instances running on your cloud, enter the following command:

  ```
  euca-describe-instances verbose
  ```

- To see all volumes in your cloud, enter the following command:

  ```
  euca-describe-volumes verbose
  ```

- To see all keypairs in your cloud, enter the following command:

  ```
  euca-describe-keypairs verbose
  ```

Manage Walrus Resources

This topic explains Walrus resources.

- **Bucket ACLs:** Access Control Lists (ACLs) allow an account to explicitly grant access to a bucket or object to another account. ACLs only work between accounts, not IAM users. You specify accounts with the CanonicalID or the email address associated with the account (for Eucalyptus this is the email of the account admin).
- **IAM Policies:** These are set by the admin of an account to control the access of users within that specific account. This is how an admin controls what users in that specific account are allowed to do. Policies can specify allow/deny on specific S3 operations (e.g. s3:GetObject, or s3:PutObject). IAM policies are set by sending the policy to the IAM (Euare) endpoint, not S3 (Walrus).
- **Bucket Policies:** These are IAM-like policies set by the bucket owner are not supported in Eucalyptus.

For more information about bucket ACLs, go to *Access Control List (ACL) Overview* and *Managing ACLs Using the REST API*.

For more information about IAM policies, go to *Using IAM Policies*.

Manage IAM Resources

To manage Euare (IAM) resources on your Eucalyptus cloud, use the `--as-account` option with any `euare-` command that describes, adds, deletes, or modifies resources. This option allows you to assume the role of the admin user for a given account. You can also use a policy to control and limit instances to specific availability zones.

The following are some examples. For more information about IAM commands, see *IAM-Compatible Commands*.

- To list all groups in an account, enter the following command:

  ```
  euare-grouplistbypath --as-account <account-name>
  ```

- To list all users in an account, enter the following command:

  ```
  euare-userslistbypath --as-account <account-name>
  ```

- To delete the login profile of a user in an account, enter the following command:

```
euare-userdelloginprofile --as-account <account-name> -u <user_name>
```

- To modify the login profile of a user in an account, enter the following command:

```
euare-usermod --as-account <account-name> -u <user_name> -n
<new_user_name>
```

- To restrict an image to a specific availability zone, edit and attach this sample policy to a user:

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":"ec2:*",
         "Resource":"*"
      },
      {
        "Effect": "Deny",
        "Action": [ "ec2:*" ],
        "Resource": "arn:aws:ec2:::availabilityzone/PARTI00",
        "Condition": {
          "ArnLike": {
            "ec2:TargetImage": "arn:aws:ec2:*:*:image/emi-239D37F2"
          }
        }
      }
    ]
  }
```

- To restrict a user to actions only within a specific availability zone, edit and attach this sample policy to a user:

```
{
    "Version": "2012-10-17",
    "Statement": [{
      "Effect": "Allow",
      "Action": [ "ec2:TerminateInstances" ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:AvailabilityZone": "PARTI00"
        }
      }
    }]
  }
```

- To deny actions at the account level, edit and attach this example policy to an account:

```
{
    "Statement": [ {
      "Effect": "Deny",
      "Action": [ "ec2:RunInstances" ],
      "Resource": "arn:aws:ec2:::availabilityzone/PARTI00",
      "Condition": {
          "ArnLike": {
              "ec2:TargetImage": "arn:aws:ec2:*:*:image/emi-239D37F2"
          }
      }
    } ]
  }
```

Manage CloudWatch Resources

To manage CloudWatch resources on a Eucalyptus cloud, use the `verbose` option in any `euwatch-` command that lists, deletes, modifies, or sets a CloudWatch resource.

The following are examples of what you can do with your CloudWatch resources. For more information about CloudWatch commands, see *CloudWatch-Compatible Commands*.

- To list all alarms for the cloud, run the following command:

  ```
  euwatch-describe-alarms verbose
  ```

Manage ELB Resources

To list and delete ELB resources on a Eucalyptus cloud, use the `verbose` option with any `eulb-describe-*` command.

The following are some examples.

- To list all detailed configuration information for the load balancers in your cloud, run the following command:

  ```
  eulb-describe-lbs verbose
  ```

- To list the details of policies for all load balancers in your cloud, run the following command:

  ```
  eulb-describe-lb-policies verbose
  ```

- To list meta information for all load balancer policies in your cloud, run the following command:

  ```
  eulb-describe-lb-policy-types verbose
  ```

- To delete any load balancer or any load balancer resource on the cloud, instead of using the ELB name, use the DNS name. For example:

  ```
  $ eulb-describe-lbs verbose
  LOAD_BALANCER MyLoadBalancer
  MyLoadBalancer-961915002812.lb.foobar.eucalyptus-systems.com
  2013-10-30T03:02:53.39Z

  $ eulb-delete-lb MyLoadBalancer-961915002812.lb.foobar.eucalyptus-systems.com

  $ eulb-describe-lbs verbose
  ```

Manage Auto Scaling Resources

You can list, delete, update, and suspend your Eucalyptus cloud's Autoscaling resources by passing the `-show-long` option with the keyword `verbose` with the appropriate `euscale-` command.

The followings are some examples you can use to act on your Auto Scaling resources. For more information about Auto Scaling commands, see .

- To show all launch configurations in your cloud, run the following command:

  ```
  euscale-describe-launch-configs --show-long verbose
  ```

- To show all Auto Scaling instances in your cloud, run the following command:

  ```
  euscale-describe-auto-scaling-groups --show-long verbose
  ```

- To show all Auto Scaling instances in your cloud, run the following command:

  ```
  euscale-describe-auto-scaling-groups --show-long verbose
  ```

- To delete an Auto Scaling resource in your cloud, first get the ARN of the resource, as in this example:

  ```
  $ euscale-describe-launch-configs --show-long verbose
  LAUNCH-CONFIG  TestLaunchConfig  emi-06663A57  m1.medium
  2013-10-30T22:52:39.392Z  true
  arn:aws:autoscaling::961915002812:launchConfiguration:5ac29caf-9aad-4bdb-b228-5f
  ce841dc062:launchConfigurationName/TestLaunchConfig
  ```

  Then run the following command with the ARN:

  ```
  euscale-delete-launch-config
  arn:aws:autoscaling::961915002812:launchConfiguration:5ac29caf-9aad-4bdb-b228-5f
  ce841dc062:launchConfigurationName/TestLaunchConfig
  ```

Manage Security

This section details concepts and tasks required to secure your cloud.

Security Overview

This topic is intended for people who are currently using Eucalyptus and who want to harden the cloud and underlying configuration.

This topic covers available controls and best practices for securing your Eucalyptus cloud. Cloud security depends on security across many layers of infrastructure and technology:

- Security of the physical infrastructure and hosts
- Security of the virtual infrastructure
- Security of instances
- Security of storage and data
- Security of users and accounts

Tip: For information about securing applications in AWS cloud, we recommend the Amazon Web Services *AWS Security Best Practices* whitepaper. The practices in this in this paper also apply to your Eucalyptus cloud.

Best Practices

This topic contains recommendations for hardening your Eucalyptus cloud.

Message Security

This topic describes which networking mode is the most secure, and describes how to enforce message security.

Replay Detection

Eucalyptus components receive and exchange messages using either Query or SOAP interfaces (or both). Messages received over these interfaces are required to have a time stamp (as defined by AWS specification) to prevent message replay attacks. Because Eucalyptus enforces strict policies when checking timestamps in the received messages, for the correct functioning of the cloud infrastructure, it is crucial to have clocks constantly synchronized (for example, with ntpd) on all machines hosting Eucalyptus components. To prevent user commands failures, it is also important to have clocks synchronized on the client machines.

Following the AWS specification, all Query interface requests containing the Timestamp element are rejected as expired after 15 minutes of the timestamp. Requests containing the Expires element expire at the time specified by the element. SOAP interface requests using WS-Security expire as specified by the WS-Security Timestamp element.

Replay detection parameters can be tuned as described in *Configure Replay Protection*.

Endpoints

Eucalyptus requires that all user requests (SOAP with WS-Security and Query) are signed, and that their content is properly hashed, to ensure integrity and non-repudiation of messages. For stronger security, and to ensure message confidentiality and server authenticity, client tools and applications should always use SSL/TLS protocols with server certification verification enabled for communications with Eucalyptus components.

By default, Eucalyptus components are installed with self-signed certificates. For public Eucalyptus endpoints, certificates signed by a trusted CA provider should be installed.

Authentication and Access Control

This topic describes best practices for Identity and Access Management and the `eucalyptus` account.

Identity and Access Management

Eucalyptus manages access control through an authentication, authorization, and accounting system. This system manages user identities, enforces access controls over resources, and provides reporting on resource usage as a basis for auditing and managing cloud activities. The user identity organizational model and the scheme of authorizations used to access resources are based on and compatible with the AWS Identity and Access Management (IAM) system, with some Eucalyptus extensions provided that support ease-of-use in a private cloud environment.

For a general introduction to IAM in Eucalyptus, see *Access Concepts* in the Administration Guide. For information about using IAM quotas to enforce limits on resource usage by users and accounts in Eucalyptus, see the *Quotas* section in the Administration Guide.

The *Amazon Web Services IAM Best Practices* are also generally applicable to Eucalyptus.

Credential Management

Protection and careful management of user credentials (passwords, access keys, X.509 certificates, and key pairs) is critical to cloud security. When dealing with credentials, we recommend:

- Limit the number of active credentials and do not create more credentials than needed.
- Only create users and credentials for the interfaces that you will actually use. For example, if a user is only going to use the Management Console, do not create credentials access keys for that user.
- Using `euca_conf --get-credentials` creates access keys and X.509 certificates; avoid unnecessary use of the command and use `euare-useraddkey` and `euare-usercreatecert` or `euare-useraddcert` instead to get a specific set of credentials if needed.
- Regularly check for active credentials using `euare-` commands and remove unused credentials. Ideally, only one pair of active credentials should be available at any time.
- Rotate credentials regularly and delete old credentials as soon as possible. Credentials can be created and deleted using `euare-` commands, such as `euare-useraddkey` and `euare-userdelkey`.
- When rotating credentials, there is an option to deactivate, instead of removing, existing access/secret keys and X.509 certificates. Requests made using deactivated credentials will not be accepted, but the credentials remain in the Eucalyptus database and can be restored if needed. You can deactivate credentials using `euare-usermodkey` and `euare-usermodcert`.

Privileged Roles

The `eucalyptus` account is a super-privileged account in Eucalyptus. It has access to all cloud resources, cloud setup, and management. The users within this account do not obey IAM policies and compromised credentials can result in a complete cloud compromisation that is not easy to contain. We recommend limiting the use of this account and associated users' credentials as much as possible.

For all unprivileged operations, use regular accounts. If you require super-privileged access (for example, management of resources across accounts and cloud setup administration), we recommend that you use one of the predefined privileged roles.

The Account, Infrastructure, and Resource Administrator *roles* provide a more secure way to gain super privileges in the cloud. Credentials returned by an assume-role operation are short-lived (unlike regular user credentials). Privileges available to each role are limited in scope and can be revoked easily by modifying the trust or access policy for the role.

Hosts

This topic describes best practices for machines that host a Eucalyptus component.

Eucalyptus recommends restricting physical and network access to all hosts comprising the Eucalyptus cloud, and disabling unused applications and ports on all machines used in your cloud.

After installation, no local access to Eucalyptus component hosts is required for normal cloud operations and all normal cloud operations can be done over remote web service APIs.

The user-facing services (UFS) and object storage gateway (OSG) are the only two components that generally expect remote connections from end users. Each Eucalyptus component can be put behind a firewall following the list of open ports and connectivity requirements described in the *Configure the Firewall* section.

For more information on securing Red Hat hosts, see the *Red Hat Enterprise Linux Security Guide*. Note that Eucalyptus does not currently support SELinux configurations, and SELinux should be disabled.

Networking Modes

This topic describes the recommendations for networking modes.

We recommend that you use Edge or Managed networking mode, to ensure a secure deployment. They provide security groups, which are used to control inbound traffic to instances, as well as Layer-2 isolation between security groups.

Layer-2 isolation protects traffic within a security group from potential eavesdropping and hijacking by instances that belong to other security groups. In Edge mode, Layer-2 isolation is also enforced between instances within the same security group. For more information about choosing a networking modes, see *Plan Networking Modes* or *Plan Networking Modes* (for HA).

Note that while Edge provides stronger Layer-2 isolation within a security group, it requires NCs to be on the data path to all VMs running on it. It means that all user traffic to VMs has to make it all the way to NCs before it can be blocked, if necessary. This is different from the Managed mode, where all user traffic goes through the CC and can be controlled in a centralized way. This needs to be taken into consideration when choosing between two modes. If Edge mode is selected, we recommend that you have a local firewall on each NC; this allows user traffic only to VMs, but not to the NC itself.

Images and Instances

Because all instances are based on images, creating a secure image helps to create secure instances. This topic lists best practices that will add additional security during image creation. As a general rule, harden your images similar to how you would harden your physical servers.

- Turn off password-based authentication by specifying the following option in `/etc/ssh/sshd_config`:

  ```
  PasswordAuthentication no
  ```

- Encourage non-root access by providing an unprivileged user account. If necessary, use sudo to allow access to privileged commands
- Always delete the shell history and any other potentially sensitive information before bundling. If you attempt more than one bundle upload in the same image, the shell history contains your secret access key.
- Bundling a running instance requires your private key and X.509 certificate. Put these and other credentials in a location that is not bundled (e.g. when using `euca-bundle-vol`, pass the folder location where the certificates are stored as part of the values for the `-e` option). AWS provides more in-depth information on *security considerations in creating a shared machine image*.
- Consider installing *cloud-init* in the image to help control root and non-root access. If cloud-init isn't available, a custom `/etc/rc.local` script can be used.
- Consider using a tool such as *http://manpages.ubuntu.com/manpages/precise/man8/zerofree.8.html*zerofree to zero-out any unused space on the image.
- Consider editing `/etc/rc.local` to clear out the swap every time the instance is booted. This can be done using the following command:

  ```
  sync && /sbin/sysctl vm.drop_caches=3 && swapoff -a && swapon -a
  ```

- Consider enabling *SELinux* or *AppArmor* for your images
- Disable all unused services and ports on the image.
- By default, all images registered have private launch permissions. Consider using `euca-modify-image-attribute` to limit the accounts that can access the image.

After locking down the image using the steps above, additional steps can be done to further secure instances started from that image. For example, restrict access to the instance by allowing only trusted hosts or networks to access ports on your instances. You can control access to instances using `euca-authorize` and `euca-revoke`.

Consider creating one security group that allows external logins and keep the remainder of your instances in a group that does not allow external logins. Review the rules in your security groups regularly, and ensure that you apply the principle of least privilege: only open up permissions as they are required. Use different security groups to deal with instances that have different security requirements.

Management Console

This topic describes things you can do to secure the Eucalyptus Management Console.

- Enable HTTPS for communications with the console and configure the console to use a CA-signed certificate.
- We do not recommend the "Remember my keys" option for "Login to AWS" because it stores AWS credentials in your browser's local storage and increases the security risk of AWS credentials being compromised.
- Change the default session timeouts if needed. For more information, see *Configure Session Timeouts*.
- If you don't use the Management Console, we recommend that you disable `GetAccessToken` (using `euca-modify-property`). For more information, see *Configure STS Actions*.
- Turn off password autocomplete for the console by setting the `browser.password.save` configuration option to false in the console’s configuration file.
- If memcached is configured to be used by the console, make sure it’s not exposed publicly because there is no authentication mechanism enabled out of the box. If the default Eucalyptus-provided configuration is used, it accepts connections only from localhost.

LDAP Security

This topic explains variables in the LIC file you should use to secure configuration.

When you enable LDAP/Active Directory (AD) integration with Eucalyptus, we recommend that you use the following variables in the LDAP/AD Integration Configuration (LIC) file. These variables are located under the `ldap-service` element in the LIC file.

Element	Description
auth-method	The LDAP/AD authentication method to perform synchronization. Supports three types of methods: • *simple*: for clear text user/password authentication. • *DIGEST-MD5*: for SASL authentication using MD5 • *GSSAPI*: SASL authentication using Kerberos V5.
user-auth-method	The LDAP/AD authentication method for normal users to perform Management Console login. Supports three types of methods: • *simple*: for clear text user/password authentication. • *DIGEST-MD5*: for SASL authentication using MD5 • *GSSAPI*: SASL authentication using Kerberos V5.
use-ssl	Specifies whether to use SSL for connecting to LDAP/AD service. If this option is enabled, make sure the SSL port for LDAP is defined as part of the server-url. The default port for LDAP+SSL is port 636.
ignore-ssl-cert-validation	Specifies whether to ignore self-signed SSL certs. This is useful when you only have self-signed SSL certs for your LDAP/AD services.

Element	Description
krb5-conf	The file path for `krb5.conf`, if you use GSSAPI authentication method.

When use-ssl is enabled, ldaps will be used. However, the `server-url` still needs to begin with `ldap://`.

We recommend using a proxy user for the `auth-principal`. Typically, proxy users are used to associate with the application that needs to do reads (and in some cases writes) against the LDAP/AD directory. Proxy users also make it easier for security audits done on the LDAP/AD directory. To use with Eucalyptus and the LDAP/AD sync, the proxy user only needs read access. For more information about using proxy authentication with OpenLDAP and Active Directory, go to the following resources:

- For LDAP: *Using SASL* (see the **SASL Proxy Authorization** section)
- For Active Directory: *Supported Types of Security Principles*
-

For more information about LDAP and security, go to the following resources:

- *Authentication Methods* (see the **"simple" method** section)
- *Using SASL*
- *Security Considerations*

For more information about Active Directory and security, go to the following resources:

- *Simple Authentication*
- *SASL Authentication*
- *LDAP Security*

Tasks

This section details the tasks needed to make your cloud secure.

Configure Managed Mode

To configure managed mode for your cloud, follow the steps in *Configure for Managed Mode* in the Installation Guide.

Configure SSL

In order to connect to Eucalyptus using SSL, you must have a valid certificate for the Cloud Controller (CLC).

Configure SSL for the CLC

This topic details tasks to configure SSL for the CLC.

Important: In a HA environment, repeat these tasks on the other CLC.

Create a Keystore

Eucalyptus uses a PKCS12-format keystore. If you are using a certificate signed by a trusted root CA, perform the following steps.

1. Enter the following command to convert your trusted certificate and key into an appropriate format:

   ```
   openssl pkcs12 -export -in [YOURCERT.crt] -inkey [YOURKEY.key] \
    -out tmp.p12 -name [key_alias]
   ```

   This command will request an export password, which is used in the following steps.
2. Save a backup of the Eucalyptus keystore, at `/var/lib/eucalyptus/keys/euca.p12`.
3. Import your keystore into the Eucalyptus keystore

```
keytool -importkeystore \
-srckeystore tmp.p12 -srcstoretype pkcs12 -srcstorepass [export_password] \
-destkeystore /var/lib/eucalyptus/keys/euca.p12 -deststoretype pkcs12 \
-deststorepass eucalyptus -alias [key_alias] \
-srckeypass [export_password]
```

Enable the CLC to Use the Keystore

To enable the CLC to use the keystore, perform the following steps.

1. Run the following commands on the CLC:

   ```
   euca-modify-property -p bootstrap.webservices.ssl.server_alias=[key_alias]
   euca-modify-property -p \
   bootstrap.webservices.ssl.server_password=[export_password]
   ```

2. Restart the CLC by running `service eucalyptus-cloud restart` or `/etc/init.d/eucalyptus-cloud restart`.

Optional: Redirect Requests

The CLC listens for both SSL and non-SSL connections on port 8773. If you have other tools that expect to speak SSL on port 443, you should forward requests on that port to port 8773. For example, the following iptables command can be used:

```
iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-ports 8773
```

Configure SSL for the UFS

This topic details tasks to configure SSL for the User-Facing Services (UFS).

Important: If you have multiple USF machines, repeat these tasks on each machine.

Create a Keystore

Eucalyptus uses a PKCS12-format keystore. If you are using a certificate signed by a trusted root CA, perform the following steps.

1. Enter the following command to convert your trusted certificate and key into an appropriate format:

   ```
   openssl pkcs12 -export -in [YOURCERT.crt] -inkey [YOURKEY.key] \
    -out tmp.p12 -name [key_alias]
   ```

   This command will request an export password, which is used in the following steps.
2. Save a backup of the Eucalyptus keystore, at `/var/lib/eucalyptus/keys/euca.p12`.
3. Import your keystore into the Eucalyptus keystore

   ```
   keytool -importkeystore \
   -srckeystore tmp.p12 -srcstoretype pkcs12 -srcstorepass [export_password] \
   -destkeystore /var/lib/eucalyptus/keys/euca.p12 -deststoretype pkcs12 \
   -deststorepass eucalyptus -alias [key_alias] \
   -srckeypass [export_password]
   ```

Enable the UFS to Use the Keystore

To enable the UFS to use the keystore, perform the following steps.

1. Run the following commands on the UFS:

   ```
   euca-modify-property -p bootstrap.webservices.ssl.server_alias=[key_alias]
   euca-modify-property -p \
   bootstrap.webservices.ssl.server_password=[export_password]
   ```

2. Restart the UFS by running `service eucalyptus-cloud restart` or `/etc/init.d/eucalyptus-cloud restart`.

Optional: Redirect Requests

The UFS listens for both SSL and non-SSL connections on port 8773. If you have other tools that expect to speak SSL on port 443, you should forward requests on that port to port 8773. For example, the following iptables command can be used:

```
iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-ports 8773
```

Configure and Enable SSL for the Management Console

You can use secure HTTP for your console.

To run your console over Secure HTTP:

1. Install nginx on your console server with the following command:

   ```
   yum install nginx
   ```

2. Overwrite the default `nginx.conf` file with the template provided in `/usr/share/doc/eucaconsole-4.1.2/nginx.conf`.

   ```
   cp /usr/share/doc/eucaconsole-4.1.2/nginx.conf /etc/nginx/nginx.conf
   ```

3. Uncomment the 'listen' directive and uncomment/modify the SSL certificate paths in `/etc/nginx/nginx.conf` (search for "SSL configuration"). For example:

   ```
   # SSL configuration
   listen 443 ssl;
   # ssl_certificate /path/to/ssl/pem_file;
   # EXAMPLE:
   ssl_certificate /etc/eucaconsole/console.crt;
   # ssl_certificate_key /path/to/ssl/certificate_key;
   # EXAMPLE:
   ssl_certificate_key /etc/eucaconsole/console.key;
   # end of SSL configuration
   ```

   Tip: For more information on generating self-signed SSL certificates, go to *http://www.akadia.com/services/ssh_test_certificate.html*.

4. Restart nginx using the following command:

   ```
   service nginx restart
   ```

5. Edit the `/etc/eucaconsole/console.ini` file, locate the `session.secure = false` parameter, change `false` to `true`, then add the `sslcert` and `sslkey` lines immediately following, per this example:

   ```
   session.secure = true
   sslcert=/etc/eucaconsole/eucalyptus.com.chained.crt
   sslkey=/etc/eucaconsole/eucalyptus.com.key
   ```

Configure SSL for LDAP

This topic details tasks required to configure SSL for LDAP.

To configure SSL for LDAP, make the following edits to your LIC template or file.

Tip: For more information about the LIC template and file, see *LDAP/AD Integration Configuration*.

1. Edit the `use-ssl` value to `true`.

   ```
   "use-ssl":"true",
   ```

2. Edit the `ignore-ssl-cert-validation` value to `false`.

   ```
   "ignore-ssl-cert-validation":"false",
   ```

Synchronize Components

To synchronize your Eucalyptus component machines with an NTP server, perform the following tasks.

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

```
# ntpdate pool.ntp.org
# service ntpd start
# chkconfig ntpd on
# ps ax | grep ntp
# hwclock --systohc
```

2. Repeat for each machine hosting a Eucalyptus component.

Configure Replay Protection

You can configure replay detection in Java components (which includes the CLC, UFS, OSG, Walrus, and SC) to allow replays of the same message for a set time period. You might need this to ensure that legitimate requests submitted by automated scripts (such as two requests to describe instances issued within the same second) are not rejected as malicious.

Important: To protect against replay attacks, the Java components cache messages only for 15 minutes. So it's important that any client tools used to interact with the components have the Expires element set to a value less than 15 minutes from the current time. This is usually not an issue with standard tools, such as euca2ools and Amazon EC2 API Tools.

1. The Java components' replay detection algorithm rejects messages with the same signatures received within 15 minutes. The time within which messages with the same signatures are accepted is controlled by the `bootstrap.webservices.replay_skew_window_sec` property. The default value of this property is 3 seconds. To change this value, enter the following command:

```
euca-modify-property -p
bootstrap.webservices.replay_skew_window_sec=[new_value_in_seconds]
```

   If you set this property to `0`, Eucalyptus will not allow any message replays. This setting provides the best protection against message replay attacks, but may break some of the client-side scripts that issue commands too quickly.

   If you set this property to any value greater than 15 minutes plus the values of ws.clock_skew_sec (that is, to a value >= 920 sec in the default installation), Eucalyptus disables replay detection completely.

2. When checking message timestamps for expiration, Eucalyptus allows up to 20 seconds of clock drift between the machines. This is a default setting. You can change this value for the Java components at runtime by setting the `bootstrap.webservices.clock_skew_sec` property as follows:

```
euca-modify-property -p
bootstrap.webservices.clock_skew_sec=[new_value_in_seconds]
```

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 8080	Port for getting user credentials on the CLC. Configurable with `euca-modify-property`.
TCP 8772	DEBUG ONLY: JMX port. This is disabled by default, and can be enabled with the `--debug` or `--jmx` options for `CLOUD_OPTS`.
TCP 8773	Web services port for the CLC, user-facing services (UFS), object storage gateway (OSG), Walrus SC; also used for external and internal communications by the CLC and Walrus. Configurable with `euca-modify-property`.

Port	Description
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.init` file.
TCP 16514	TLS port on Node Controller, required for node migrations
UDP 7500	Port for diagnostic probing on CLC, UFS, OSG, Walrus SC
UDP 8773	HA membership port
UDP 8778	The bind port used to establish multicast communication
TCP/UDP 53	DNS port on UFS

Configure the Firewall

This topic provides guidelines for restricting network access and managing iptables rules.

Restricting Network Access

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

On the Cloud Controller (CLC), Walrus, and Storage Controller (SC), allow for the following jGroups traffic:

- TCP connections between CLC, user-facing services (UFS), object storage gateway (OSG), Walrus, and SC on port 8779 (or the first available port in range 8779-8849)
- UDP connections between CLC, UFS, OSG, Walrus, and SC on port 7500
- Multicast connections between CLC, UFS, OSG, Walrus, and SC to IP 228.7.7.3 on UDP port 8773

On the UFS, allow the following connections:

- TCP connections from end-users and instances on ports 8773
- End-user and instance connections to DNS ports

On the CLC, allow the following connections:

- TCP connections from UFS, CC and Eucalyptus instances (public IPs) on port 8773 (for metadata service)
- TCP connections from UFS, OSG, Walrus, and SC on port 8777

On the CC, make sure that all firewall rules are compatible with the dynamic changes performed by Eucalyptus, described in the section below. Also allow the following connections:

- TCP connections from CLC on port 8774

On OSG, allow the following connections:

- TCP connections from end-users and instances on port 8773
- TCP connections from SC and NC on port 8773

On Walrus, allow the following connections:

- TCP connections from OSG on port 8773

On the SC, allow the following connections:

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

On the NC, allow the following connections:

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

Managing iptables Rules for the CC

In Managed and Managed (No VLAN) modes, Eucalyptus flushes the CC's iptables rules for both `filter` and `nat`, then it sets the default policy for the `FORWARD` chain in `filter` to `DROP`. At run time, the CC adds and removes rules from `FORWARD` as users add and remove ingress rules from their active security groups. In addition, the `nat` table is configured to allow VMs access to the external network using IP masquerading, and dynamically adds/removes rules in the `nat` table as users assign and unassign public IPs to VMs at instance boot or run-time.

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

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

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

Configure Session Timeouts

To set the session timeouts in the Management Console:

Modify the `session.timeout` and `session.cookie_expires`entries in the `[app:main]` section of the configuration file. The `session.timeout` value defines the number of seconds before an idle session is timed out. The `session.cookie_expires` is the maximum length that any session can be active before being timed out. All values are in seconds:

```
session.timeout=1800
```

```
session.cookie_expires=43200
```

Start a LIC File

The LIC is a file in JSON format and specifies what Eucalyptus needs for synchronizing with an LDAP or AD service. Eucalyptus provides a LIC template at `${EUCALYPTUS}/usr/share/eucalyptus/lic_template`. This template shows all the fields of the LIC, and provides detailed documentation and example values for each field.

To start a LIC file:

1. Enter the following command:

   ```
   /usr/sbin/euca-lictool --password secret --out example.lic
   ```

   This command tells the LIC tool to create a template LIC and fill in the encrypted password for authenticating to LDAP/AD service (that is, the password of the administrative user for accessing the LDAP/AD during synchronization). The LIC tool’s primary functions are to encrypt the LDAP/AD password and to generate the starting LIC template. The usage of the LIC tool shows different ways to invoke the command.
2. Once you have the LIC template, fill in the details by editing the `*.lic` file using a text editor. Each top level entity specifies one aspect of the LDAP/AD synchronization.

Configure STS Actions

The Security Token Service (STS) allows you to enable or disable specific token actions.

By default, the enabled actions list is empty. However, this means that all actions are enabled. To disable actions, list each action in the `disabledactions` property. To enable specific actions, list them in the `enabledactions` property.

```
# euca-describe-properties tokens
PROPERTY tokens.disabledactions {}
PROPERTY tokens.enabledactions {}
```

The values for each property are case-insensitive, space or comma-separated lists of token service actions. If an action is in the disable list it will not be permitted. Eucalyptus returns an HTTP status 503 and the code `ServiceUnavailable`.

If the enable list is not empty, Eucalyptus only permits the actions specifically listed.

Action	Description
AssumeRole	Roles as per AWS/STS and Eucalyptus-specific personas admin functionality
GetAccessToken	Eucalyptus extension for password logins (for example, the Management Console)
GetImpersonationToken	Eucalyptus extension that allows cloud administrators to act as specific users
GetSessionToken	Session tokens in the sameas per AWS/STS

For more information about STS, go to *STS section of the AWS CLI Reference*.

Manage Reporting

Eucalyptus provides two ways for getting metrics for your cloud: you can get a report directly from the Cloud Controller (CLC), or you can get a report from data exported from the CLC and imported to a data warehouse.

When you install Eucalyptus, you automatically get the reporting system in place to generate reports from the CLC. However, the down side to using the CLC for reports is latency. Because of this, Eucalyptus also supports a data warehouse that resides outside the Eucalyptus system to store report data.

This section describes the concepts and best practices for Eucalyptus reporting, and how to generate reports.

Reporting Overview

Eucalyptus lets you generate reports to monitor cloud resource use. Each type of report is for a specified time range.

Eucalyptus supports the following report types:

- **Instance:** The instance report provides information about the amount, duration, and utilization of all running instances. Use this report to understand how many instances each user is running, whether your instance types are large enough, etc.
- **S3:** The S3 report provides information about the number of buckets and objects stored in Walrus. Empty buckets are not reported. Use this report to understand the storage needs of each user and your cloud's storage needs.
- **Volume:** The volume report provides information about the amount, duration, and size of all volumes in use. Use this report to understand how many volumes are running, and what the storage size of each volume is.
- **Snapshot:** The snapshot report provides information about the amount of your cloud's snapshots. Use this report to understand how many snapshots there are and from which volumes, and what the size of each snapshot is.
- **Elastic IP:** The elastic IP report provides information about the lifecycle of elastic IPs in your cloud, including which user is using which IPs, which IPs are currently in use, and how often and for how long does IP get allocated. Use this report to understand how many IPs each user is assigned and to which instance the IP is assigned to, and the running time of each IP.
- **Capacity:** The capacity report provides overall information about your cloud's resources, including instance types and storage. Use this report to determine if your resources are being used adequately, and whether you need to scale up or down.

You can generate reports in either CSV or HTML formats for use with external tools.

If you want to use the CLC for your reports, see *Reporting Tasks: CLC*.

If you want to use the data warehouse for your reports, see *Set Up the Data Warehouse*.

Understanding the Report Format

All Eucalyptus reports contain a usage section. The instance report also contains a running time section.

The usage section shows cumulative (**cumul.**) metrics for each zone, account, and user. Then the report lists metrics for each resource. The column for each resource type (for example, **Instance Id** or **Volume Id** displays **cumul.** for all cumulative metrics. When individual resources are reported, the individual resource's name or identifier displays in that column.

Instance Report

The Instance Report has the following column headings:

Heading	Description
Net Total GB In	Total instance network input communication between instances with in the cloud

Heading	Description
Net Total GB Out	Total instance network output communication between instances with in the cloud
Net External GB In	Total instance network input communication between connections from outside of the cloud
Net External GB Out	Total instance network output communication between connections from outside of the cloud
Disk GB Read	Total instance disk reads
Disk GB Write	Total instance disk writes
Disk IOPS (M) Read	Disk read transfer rate and I/Os per second
Disk IOPS (M) Write	Disk write transfer rate and I/Os per second
Disk Time (hrs) Read	Total disk read time per hour
Disk Time (hrs) Write	Total disk write time per hour

S3 Report

The S3 Report has the following column headings:

Heading	Description
Bucket	Name of the container used to store objects
# Objects	Total number of objects created
# Snap	Total number of snapshots created
Total Obj Size (BYTES)	Total object size in bytes
Obj GB-Days	Object size reporting interval, in gigabytes

Volume Report

The S3 Report has the following column headings:

Heading	Description
Instance Id	Identifier of the instance
Volume Id	Identifier of the Eucalyptus block volume attached to the instance
# Vol	Total number of volumes created
Size (BYTES)	Size of the volumes, in bytes
GB-Days	Gigabytes used per day

Snapshot Report

The Snapshot Report has the following column headings:

Heading	Description
Volume Id	Identifier of the Eucalyptus block volume

Heading	Description
Snapshot Id	Identifier of the snapshot
# Snap	Total number of snapshots created
Size (BYTES)	Size of the snapshots, in bytes
GB-Days	Gigabytes used per day

Elastic IP Report

The Elastic IP Report has the following column headings:

Heading	Description
Elastic IP	IP address
Instance ID	Identifier of the instance that is assigned the elastic IP
# IPs	Number of IPs used by a user(s)
Duration	Length in time that the elastic IP is in use by an instance

Capacity Report

The Capacity Report has the following column headings:

Heading	Description
Resource	The resource whose capacity is being reported. A resource can be: • S3 Storage in GB • Elastic IP count • EBS Storage in GB • EC2 Compute in cores • EC2 Disk in GB • EC2 Memory in MB • VM Types by type (for example, “c1.medium”) count
Available	Quantity of the resource free for use
Total	Total available quantity for the resource

Reporting Best Practices

This topic provides guidelines for using the reporting feature in Eucalyptus.

- Eucalyptus recommends that you run reports from the data warehouse. The Cloud Controller (CLC) generates the data. The data warehouse is a store of the stale data exported from the CLC.
- Monitor the rate of information collected and written to the CLC database. The database expands through usage and event-driven records. More report information stored in the CLC database lessens the effectiveness of the CLC to perform its cloud duties. If the database gets too large, export the data to the data warehouse then delete the data from the CLC.
- Be careful about deleting data in the CLC. If you delete data in the CLC after you export it, you should use the data warehouse to generate all future reports. This ensures that you have a comprehensive picture of your cloud data.
- You can't import data from different clouds into the same data warehouse.

Reporting Tasks

This section explains the tasks associated with the Eucalyptus reporting feature.

Setting up a data warehouse allows you to remove data from the Cloud Controller (CLC). This ensures that you have enough disk space to operate the CLC. This section contains information needed to install the data warehouse and run those reports.

Once the data warehouse is installed, the workflow for running reports against the data warehouse is:

1. Export the data from the CLC. For more information, see *Export Data*.
2. Import the data to the data warehouse. For more information, see *Import Data*.
3. Create the report from the data in the data warehouse. For more information, see *Create a Report: Data Warehouse*.

Set Up the Data Warehouse

This section explains how to set up the data warehouse and how to generate reports using data in the data warehouse.

Install the Data Warehouse

To install the Data Warehouse on hosts running RHEL 6 or CentOS 6:

Important: Do not install the Data Warehouse on a machine running Eucalyptus services.

1. Configure the Eucalyptus package repository on the Data Warehouse host:

```
yum --nogpgcheck install
http://downloads.eucalyptus.com/software/eucalyptus/4.1/centos/6/x86_64/
eucalyptus-release-4.1.el6.noarch.rpm
```

2. Install the Data Warehouse packages:

```
yum install eucadw
```

3. Install the PostgreSQL server:

```
yum install postgresql91-server
```

You are now ready to *Configure the Database*.

Configure the Database

To configure the database in your data warehouse perform the tasks

1. Initialize the PostgreSQL database.

```
service postgresql-9.1 initdb
```

2. Start the PostgreSQL service.

```
service postgresql-9.1 start
```

3. Log in to the PostgreSQL server.

```
su - postgres
```

4. Start the PostgreSQL terminal.

```
psql
```

5. At the psql prompt run:

```
create database eucalyptus_reporting;
create user eucalyptus with password 'mypassword';
```

```
grant all on database eucalyptus_reporting to eucalyptus;
\q
```

6. Log out.

```
exit
```

7. Edit the `/var/lib/pgsql/9.1/data/pg_hba.conf` file to contain the following content:

```
local   all             all                                     password
host    all             all             127.0.0.1/32            password
host    all             all             ::1/128                 password
```

8. Reload the PostgreSQL service.

```
service postgresql-9.1 reload
```

Your machine is now configured as a data warehouse.

Check the Data Warehouse Status

To check the data warehouse status perform the steps listed in this topic.

Enter the following command:

```
eucadw-status -p <your_password>
```

For more information about `eucadw-status`, go to the *Euca2ools Reference Guide*.

Export Data

To export data from the Cloud Controller (CLC):

Run the following command:

```
eureport-export-data [filename] -s [start_date] -e [end_date]
  -d
```

For more information about the `eureport-export-data` command, go to the *Euca2ools Reference Guide*.

Import Data

To import data into the data warehouse:

Run the following command:

```
eucadw-import-data -e [filename] -p [your_password]
```

where `filename` is the name of the imported file that you want to get data from.

For more information about `eucadw-import-data`, go to the *Euca2ools Reference Guide*.

Create a Report: Data Warehouse

To create a report from data in the data warehouse:

Run the following command:

```
eucadw-generate-report -s <start_date> -e <end_date> -t <report_type> -p
<your_password
```

where:

- `start_date` is the date you want data from. For example, `2012-11-05`.
- `end_date` is the date you want data to.

- `report_type` is the type of report you want to run: `instance`, `S3`, `volume`, `snapshot`, `IP`, or `capacity`.
- `your_password` is the administrator password you configured in the data warehouse installation.

For more information about `eucadw-generate-report`, go to the *Euca2ools Reference Guide*.

Eucalyptus Commands

This section contains reference information for Eucalyptus administration and reporting commands.

Eucalyptus Administration Commands

Eucalyptus offers commands for common administration tasks and inquiries. This section provides a reference for these commands.

euca_conf

This is the main configuration file for Eucalyptus.

Syntax

```
euca_conf
```

Options

Option	Description	Required
`--initialize`	Begin the one-time initialization of the CLC	No
`--heartbeat`	Return heartbeat data for the specified host	No
`--synckey`		No
`--no-rsync`	Do not use rsync when registering	No
`--no-scp`	Do not use scp when registering	No
`--skip-scp-hostcheck`	Skip scp interactive host keycheck	No
`--get-credentials`	Download credentials to the specified zip file. By default, the admin credentials will be downloaded but this can be adjusted with the `--cred-user` option. Each time this is called, new X.509 certificates will be created for the specified user.	No
`--cred-account`	Set `get-credentials` for the specified account	No
`--cred-user`	Set `get-credentials` for the specified user	No
`--register-nodes`	Add specified NCs to Eucalyptus	No
`--deregister-nodes`	Remove specified NC from Eucalyptus	No
`--register-arbitrator`	Add arbitrator service to Eucalyptus	No
`--deregister-arbitrator`	Remove arbitrator service from Eucalyptus	No
`--register-cloud`	Add new Cloud Controller to Eucalyptus	No
`--register-cluster`	Add a Cluster Controller to Eucalyptus	No
`--deregister-cluster`	Remove a Cluster Controller from Eucalyptus	No
`--register-walrus`	Add Walrus to Eucalyptus	No
`--deregister-walrus`	Remove Walrus from Eucalyptus	No
`--register-sc`	Add Storage Controller to Eucalyptus	No

Option	Description	Required
`--deregister-sc`	Remove Storage Controller from Eucalyptus	No
`--list-walruses`	Return all registered Walruses	No
`--list-clouds`	Return all registered Cloud Controllers	No
`--list-clusters`	List all registered Cluster Controllers	No
`--list-arbitrators`	Return all registered arbitrator services	No
`--list-nodes`	Return all registered Node Controllers	No
`--list-components`	return all registered Eucalyptus components	No
`--list-services`	Return all registered services	No
`-list-scs`	Return all registered Storage Controllers	No
`--no-sync`	Used with `--register-*` to skip syncing keys	No
`-d`	Point Eucalyptus to the specified directory	No
`--cc-port`	Set the Cluster Controller to the specified port	No
`--sc-port`	Set the Storage Controller to the specified port	No
`--walrus-port`	Set Walrus to the specified port	No
`--nc-port`	Set the Node Controller to the specified port	No
`--instances`	Set the instance path	No
`--hypervisor`	Set which hypervisor to use. Valid values: `xen` | `kvm`	No
`--user`	Set the user to use for EUCA_USER	No
`--dhcpd`	Set the DHCP daemon binary to the specified path	No
`--dhcp_user`	Set the specified user name to run dhcpd as	No
`--bridge`	Set the bridge as the specified name	No
`--name`	Returns the value for the specified name	No
`--import-conf`	Import variables from a specified `eucalyptus.conf` file	No
`--upgrade-conf`	Upgrade `eucalyptus.conf` from the specified older installation file	No
`--setup`	Perform initial setup	No
`--enable`	Enable specified service at next start Valid values: `cloud` | `walrus` | `sc`	No
`--disable`	Disable specified service at next start Valid values: `cloud` | `walrus` | `sc`	No
`--check`	Pre-flight checks Valid values: `common`	No
`-P, --partition`	Name of partition. Used with `--register-*` and `--deregister-*`	No

Option	Description	Required
`-H, --host`	Name or IP address of host. Used with `--register-*`	No
`-C, --component`	Name of the component. Used with `--register-*` and `--deregister-*`	No
`--help-register`	Display help for register deregister	No

Common Options

Option	Description
`--show-empty-fields`	Show empty fields as "(nil)".
`--region` *`user@region`*	Region and/or user name to search when looking up config file data. Only valid for EC2 endpoints.
`-U,--url` *`url`*	URL of the cloud service to connect to. For administrative commands, this should be `<ip_address>:8773/services/Empyrean`.
`-I,--access-key-id` *`key_id`*	User's access key ID.
`-S,--secret-key` *`secret_key`*	User's secret key.
`--security-token` *`token`*	User's security token.
`--debug`	Prints what the command sends to the server and what it receives from the server. Use when you're trying to debug Euca2ools.
`--debugger`	Enable interactive debugger on error.
`-h,--help`	Display the manual page for the command.
`--version`	Display the version of this tool.

euca-describe-properties

This command lists properties.

Syntax

```
euca-describe-properties
```

Options

None

Common Options

Option	Description
`--show-empty-fields`	Show empty fields as "(nil)".
`--region` *`user@region`*	Region and/or user name to search when looking up config file data. Only valid for EC2 endpoints.
`-U,--url` *`url`*	URL of the cloud service to connect to. For administrative commands, this should be `<ip_address>:8773/services/Empyrean`.

Option	Description
`-I,--access-key-id` *`key_id`*	User's access key ID.
`-S,--secret-key` *`secret_key`*	User's secret key.
`--security-token` *`token`*	User's security token.
`--debug`	Prints what the command sends to the server and what it receives from the server. Use when you're trying to debug Euca2ools.
`--debugger`	Enable interactive debugger on error.
`-h,--help`	Display the manual page for the command.
`--version`	Display the version of this tool.

euca-modify-property

This command modifies a Eucalyptus cloud property.

Syntax

```
euca-modify-property
```

Options

Option	Description	Required
`-p, --property` *name=value*	Set the named property to the specified value.	Conditional
`-r` *name*	Resets the named property to the default value.	No

Common Options

Option	Description
`--show-empty-fields`	Show empty fields as "(nil)".
`--region` *`user@region`*	Region and/or user name to search when looking up config file data. Only valid for EC2 endpoints.
`-U,--url` *`url`*	URL of the cloud service to connect to. For administrative commands, this should be `<ip_address>:8773/services/Empyrean`.
`-I,--access-key-id` *`key_id`*	User's access key ID.
`-S,--secret-key` *`secret_key`*	User's secret key.
`--security-token` *`token`*	User's security token.
`--debug`	Prints what the command sends to the server and what it receives from the server. Use when you're trying to debug Euca2ools.
`--debugger`	Enable interactive debugger on error.

Option	Description
`-h,--help`	Display the manual page for the command.
`--version`	Display the version of this tool.

euca-describe-services

This command returns information about all running services.

Syntax

```
euca-describe-services
```

Options

Option	Description	Required
`-A, --all`	Include all public service information. Reported state information is determined by the view available to the target host, which should be treated as advisory (See documentation for guidance on interpreting this information).	No
`--system-internal`	Include internal services information **Note:** This information is only for the target host.	No
`--user-services`	Include services that are user-facing and co-located with some other top-level service **Note:** This information is only for the target host.	No
`-T, --filter-type`	Filter services by specified component type	No
`-H, --filter-host`	Filter services by specified host	No
`-F, --filter-state`	Filter services by state	No
`-P, --filter-partition`	Filter services by specified partition	No
`-E, --events`	Return service event details	No
`-events-verbose`	Return verbose service event details	No

Common Options

Option	Description
`--show-empty-fields`	Show empty fields as "(nil)".
`--region` *`user@region`*	Region and/or user name to search when looking up config file data. Only valid for EC2 endpoints.
`-U,--url` *`url`*	URL of the cloud service to connect to. For administrative commands, this should be `<ip_address>:8773/services/Empyrean`.

Option	Description
`-I,--access-key-id` *`key_id`*	User's access key ID.
`-S,--secret-key` *`secret_key`*	User's secret key.
`--security-token` *`token`*	User's security token.
`--debug`	Prints what the command sends to the server and what it receives from the server. Use when you're trying to debug Euca2ools.
`--debugger`	Enable interactive debugger on error.
`-h,--help`	Display the manual page for the command.
`--version`	Display the version of this tool.

Eucalyptus Report Commands

Eucalyptus lets you to generate reports for your cloud. These reports show data useful for understanding how your resources are being allocated, who is using the resources, and how much time resources are running.

Eucalyptus lets you to get reports from either the Cloud Controller (CLC) or the data warehouse. Reports from the data warehouse are from data exported from the CLC.

Commands that begin the `eureport-` are for the CLC. For more information, see *Reports Commands: CLC*. Commands that begin with `eucadw-` are for the data warehouse. For more information, see *Report Commands: Data Warehouse*.

Reports Commands: CLC

This section contains reference information for reporting commands that use the Cloud Controller (CLC).

Normally, you will just use *eureport-generate-report* command. If you want to run reports against the data warehouse, you need to export data from the CLC using the *eureport-export-data* command.

Caution: Be careful if you use the `eureport-delete-data` command. Once you delete data from the CLC, you have to run reports using the data warehouse. You can't use the CLC for reporting.

eureport-generate-report

Generates a report from the CLC.

Syntax

```
eureport-generate-report [filename] [-t report_type]
     [-f report_format] [-s start_date] [-e end_date]
     [--size-unit size_unit] [--time-unit time_unit]
     [--size-time-size-unit size_time_size_unit]
     [--size-time-time-unit size_time_time_unit] [-d] [-F]
```

Options

Option	Description	Required
filename	Path to the resulting reporting file.	No

Option	Description	Required
`-t, --type` *`report_type`*	Type of report to generate. Valid values: `elastic-ip` | `instance` | `s3` | `snapshot` | `volume` Default: `instance`	No
`-f, --format` *`report_format`*	Format of report generate. Valid values: `csv` | `html` Default: `html`	No
`-s, --start-date` *`start_date`*	Inclusive start date for the exported data in YYYY-MM-DD format. For example, `2012-08-19`.	Yes
`-e, --end-date` *`end_date`*	Exclusive end date for the exported data in YYYY-MM-DD format. For example, `2012-08-26`.	Yes
`--size-unit` *`size_unit`*	The level of granularity for reporting metrics by size alone. Valid values: `b` | `kb` | `mb` | `gb` Default: `gb`	No
`--time-unit` *`time_unit`*	The level of granularity for reporting interval. Valid values: `seconds` | `minutes` | `hours` | `days` Default: `days`	No
`--size-time-size-unit` *`size_time_size_unit`*	The level of granularity for reporting size metrics for the time interval. Valid values: `b` | `kb` | `mb` | `gb` Default: `gb`	No
`--size-time-time-unit` *`size_time_time_unit`*	The level of granularity for reporting size metrics for the time interval. Valid values: `seconds` | `minutes` | `hours` | `days` Default: `days`	No
`-d, --dependencies`	Include event dependencies from outside the requested time period.	No
`-F, --force`	Overwrite output file if it exists.	No

Common Options

Option	Description
`--show-empty-fields`	Show empty fields as "(nil)".
`--region` *`user@region`*	Region and/or user name to search when looking up config file data. Only valid for EC2 endpoints.
`-U,--url` *`url`*	URL of the cloud service to connect to. For administrative commands, this should be `<ip_address>:8773/services/Empyrean`.
`-I,--access-key-id` *`key_id`*	User's access key ID.
`-S,--secret-key` *`secret_key`*	User's secret key.

Option	Description
`--security-token` *token*	User's security token.
`--debug`	Prints what the command sends to the server and what it receives from the server. Use when you're trying to debug Euca2ools.
`--debugger`	Enable interactive debugger on error.
`-h,--help`	Display the manual page for the command.
`--version`	Display the version of this tool.

Output

Eucalyptus returns a message stating that report was generated to the file you specified.

Example

```
eureport-generate-report -s 2012-11-05 -e 2012-11-07 --size-unit=b
--size-time-size-unit=b -t instance Report2.html
Exported data to Report2.html
```

eureport-delete-data

Deletes report data generated before a specified date.

Syntax

```
eureport-delete-data -s start_date -e end_date
  [-d] [filename] [-F]
```

Options

Option	Description	Required
`-s, --start-date` *start_date*	Inclusive start date for the deleted report data in YYYY-MM-DD format. For example, `2012-08-19`.	Yes
`-e, --end-date` *end_date*	Exclusive end date for the deleted report data. For example, `2012-08-26`.	Yes
`-d, --dependencies`	Include event dependencies from outside the requested time period.	No
filename	Path to the reporting data export file	No
`-F, --force`	Overwrite output file if it exists.	No

Common Options

Option	Description
`--show-empty-fields`	Show empty fields as "(nil)".
`--region` *user@region*	Region and/or user name to search when looking up config file data. Only valid for EC2 endpoints.
`-U,--url` *url*	URL of the cloud service to connect to. For administrative commands, this should be `<ip_address>:8773/services/Empyrean`.

Option	Description
`-I,--access-key-id` *`key_id`*	User's access key ID.
`-S,--secret-key` *`secret_key`*	User's secret key.
`--security-token` *`token`*	User's security token.
`--debug`	Prints what the command sends to the server and what it receives from the server. Use when you're trying to debug Euca2ools.
`--debugger`	Enable interactive debugger on error.
`-h,--help`	Display the manual page for the command.
`--version`	Display the version of this tool.

Output

Eucalyptus returns a message detailing the number of data entries it deleted.

Example

```
eureport-delete-data -e 2012-11-06
Deleted 153415 reporting data entries.
```

eureport-export-data

Exports report data to a file. This file can be imported into the data warehouse.

Syntax

```
eureport-export-data [filename] -s start_date -e end_date
  [-d] [-F]
```

Options

Option	Description	Required
filename	Path to the resulting reporting data export file	No
`-s, --start-date` *`start_date`*	Inclusive start date for the exported data in YYYY-MM-DD format. For example, `2012-08-19`.	Yes
`-e, --end-date` *`end_date`*	Exclusive end date for the exported data in YYYY-MM-DD format. For example, `2012-08-26`.	Yes
`-d, --dependencies`	Include event dependencies from outside the requested time period.	No
`-F, --force`	Overwrite output file if it exists.	No

Common Options

Option	Description
`--show-empty-fields`	Show empty fields as "(nil)".
`--region` *`user@region`*	Region and/or user name to search when looking up config file data. Only valid for EC2 endpoints.

Option	Description
`-U,--url` *`url`*	URL of the cloud service to connect to. For administrative commands, this should be `<ip_address>:8773/services/Empyrean`.
`-I,--access-key-id` *`key_id`*	User's access key ID.
`-S,--secret-key` *`secret_key`*	User's secret key.
`--security-token` *`token`*	User's security token.
`--debug`	Prints what the command sends to the server and what it receives from the server. Use when you're trying to debug Euca2ools.
`--debugger`	Enable interactive debugger on error.
`-h,--help`	Display the manual page for the command.
`--version`	Display the version of this tool.

Output

Eucalyptus returns a message stating that the data was exported to the file you specified.

Example

```
eureport-export-data -s 2012-11-05 -e 2012-11-07 -F iReport.dat
Exported data to iReport.dat
```

Report Commands: Data Warehouse

This section contains the reference for reporting commands that use the data warehouse.

The workflow for reporting against the data is the data warehouse is as follows:

1. Export data from the Cloud Controller (CLC) using the *eureport-export-data* command.
2. Import the data into the data warehouse using the *eucadw-import-data* command.
3. Run a report using the *eucadw-generate-report* command.

eucadw-status

Checks for a connection to the data warehouse and for available data stored in the data warehouse.

Syntax

```
eucadw-status -p password
```

Options

Option	Description	Required
`-p,` *`password`*	Administrator password you configured in the data warehouse installation.	Yes

Common Options

None.

Output

Eucalyptus returns the connection status.

Examples

The following example shows a successful connection.

```
eucadw-status -p mypassword
Connected to database: localhost:8777/reporting as eucalyptus
Data present from 2012-05-27 22:25:01 to 2012-09-24 22:58:01
```

The following example shows an unsuccessful connection.

```
eucadw-status -p mypassword
Database access failed with the following details.
SQLState  : 3D000
Error Code: 0
FATAL: database "blah" does not exist
```

eucadw-import-data

Imports data into the data warehouse. This data is in a specified file that has first been generated from the `eureport-export-data` command.

Syntax

```
eucadw-import-data -e filename -p password[-r]
```

Options

Option	Description	Required
`-e, --export` *`export_filename`*	Name of the export file you want to import into the data warehouse.	Yes
`-p,` *`password`*	Administrator password you configured in the data warehouse installation.	Yes
`-r, --replace`	Use this option if you want to replace an existing file that has the same name as the file you are importing.	No

Common Options

None.

Output

Eucalyptus returns a message detailing the number of entries imported and the timefrome of those entries.

Example

```
eucadw-import-data -e iReport.dat -p mypassword
Imported 45 entries from 2012-11-07 23:08:17 to 2012-11-07 23:37:59
```

eucadw-generate-report

Generates a report from the data warehouse.

Syntax

```
eucadw-generate-report -p password[filename]
     [-t report_type] [-f report_format] [-s start_date]
```

```
    [-e end_date] [--size-unit size_unit]
    [--time-unit time_unit]
    [--size-time-size-unit size_time_size_unit]
    [--size-time-time-unit size_time_time_unit] [-d] [-F]
```

Options

Option	Description	Required
`-p, password`	Administrator password you configured in the data warehouse installation.	Yes
`filename`	Name of the file to output report data to. If you do not enter a filename, Eucalyptus generates report data to the console.	No
`-t, --type report_type`	Type of report to generate. Valid values: `elastic-ip` | `instance` | `s3` | `snapshot` | `volume` Default: `instance`	No
`-f, --format report_format`	Format of report generate. Valid values: `csv` | `html` Default: `html`	No
`-s, --start_date start_date`	Inclusive start date for the exported data in YYYY-MM-DD format. For example, `2012-08-19`. Default: `html`	No
`-e, --end-date end_date`	Exclusive end date for the exported data in YYYY-MM-DD format. For example, `2012-08-26`.	Yes
`--size-unit size_unit`	The level of granularity for reporting metrics by size alone. Valid values: `b` | `kb` | `mb` | `gb` Default: `gb`	No
`--time-unit time_unit`	The level of granularity for reporting interval. Valid values: `seconds` | `minutes` | `hours` | `days` Default: `days`	No
`--size-time-size-unit size_time_size_unit`	The level of granularity for reporting size metrics for the time interval. Valid values: `b` | `kb` | `mb` | `gb` Default: `gb`	No
`--size-time-time-unit size_time_time_unit`	The level of granularity for reporting size metrics for the time interval. Valid values: `seconds` | `minutes` | `hours` | `days` Default: `DAYS`	No
`-d, --dependencies`	Include event dependencies from outside the requested time period.	No
`-F, --force`	Overwrite output file if it exists.	No

Common Options

None.

Output

Eucalyptus returns a message stating that report was generated to the file you specified.

Example

```
eucadw-generate-report -s 2012-11-05 -e 2012-11-07 --size-unit=b
     --size-time-size-unit=b -t instance Report2.html -p mypassword
Exported data to Report2.html
```

Eucalyptus Configuration Properties

Eucalyptus exposes a number of properties that can be configured using the `euca-modify-property` command. This topic explains what types of properties Eucalyptus uses, and lists the most common configurable properties.

Eucalyptus Property Types

Eucalyptus uses two types of properties: ones that can be changed (as configuration options), and ones that cannot be changed (they are displayed as properties, but configured by modifying the `eucalyptus.conf` file on the CC).

Non-Changeable Properties

The following properties are 'discovered' by the CLC by asking the CC for the values of the properties. They are configured by setting them in eucalyptus.conf on a CC, and define the 'maximum values that a cluster can possibly support, based on the settings in eucalyptus.conf', and some static values (such as mode and usenetworktags).

- PROPERTY ecc-cluster-1.cluster.addressespernetwork 128
- PROPERTY ecc-cluster-1.cluster.maxnetworkindex 126
- PROPERTY ecc-cluster-1.cluster.maxnetworktag 511
- PROPERTY ecc-cluster-1.cluster.minnetworkindex 9
- PROPERTY ecc-cluster-1.cluster.minnetworktag 2
- PROPERTY ecc-cluster-1.cluster.networkmode MANAGED
- PROPERTY ecc-cluster-1.cluster.usenetworktags true

If you attempt to change these properties on the CLC, they will revert back to the values that are set on the CC. This information is discovered through an internal call (from CLC to CC) to describeNetworks().

Note that all of these properties are 'cluster.' properties, and they are all unsettable (as properties). The meaning of each follows:

Property	Description
cluster.addressespernetwork	Value set on the CC as VNET_ADDRSPERNET.
cluster.maxnetworkindex	Value calculated as the maximum index into a sec. group subnet (VNET_ADDRSPERNET - 2).
cluster.minnetworkindex	Value calculated as the minimum index into a sec. group subnet (0 is reserved, 1-8 are reserved as cluster def. GW IPs, so 9).
cluster.minnetworktag	Minimum network tag (VLAN, in MANAGED mode, just an index in MANAGED-NOVLAN mode) that the cluster will support (VLAN 0 and 1 is reserved, so 2).
cluster.maxnetworktag	Maximum network tag that the cluster will support (calculated as VNET_SUBNET/VNET_NETMASK size divided by VNET_ADDRSPERNET minus 1 (starts at 0)).
ecc-cluster-1.cluster.usenetworktags	Determines whether or not the system will be using network tags/indices at all (true in MANAGED* modes, false in SYSTEM/STATIC)

Configurable Properties

The following are the properties that can be set on the CLC, by the admin, as configuration options:

- PROPERTY cloud.network.global_max_network_index 4096
- PROPERTY cloud.network.global_max_network_tag 160
- PROPERTY cloud.network.global_min_network_index 2
- PROPERTY cloud.network.global_min_network_tag 30

These are properties that must be either identical or non-overlapping subsets of their equivalent cluster. properties. Their meanings are similar to the cluster level properties, but they can be constrained by setting them to a subset range of the range that the cluster supports.

For example, if an administrator wishes a cluster to only use VLANs 30 - 160 (the above case), then they would set these cloud.network.global. settings appropriately.

The above example shows that while the cluster settings permit the software to use VLANs 2 - 511, the administrator has configured the cloud to only use VLANs 30 - 160. In MANAGED-NOVLAN mode, there is no reason to change these parameters from defaults, which match the cluster configuration.

Note: Once a cloud is in use and has started operating based on these properties, it is not safe to change them at runtime.

Eucalyptus Properties

The following table contains a list of common Eucalyptus cloud properties.

Property	Description
authentication.credential_download_host_match	CIDR to match against for host address selection.
authentication.ldap_integration_configuration	LDAP integration configuration, in JSON.
authentication.websession_life_in_minutes	Web session lifetime in minutes.
autoscaling.activityexpiry	Expiry age for scaling activities. Older activities are deleted.
autoscaling.activityinitialbackoff	Initial backoff period for failing activities.
autoscaling.activitymaxbackoff	Maximum backoff period for failing activities.
autoscaling.activitytimeout	Timeout for a scaling activity.
autoscaling.maxlaunchincrement	Maximum instances to launch at one time.
autoscaling.maxregistrationretries	Maximum number of times to attempt load balancer registration for each instance.
autoscaling.pendinginstancetimeout	Timeout for a pending instance.
autoscaling.suspendedprocesses	Globally suspended scaling processes.
autoscaling.suspendedtasks	Suspended scaling tasks.
autoscaling.suspensionlaunchattemptsthreshold	Minimum launch attempts for administrative suspension of scaling activities for a group.
autoscaling.suspensiontimeout	Timeout for administrative suspension of scaling activities for a group.
autoscaling.untrackedinstancetimeout	Timeout for termination of untracked auto scaling instances.
autoscaling.zonefailurethreshold	Time after which an unavailable zone should be treated as failed.

Property	Description
bootstrap.async.future_listener_debug_limit_secs	Number of seconds a future listener can execute before a debug message is logged.
bootstrap.async.future_listener_error_limit_secs	Number of seconds a future listener can execute before an error message is logged.
bootstrap.async.future_listener_get_retries	Total number of seconds a future listener's executor waits to get().
bootstrap.async.future_listener_get_timeout	Number of seconds a future listener's executor waits to get() per call.
bootstrap.async.future_listener_info_limit_secs	Number of seconds a future listener can execute before an info message is logged.
bootstrap.hosts.state_initialize_timeout	Timeout for state initialization (in msec).
bootstrap.hosts.state_transfer_timeout	Timeout for state transfers (in msec).
bootstrap.notifications.batch_delay_seconds	Interval (in seconds) during which a notification will be delayed to allow for batching events for delivery.
bootstrap.notifications.digest	Send a system state digest periodically.
bootstrap.notifications.digest_frequency_hours	Period (in hours) with which a system state digest will be delivered.
bootstrap.notifications.digest_only_on_errors	If sending system state digests is set to true, then only send the digest when the system has failures to report.
bootstrap.notifications.email_from	From email address used for notification delivery.
bootstrap.notifications.email_from_name	From email name used for notification delivery.
bootstrap.notifications.email_subject_prefix	Email subject used for notification delivery.
bootstrap.notifications.email_to	Email address where notifications are to be delivered.
bootstrap.notifications.include_fault_stack	Period (in hours) with which a system state digest will be delivered.
bootstrap.notifications.email.email_smtp_host	SMTP host to use when sending email. If unset, the following values are tried: 1) the value of the 'mail.smtp.host' system property, 2) localhost, 3) mailhost.
bootstrap.notifications.email.email_smtp_port	SMTP port to use when sending email. Defaults to 25.
bootstrap.servicebus.context_timeout	Message context timeout (seconds).
bootstrap.servicebus.hup	Do a soft reset.
bootstrap.servicebus.max_outstanding_messages	Max queue length allowed per service stage.
bootstrap.servicebus.min_scheduler_core_size	Internal connector core pool size.
bootstrap.servicebus.workers_per_stage	Max queue length allowed per service stage.
bootstrap.timer.rate	Amount of time (in milliseconds) before a previously running instance which is not reported will be marked as terminated.
bootstrap.topology.coordinator_check_backoff_secs	Backoff between service state checks (in seconds).
bootstrap.topology.local_check_backoff_secs	Backoff between service state checks (in seconds).

Property	Description
bootstrap.tx.concurrent_update_retries	Maximum number of times a transaction may be retried before giving up.
bootstrap.webservices.async_internal_operations	Execute internal service operations from a separate thread pool (with respect to I/O).
bootstrap.webservices.async_operations	Execute service operations from a separate thread pool (with respect to I/O).
bootstrap.webservices.async_pipeline	Execute service specific pipeline handlers from a separate thread pool (with respect to I/O).
bootstrap.webservices.channel_connect_timeout	Channel connect timeout (ms).
bootstrap.webservices.channel_keep_alive	Socket keep alive.
bootstrap.webservices.channel_nodelay	Server socket TCP_NODELAY.
bootstrap.webservices.channel_reuse_address	Socket reuse address.
bootstrap.webservices.client_http_chunk_buffer_max	Server http chunk max.
bootstrap.webservices.client_idle_timeout_secs	Client idle timeout (secs).
bootstrap.webservices.client_internal_timeout_secs	Client idle timeout (secs).
bootstrap.webservices.client_pool_max_mem_per_conn	Server worker thread pool max.
bootstrap.webservices.client_pool_max_threads	Server worker thread pool max.
bootstrap.webservices.client_pool_timeout_millis	Client socket select timeout (ms).
bootstrap.webservices.client_pool_total_mem	Server worker thread pool max.
bootstrap.webservices.clock_skew_sec	A max clock skew value (in seconds) between client and server accepted when validating timestamps in Query/REST protocol.
bootstrap.webservices.cluster_connect_timeout_millis	Cluster connect timeout (ms).
bootstrap.webservices.default_aws_sns_uri_scheme	Default scheme for AWS_SNS_URL in eucarc.
bootstrap.webservices.default_ec2_uri_scheme	Default scheme for EC2_URL in eucarc.
bootstrap.webservices.default_euare_uri_scheme	Default scheme for EUARE_URL in eucarc.
bootstrap.webservices.default_eustore_url	Default EUSTORE_URL in eucarc.
bootstrap.webservices.default_https_enabled	Default scheme prefix in eucarc.
bootstrap.webservices.default_s3_uri_scheme	Default scheme for S3_URL in eucarc.
bootstrap.webservices.http_max_chunk_bytes	Maximum HTTP chunk size (bytes).
bootstrap.webservices.http_max_header_bytes	Maximum HTTP headers size (bytes).
bootstrap.webservices.http_max_initial_line_bytes	Maximum HTTP initial line size (bytes).
bootstrap.webservices.listener_address_match	CIDRs matching addresses to bind on (note: default interface is always bound regardless).
bootstrap.webservices.oob_internal_operations	Execute internal service operations out of band from the normal service bus.
bootstrap.webservices.pipeline_idle_timeout_seconds	Server socket idle timeout.
bootstrap.webservices.pipeline_read_timeout_seconds	Server socket read timeout.

Property	Description
bootstrap.webservices.pipeline_write_timeout_seconds	Server socket write timeout.
bootstrap.webservices.port	Port to bind (note: port 8773 is always bound regardless).
bootstrap.webservices.replay_skew_window_sec	Time interval duration (in seconds) during which duplicate signatures will be accepted to accomodate collisions for legitimate requests inherent in Query/REST signing protocol.
bootstrap.webservices.server_boss_pool_max_mem_per_conn	Server max selector memory per connection.
bootstrap.webservices.server_boss_pool_max_threads	Server selector thread pool max.
bootstrap.webservices.server_boss_pool_timeout_millis	Service socket select timeout (ms).
bootstrap.webservices.server_boss_pool_total_mem	Server worker thread pool max.
bootstrap.webservices.server_channel_nodelay	Server socket TCP_NODELAY.
bootstrap.webservices.server_channel_reuse_address	Server socket reuse address.
bootstrap.webservices.server_pool_max_mem_per_conn	Server max worker memory per connection.
bootstrap.webservices.server_pool_max_threads	Server worker thread pool max.
bootstrap.webservices.server_pool_timeout_millis	Service socket select timeout (ms).
bootstrap.webservices.server_pool_total_mem	Server max worker memory total.
bootstrap.webservices.statistics	Record and report service times.
bootstrap.webservices.use_dns_delegation	Use DNS delegation for eucarc.
bootstrap.webservices.use_instance_dns	Use DNS names for instances.
bootstrap.webservices.ssl.server_alias	Alias of the certificate entry in euca.p12 to use for SSL for webservices.
bootstrap.webservices.ssl.server_password	Password of the private key corresponding to the specified certificate for SSL for webservices.
bootstrap.webservices.ssl.server_ssl_ciphers	SSL ciphers for webservices.
bootstrap.webservices.unknown_parameter_handling	Allows unknown parameters to be ignored for all services or treated an error for all services. Valid values: • `default`: Use each services default handling (i.e., error with EC2, ignore unknown parameters for other services) • `ignore`: All services ignore unknown parameters • `error`: All services fail with an error for unknown parameters
cloud.addresses.dodynamicpublicaddresses	Public addresses are assigned to instances by the system as available.
cloud.addresses.maxkillorphans	Number of times an orphaned address is reported by a cluster before it is reclaimed by the system.
cloud.addresses.orphangrace	Time after the last recorded state change where an orphaned address will not be modified by the system (minutes).

Property	Description
cloud.addresses.systemreservedpublicaddresses	Public addresses are assigned to instances by the system only from a pool of reserved instances whose size is determined by this value.
cloud.cluster.disabledinterval	The time period between service state checks for a Cluster Controller which is DISABLED.
cloud.cluster.enabledinterval	The time period between service state checks for a Cluster Controller which is ENABLED.
cloud.cluster.notreadyinterval	The time period between service state checks for a Cluster Controller which is NOTREADY.
cloud.cluster.pendinginterval	The time period between service state checks for a Cluster Controller which is PENDING.
cloud.cluster.requestworkers	The number of concurrent requests which will be sent to a single Cluster Controller.
cloud.cluster.startupsyncretries	The number of times a request will be retried while bootstrapping a Cluster Controller.
cloud.db_check_poll_time	Poll time (ms) for db connection check.
cloud.db_check_threshold	Threshold (num connections or %) for db connection check.
cloud.euca_log_level	Log level for dynamic override.
cloud.identifier_canonicalizer	Name of the canonicalizer for resource identifiers.
cloud.images.cleanupperiod	The period between runs for clean up of deregistered images.
cloud.images.defaultkernelid	The default used for running images which do not have a kernel specified in either the manifest, at register time, or at run-instances time.
cloud.images.defaultramdiskid	The default used for running images which do not have a ramdisk specified in either the manifest, at register time, or at run-instances time.
cloud.images.defaultvisibility	The default value used to determine whether or not images are marked 'public' when first registered.
cloud.images.maximagesizegb	The maximum registerable image size in GB.
cloud.log_file_disk_check_poll_time	Poll time (ms) for log file disk check.
cloud.log_file_disk_check_threshold	Threshold (bytes or %) for log file disk check.
cloud.memory_check_poll_time	Poll time (ms) for memory check.
cloud.memory_check_ratio	Ratio (of post-garbage collected old-gen memory) for memory check.
cloud.monitor.default_poll_interval_mins	How often the reporting system requests information from the Cluster Controller.
cloud.monitor.history_size	The initial history size of metrics to be sent from the Cluster Controller to the Cloud Controller.

Property	Description
cloud.network.ec2_classic_additional_protocols_allowed	Comma delimited list of protocol numbers supported in in EDGE mode for security group rules beyond the EC2-Classic defaults (TCP,UDP,ICMP). Only *valid IANA protocol numbers* are accepted. Default: None
cloud.network.global_max_network_index	Default max network index.
cloud.network.global_max_network_tag	Default max vlan tag.
cloud.network.global_min_network_index	Default min network index.
cloud.network.global_min_network_tag	Default min vlan tag.
cloud.network.min_broadcast_interval	Minimum interval between broadcasts of network information (seconds).
cloud.network.network_configuration	Network configuration document.
cloud.network.network_index_pending_timeout	Minutes before a pending index allocation times out and is released.
cloud.network.network_tag_pending_timeout	Minutes before a pending tag allocation times out and is released.
cloud.perm_gen_memory_check_poll_time	Poll time (ms) for perm-gen memory check.
cloud.perm_gen_memory_check_ratio	Ratio (of used memory) for perm-gen memory check.
cloud.trigger_fault	Fault ID last used to trigger test.
cloud.vmstate.buried_time	Amount of time (in minutes) to retain unreported terminated instance data.
cloud.vmstate.ebs_root_device_name	Name for root block device mapping.
cloud.vmstate.ebs_volume_creation_timeout	Amount of time (in minutes) before a EBS volume backing the instance is created.
cloud.vmstate.instance_subdomain	Subdomain to use for instance DNS.
cloud.vmstate.instance_timeout	Amount of time (in minutes) before a previously running instance which is not reported will be marked as terminated.
cloud.vmstate.instance_touch_interval	Amount of time (in minutes) between updates for a running instance.
cloud.vmstate.mac_prefix	Prefix to use for instance MAC addresses.
cloud.vmstate.max_state_threads	Maximum number of threads the system will use to service blocking state changes.
cloud.vmstate.migration_refresh_time	Maximum amount of time (in seconds) that migration state will take to propagate state changes (e.g., to tags).
cloud.vmstate.network_metadata_refresh_time	Maximum amount of time (in seconds) that the network topology service takes to propagate state changes.
cloud.vmstate.shut_down_time	Amount of time (in minutes) before a VM which is not reported by a cluster will be marked as terminated.
cloud.vmstate.stopping_time	Amount of time (in minutes) before a stopping VM which is not reported by a cluster will be marked as terminated.

Property	Description
cloud.vmstate.terminated_time	Amount of time (in minutes) that a terminated VM will continue to be reported.
cloud.vmstate.tx_retries	Number of times to retry transactions in the face of potential concurrent update conflicts.
cloud.vmstate.user_data_max_size_kb	Max length (in KB) that the user data file can be for an instance (after base 64 decoding).
cloud.vmstate.vm_initial_report_timeout	Amount of time (in seconds) since completion of the creating run instance operation that the new instance is treated as `unreported if not...` reported.
cloud.vmstate.vm_metadata_instance_cache	Instance metadata cache configuration.
cloud.vmstate.vm_metadata_request_cache	Instance metadata instance resolution cache configuration.
cloud.vmstate.vm_metadata_user_data_cache	Instance metadata user data cache configuration.
cloud.vmstate.vm_state_settle_time	Amount of time (in seconds) to let instance state settle after a transition to either stopping or shutting down.
cloud.vmstate.volatile_state_interval_sec	Period (in seconds) between state updates for actively changing state.
cloud.vmstate.volatile_state_timeout_sec	Timeout (in seconds) before a requested instance terminate will be repeated.
cloud.vmtypes.default_type_name	Default type used when no instance type is specified for run instances.
cloudformation.region	The value of AWS::Region and value in CloudFormation ARNs for Region.
cloudwatch.disable_cloudwatch_service	Set this to true to stop cloud watch alarm evaluation and new alarm/metric data entry.
dns.dns_listener_address_match	Additional address patterns to listen on for DNS requests.
dns.enabled	Enable pluggable DNS resolvers. **Note**: This must be 'true' for any pluggable resolver to work. Also, each resolver may need to be separately enabled. See `euca-describe-properties dns`.
dns.instancedata.enabled	Enable the instance-data resolver. **Note**: `dns.enabled` must also be 'true'.
dns.ns.enabled	Enable the NS resolver. **Note**: `dns.enabled` must also be 'true'.
dns.recursive.enabled	Enable the recursive DNS resolver. **Note**: `dns.enabled` must also be 'true'.
dns.services.enabled	Enable the service topology resolver. **Note**: `dns.enabled` must also be 'true'.
dns.services.hostmapping	Comma separated list of listener address CIDRs to desired host address CIDRS for services.
dns.split_horizon.enabled	Enable the split-horizon DNS resolution for internal instance public DNS name queries.**Note**: `dns.enabled` must also be 'true'.

Property	Description
dns.spoof_regions.enabled	Enable the spoofing resolver which allows for AWS DNS name emulation for instances.**Note**: `dns.enabled` must also be 'true'.
dns.spoof_regions.region_name	Internal region name. If set, the region name to expect as the second label in the DNS name. For example, to treat your Eucalyptus install like a region named 'eucalyptus', set this value to `eucalyptus`. Then, e.g., autoscaling.eucalyptus.amazonaws.com will resolve to the service address when using this DNS server. The specified name creates a pseudo-region with DNS names like ec2.pseudo-region.amazonaws.com will resolve to Eucalyptus endpoints from inside of instances. Here `ec2` is any service name supported by Eucalyptus. Those that are not supported will continue to resolve through AWS's DNS.
dns.spoof_regions.spoof_aws_default_regions	Enable spoofing of the default AWS DNS names, e.g., ec2.amazonaws.com would resolve to the ENABLED Cloud Controller. Here `ec2` is any service name supported by Eucalyptus. Those that are not supported will continue to resolve through AWS's DNS.
dns.spoof_regions.spoof_aws_regions	Enable spoofing for the normal AWS regions, e.g., ec2.us-east-1.amazonaws.com would resolve to the ENABLED Cloud Controller. Here `ec2` is any service name supported by Eucalyptus. Those that are not supported will continue to resolve through AWS's DNS.
dns.tcp.timeout_seconds	Parameter controlling tcp handler timeout in seconds.
imaging.imaging_worker_availability_zones	Availability zones for imaging worker.
imaging.imaging_worker_emi	EMI containing imaging worker.
imaging.imaging_worker_enabled	Enabling imaging worker.
imaging.imaging_worker_healthcheck	Enabling imaging worker healthcheck.
imaging.imaging_worker_instance_type	Instance type for imaging worker.
imaging.imaging_worker_keyname	Keyname to use when debugging imaging worker.
imaging.imaging_worker_log_server	Address/IP of the server that collects logs from imaging workers.
imaging.imaging_worker_log_server_port	UDP port that log server is listening to.
imaging.imaging_worker_ntp_server	Address of the NTP server used by imaging worker.
imaging.import_task_expiration_hours	Expiration hours of import volume/instance tasks.
imaging.import_task_timeout_minutes	Expiration time in minutes of import tasks.
loadbalancing.loadbalancer_app_cookie_duration	Duration of app-controlled cookie to be kept in-memory (hours).
loadbalancing.loadbalancer_dns_subdomain	Loadbalancer dns subdomain.
loadbalancing.loadbalancer_emi	EMI containing haproxy and the controller.
loadbalancing.loadbalancer_instance_type	Instance type for loadbalancer instances.

Property	Description
loadbalancing.loadbalancer_num_vm	Number of VMs per loadbalancer zone.
loadbalancing.loadbalancer_restricted_ports	The ports restricted for use as a loadbalancer port. Format should be port(, port) or [port-port].
loadbalancing.loadbalancer_vm_keyname	Keyname to use when debugging loadbalancer VMs.
loadbalancing.loadbalancer_vm_ntp_server	The address of the NTP server used by loadbalancer VMs.
objectstorage.bucket_creation_wait_interval_seconds	Interval, in seconds, during which buckets in creating-state are valid. After this interval, the operation is assumed failed.
objectstorage.bucket_naming_restrictions	The S3 bucket naming restrictions to enforce. Values are 'dns-compliant' or 'extended'. Default is 'extended'. dns_compliant is non-US region S3 names, extended is for US-Standard Region naming. See *http://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html*.
objectstorage.cleanup_task_interval_seconds	Interval, in seconds, at which cleanup tasks are initiated for removing old/stale objects.
objectstorage.dogetputoncopyfail	Should provider client attempt a GET / PUT when backend does not support Copy operation.
objectstorage.failed_put_timeout_hrs	Number of hours to wait for object PUT operations to be allowed to complete before cleanup.
objectstorage.max_buckets_per_account	Maximum number of buckets per account.
objectstorage.max_total_reporting_capacity_gb	Total ObjectStorage storage capacity for Objects soley for reporting usage percentage. Not a size restriction. No enforcement of this value.
objectstorage.providerclient	Object Storage Provider client to use for backend.
objectstorage.queue_size	Channel buffer queue size for uploads.
objectstorage.s3provider.s3accesskey	Local Store S3 Access Key.
objectstorage.s3provider.s3endpoint	External S3 endpoint.
objectstorage.s3provider.s3secretkey	Local Store S3 Secret Key.
objectstorage.s3provider.s3usebackenddns	Use DNS virtual-hosted-style bucket names for communication to service backend.
objectstorage.s3provider.s3usehttps	Use HTTPS for communication to service backend.
<partition>.cluster.addressespernetwork	Number of total addresses per network (including unusable gateway addresses controlled by the system).
<partition>.cluster.maxnetworkindex	Maximum usable network index (0 < x < max_network_index).
<partition>.cluster.maxnetworktag	Maximum vlan tag to use (0 < min_vlan < x < 4096).
<partition>.cluster.minnetworkindex	Maximum usable network index (0 < min_network_index < x).
<partition>.cluster.minnetworktag	Minimum vlan tag to use (0 < x < max_vlan <= 4096).
<partition>.cluster.networkmode	Currently configured network mode.
<partition>.cluster.sourcehostname	Alternative address which is the source address for requests made by the component to the cloud controller.

Property	Description
<partition>.cluster.usenetworktags	Indicates whether vlans are in use or not.
<partition>.cluster.vnetnetmask	Netmask used by the cluster's virtual private networking.
<partition>.cluster.vnetsubnet	IP subnet used by the cluster's virtual private networking.
<partition>.cluster.vnettype	IP version used by the cluster's virtual private networking.
<partition>.storage.blockstoragemanager	EBS Block Storage Manager to use for backend.
<partition>.storage.chapuser	User ID for CHAP authentication.
<partition>.storage.dasdevice	Direct attached storage device location.
<partition>.storage.deletedvolexpiration	Expiration time for deleted volumes (hours).
<partition>.storage.majornumber	AOE Major Number.
<partition>.storage.maxconcurrentsnapshotuploads	Maximum number of snapshots that can be uploaded concurrently.
<partition>.storage.maxsnapshotpartsqueuesize	Maximum number of snapshot parts per snapshot that can be spooled on the disk.
<partition>.storage.maxsnaptransferretries	Maximum retry count for snapshot transfer.
<partition>.storage.maxtotalvolumesizeingb	Total disk space reserved for volumes.
<partition>.storage.maxvolumesizeingb	Max volume size.
<partition>.storage.minornumber	AOE Minor Number.
<partition>.storage.ncpaths	iSCSI paths for Node Controller.
<partition>.storage.resourceprefix	Prefix for resource name on SAN.
<partition>.storage.resourcesuffix	Suffix for resource name on SAN.
<partition>.storage.sanhost	Hostname for SAN device.
<partition>.storage.sanpassword	Password for SAN device.
<partition>.storage.sanuser	Username for SAN device.
<partition>.storage.scpaths	iSCSI paths for Storage Controller.
<partition>.storage.shouldtransfersnapshots	Should transfer snapshots.
<partition>.storage.snapshotpartsizeinmb	Snapshot part size in MB for snapshot transfers using multipart upload. Minimum part size is 5MB.
<partition>.storage.snapshotuploadtimeoutinhours	Snapshot upload wait time in hours after which the upload will be cancelled.
<partition>.storage.storageinterface	Storage network interface.
<partition>.storage.storeprefix	Prefix for ISCSI device.
<partition>.storage.tasktimeout	Timeout for SAN commands.
<partition>.storage.tid	Next Target ID for ISCSI device.
<partition>.storage.timeoutinmillis	Timeout value in milliseconds for storage operations.
<partition>.storage.volumesdir	Storage volumes directory.
<partition>.storage.zerofillvolumes	Should volumes be zero filled.

Property	Description
reporting.default_write_interval_mins	How often the reporting system requests information from the Cluster Controller.
reporting.default_size_time_size_unit	Default size-time size unit (GB-days, etc).
reporting.default_size_time_time_unit	Default size-time time unit (GB-days, etc).
reporting.default_size_unit	Default size unit.
reporting.default_time_unit	Default time unit.
reporting.default_write_interval_secs	How often the reporting system writes instance snapshots.
services.imaging.import_task_expiration_hours	Expiration hours of import volume/instance tasks. Default value is 168 hours.
services.imaging.import_task_timeout_minutes	Expiration time in minutes of import tasks. Default value is 180 minutes.
services.imaging.worker.availability_zones	Availability zones for imaging worker.
services.imaging.worker.configured	Configure imaging so a worker can be launched. Default value is true. If a service fails for some reason, there is a chance that setting it to false and back to true would solve issues.
services.imaging.worker.expiration_days	The days after which imaging work VMs expire. Default value is 180 days.
services.imaging.worker.healthcheck	Enables the imaging worker healthcheck. Default value is true.
services.imaging.worker.image	EMI containing imaging worker. The EMI is registered when the Load Balancer is configured.
services.imaging.worker.init_script	Bash script that will be executed before service configuration and start up.
services.imaging.worker.instance_type	Instance type for imaging worker. Default value is m1.small.
services.imaging.worker.keyname	Keyname to use when debugging imaging worker.
services.imaging.worker.log_server	Address/IP of the server that collects logs from the imaging workers.
services.imaging.worker.log_server_port	The UDP port that log server is listening to. Default value is 514.
services.imaging.worker.ntp_server	Address of the NTP server used by the imaging worker.
services.loadbalancing.dns_resolver_enabled	Enable the load balancing DNS resolver. **Note**: `dns.enabled` must also be 'true'.
storage.global_total_snapshot_size_limit_gb	Maximum total snapshot capacity (GB).
system.dns.nameserveraddress	Nameserver IP address.
system.dns.dnsdomain	Domain name to use for DNS.
system.dns.nameserver	Nameserver address.
system.dns.registrationid	Unique ID of this cloud installation.

Property	Description
system.exec.max_restricted_concurrent_ops	Maximum number of concurrent processes which match any of the patterns in system.exec.restricted_concurrent_ops.
system.exec.io_chunk_size	Size of IO chunks for streaming IO.
system.exec.restricted_concurrent_ops	Comma-separated list of commands which are restricted by system.exec.max_restricted_concurrent_ops.
tagging.max_tags_per_resource	The maximum number of tags per resource for each account.
tokens.disabledactions	Actions to disable.
tokens.enabledactions	Actions to enable (ignored if empty).
walrusbackend.blockdevice	DRBD block device.
walrusbackend.resource	DRBD resource name.
walrusbackend.storagedir	Path to buckets storage.
walrusbackend.storagemaxtotalcapacity	Total WalrusBackend storage capacity for Objects.
www.http_port	Listen to HTTP on this port.
www.httpproxyhost	Http Proxy Host.
www.httpproxyport	Http Proxy Port
www.https_ciphers	SSL ciphers for HTTPS listener.
www.https_port	Listen to HTTPS on this port.

Advanced Storage Configuration

This section covers advanced storage provider configuration options.

EMC VNX Advanced Configuration

This section contains advanced configuration, best practices, and troubleshooting tips for the EMC VNX SAN provider.

Configure EMC VNX Synchronous Snapshots

To configure synchronous snapshots for an EMC VNX SAN perform the tasks listed in this topic.

Setting the `<partition>.storage.enablesyncsnaps` property to `true` will cause snapshots to be set synchronously during a `euca-create-snapshot` operation. In this mode, the snapshot is created synchronously before the `euca-create-snapshot` command returns, while the copy and upload to Walrus still takes place asynchronously. This helps ensure that the `euca-create-snapshot` command returns quickly.

If the CLC loses the connection with the SC or if the connection times out (the default timeout is 60 seconds), the SC will detect the connection has been closed and will mark the snapshot as failed and will clean up. This detection occurs after the VNX snapshot has been created, but before it initiates the thread that performs the asynchronous migration and transfer of the snapshot LUN to Walrus. When using synchronous snapshot mode, if the CLC returns an error to the user on the `euca-create-snapshot` command then the snapshot will be marked as failed when listing snapshots using the `euca-describe-snapshots` command.

To configure synchronous snapshots for an EMC VNX SAN:

Set the `<partition>.storage.enablesyncsnaps` property to `true` :

```
euca-modify-property -p mypartition.storage.enablesyncsnaps=true
```

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

Best Practices for Multipathing with EMC VNX

This topic details some best practice suggestions for multipathing with EMC VNX.

Note: FEATURE PREVIEW: The multipathing feature is not yet complete, and may change or be removed from future releases. It is included in this release so that users can try it out and provide feedback.

The primary goals for multipathing with EMC VNX as a Eucalyptus EBS backend are to:

- Avoid single points of failure
- Maximize bandwidth for data access
- Isolate control traffic from data traffic to avoid performance problems

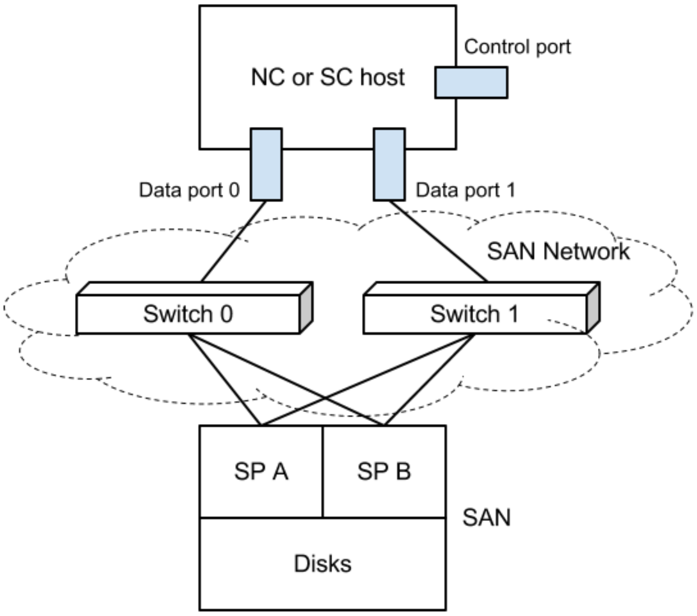
To achieve these goals, some best practice suggestions for multipathing are:

- Have at least two distinct networks for the data paths between NC/SC hosts and the SAN, so that there is no single point failure on the data path.
- Have separate network interfaces for NC and SC data and control traffic, to minimize the traffic interferences and maximize data bandwidth. Data access interfaces can use larger pipes, like 10GB Ethernet.
- Connect both SPs on the SAN to all of the data access networks.

The following diagram shows a typical multipathing configuration with EMC VNX. In this diagram, NC/SC hosts have 3 network interfaces: data port 0 and data port 1 for iSCSI data access, and the control port, which is used for control messages for Eucalyptus internal traffic. Each of the data port connects to a separate switch: switch 0 and switch 1. Each of the SAN storage processors, SP A and SP B, connects to both switches. In this diagram, we have 4 distinct iSCSI paths for each storage volume:

1. Data port 0 Switch 0 SP A
2. Data port 0 Switch 0 SP B
3. Data port 1 Switch 1 SP A
4. Data port 1 Switch 1 SP B

In this scenario, failure of any of the paths will not affect the storage access to the volumes:



Troubleshooting EMC VNX Multipathing

This topic is intended to help you troubleshoot EMC VNX multipathing.

Note: FEATURE PREVIEW: The multipathing feature is not yet complete, and may change or be removed from future releases. It is included in this release so that users can try it out and provide feedback.

The Eucalyptus EMC VNX Multipathing feature requires the following to function properly:

- Properly installed and configured Linux Device Mapper Multipathing software on both the Storage Controller and Node Controller hosts.
- Correctly configured iSCSI path system property and related STORAGE_INTERFACES parameters in the “eucalyptus.conf” configuration file for both SC and NC.

Prerequisites for Troubleshooting Typical

Before you start diagnosing the problems with multipathing, make sure you set the proper logging level on both SC and NC machines, so that you can get detailed failure logs. To do that:

- Set the “cloud.euca_log_level” system property to “DEBUG”

Multipathing Failures

- Uncomment the “LOGLEVEL=DEBUG” entry in the “eucalyptus.conf” file on the NC, and then restart the NC service

General Troubleshooting Techniques for Multipathing Failures

The following are general tips to help diagnose multipathing problems:

- Make sure you turn on the DEBUG log level for both SC and NC so that you can get detailed information from the logs.
- Eucalyptus calls some external Perl scripts to perform the actual iSCSI connect/disconnect operations. These scripts are:

  - /usr/share/eucalyptus/connect_iscsitarget.pl
  - /usr/share/eucalyptus/disconnect_iscsitarget.pl
  - /usr/share/eucalyptus/get_iscsitarget.pl

  The STDERR output of these scripts is logged; you can add debug code to print information to STDERR to see what happens during connection or disconnection operations.

- The `iscsiadm` open-iscsi initiator command line tool can help you get the current status of all the iSCSI connections in the system. For example:

  ```
  iscsiadm -m session -P 3
  ```

- Use the multipath command line tool to see multipathing status. For example:

  ```
  multipath -ll -v 3
  ```

Cannot attach volumes

This can occur for a number of reasons. To diagnose this, try some of the following:

- Make sure you can attach a volume without using multipathing.
- Check your SAN-related system properties to see if you have set the correct values.
- Use a single path for the NC; for example, set “PARTITION.storage.ncpaths” to something like “192.168.25.182”. If you specify an iface in your path, like “iface0:192.168.25.182”, also make sure you have “iface0” defined with “STORAGE_INTERFACES” in “eucalyptus.conf” configuration file on the NC.
- If you have no problem attaching a volume with a single path, the failure may be due to the incorrect state of the Linux device mapper multipathing tool. Check if the “multipathd” service is running on the NC hosts and if “/etc/multipath.conf” is installed and configured properly (for example, copy the example configuration provided by Eucalyptus). Remember to set “user_friendly_names” to “yes” in “/etc/multipath.conf”. You can try restarting “multipathd” and/or reloading “/etc/multipath.conf” if you changed it previously. Run “multipath -ll” on NC host and see if it returns reasonable output without any error.
- Check that the “PARTITION.storage.ncpaths” configuration file entries are correct. A typo can cause volume attach failures.
- Make sure that the networking configuration is correct for the NC hosts. If you set the paths without specific ifaces, check to see if you can connect to each IP in the path using default network interface; otherwise, check each path’s connectivity using a specific network interface.
- Check network connectivity with all of the configured paths.
- Check the “nc.log” log file for the string “connect_iscsitarget”. Examine the return results, especially the “stderr” output.

Not all paths are connected

Sometimes when you run “multipath -ll” on NC hosts after attaching a volume, you find that the multipath device does not have all of the paths connected. In this case, the problem could be due to one of the following:

- There is a mistake in the paths in one of the “PARTITION.storage.ncpaths” entries. If one of the paths specified in the system property is wrong, then it is possible that the specific path can not be connected. Make sure you have all the paths specified correctly.
- The missing paths are not valid networking paths, or have networking issues. For example, when you ignore the iface part of a path, are you sure that the destination of the path (the IP

	part of the path) can be connected via the default network interface? Or if you specified the iface, are you sure you defined the iface in the “eucalyptus.conf” configuration file, and that the destination can be connected with the specified network interface? • If the paths specified are all valid, but some of them do not have connectivity, try to ping each of the specified paths on the NC hosts to check for connectivity. If there are connectivity issues, contact your network administrator.
Snapshotting failed	The Eucalyptus Storage Controller needs to attach a volume on the machine it runs so it can upload to Walrus during an EC2 snapshot call. To help ensure maximum reliability for snapshotting, you should use multipathing for the SC host; this is configured with the “PARTITION.storage.scpaths” system property. When multipathing is enabled for the SC, if you see a snapshot failure, it may be caused by multipathing. Techniques for diagnosing SC multipathing failures is similar to those used for NC multipathing failures. In the case of SC multipathing failures, the logs are in “/var/log/eucalyptus/cloud-*.log”, not “nc.log”, since the iSCSI connect scripts are invoked by Java code.

NetApp Advanced Configuration

This section contains advanced configuration, best practices, and troubleshooting tips for the NetApp SAN provider.

NetApp Clustered Data ONTAP

A clustered ONTAP system consists of two or more individual NetApp storage controllers with attached disks. The basic building block is the HA pair, a term familiar from Data ONTAP 7G or 7-Mode environments.

An HA pair consists of two identical controllers; each controller actively provides data services and has redundant cabled paths to the other controller’s disk storage.

One of the key differentiators in a clustered ONTAP environment is that multiple HA pairs are combined together into a cluster to form a shared pool of physical resources available to applications. The shared pool appears as a single system image for management purposes. This means there is a single common point of management, whether through GUI or CLI tools, for the entire cluster. While the members of each HA pair must be the same controller type, the cluster can consist of heterogeneous HA pairs. Each NetApp storage controller with in a cluster is also referred to as a node.

The primary logical cluster component is the Virtual Storage Server, known as Vserver. Clustered ONTAP supports from one to hundreds of Vservers in a single cluster. A Vserver is configured for the client and host access protocols (such as iSCSI). Each Vserver contains at least one volume and at least one logical interface. The accessing hosts and clients connect to the Vserver using a logical interface (or LIF). LIFs present an IP address which will be used by iSCSI hosts. Each LIF has a home port on a NIC or HBA. LIFs are used to virtualize the NIC and HBA ports rather than mapping IP addresses or WWNs directly to the physical ports. Each Vserver requires its own dedicated set of LIFs, and up to 128 LIFs can be defined on any cluster node.

Each Vserver consists of different volumes and LIFs, providing secure compartmentalized access. Although the volumes and LIFs in each Vserver share the same physical resources (network ports and storage aggregates), a host or client can only access the data in a Vserver through a LIF defined in that Vserver. Administrative controls make sure that a delegated administrator with access to a Vserver can only see the logical resources assigned to that Vserver.

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

Eucalyptus integrates with NetApp Clustered ONTAP system by operating against a Vserver. SC must be configured to operate against Vserver contained in the NetApp Clustered ONTAP environment. SCs in other Eucalyptus clusters can be configured to use the same or different Vservers. SC and NC only interact with the configured Vserver and do not communicate with the Clustered ONTAP interfaces directly.

Configurable NetApp SAN Properties

This topic lists the NetApp SAN-specific properties you can configure using `euca-modify-property`, along with their valid values and Eucalyptus default values.

Note: The following configuration options are a subset of the Netapp SAN configuration parameters. Changing these default values may cause storage operations to fail. Please proceed at your own risk. For more information on NetApp configuration, please refer to the *NetApp Data ONTAP 7G documentation* and the *NetApp Data ONTAP 8G documentation* (these links require you to register and login).

7-Mode and Cluster Mode Properties

The following table lists properties that are applicable to both 7-mode and cluster mode:

Eucalyptus Property	Description	Valid Values
<region>.storage.enablespacereservation	LUN space reservation determines when space for the LUN is reserved or allocated from the flex volume. With reservations enabled the space is subtracted from the volume total when the LUN is created. If reservations are disabled, space is first taken out of the volume as writes to the LUN are performed.	Default value: true
<region>.storage.enablededup	Data deduplication removes duplicate blocks, storing only unique blocks of data in the flex volume, and it creates a small amount of additional metadata in the process. It is disabled by default. <region>.storage.enablecompression must be `false` before disabling deduplication.	Default value: false
<region>.storage.enablecompression	Data compression is a software-based solution that provides transparent data compression. It has the ability to run either as an inline process as data is written to disk or as a scheduled process. Compression is disabled by default. <region>.storage.enablededup must be true before enabling data compression. <region>.storage.enableinlinecompression must be false before disabling compression.	Default value: false
<region>.storage.enableinlinecompression	When data compression is configured for inline operation, data is compressed in memory before it is written to disk. It is disabled by default. <region>.storage.enablecompression must be true before enabling inline compression.	Default value: false

Eucalyptus Property	Description	Valid Values
<region>.storage.dedupschedule	Schedule string for the dedup and or compression operation on flex volumes. `<region>.storage.enablededup` must be `true` before configuring the schedule. If the schedule is not configured, NetApp applies a default schedule to the flex volume. In Cluster-Mode, either the schedule or the policy can be configured for the flex volume. Both cannot be configured together. The format of the schedule string is: "day_list@hour_list" or "hour_list@day_list" or "-" or "auto". day_list specifies which days of the week the sis operation should run. It is a comma-separated list of the first three letters of the day: sun, mon, tue, wed, thu, fri, sat. Day ranges such as mon-fri can also be used. hour_list specifies which hours of the day the sis operation should run on each scheduled day. hour_list is a comma-separated list of the integers from 0 to 23. Hour ranges such as 8-17 are allowed. Step values can be used in conjunction with ranges. If "-" is specified, no schedule is set. The "auto" schedule string means the sis operation will be triggered by the amount of new data written to the volume.	Default value: n/a
<region>.storage.lunostype	The operating system of the host accessing the LUN. This determines the layout of the data on the LUN, the geometry used to access that data, and property offsets for the LUN to ensure it is properly aligned with the upper layers of the file system	Default value: linux Valid values: `solaris`, `Solaris_efi`, `windows`, `windows_gpt`, `windows_2008`, `hpux`, `aix`, `linux`, `netware`, `xen`, or `hyper_v`
<region>.storage.initiatorostype	Operating system type of the hypervisor hosting the instances.	Default value: linux Valid values: `solaris`, `windows`, `hpux`, `aix`, `linux`, `netware`, `xen`, or `hyper_v`
<region>.storage.fractionalreserve	The percentage of space reserved for overwrites of reserved objects (LUNs or files) in a volume.	0-100; default is 0
<region>.storage.noatimeupdate	Prevents the update of inode access times when a file is read.	"on" (default) or "off"

Eucalyptus Property	Description	Valid Values
<region>.storage.tryfirst	Determines if the volume size is increased before deleting snapshots if `enableautosize` property is "true".	"volume_grow" (default) or "snap_delete"
<region>.storage.guarantee	Controls space reservation for flexible volumes. See the NetApp SDK documentation for more information.	"none", "file", or "volume" (default)
<region>.storage.enableautosize	Toggles the flex volume autosize feature.	"true" (default) or "false"
<region>.storage.volautosizemaxmultiplier	Flex volume's maximum size allowed, specified as a multiple of the original size	Integer >= 1; default is 3
<region>.storage.volautosizeincrementinmb	Flex volume's increment size in megabytes.	Integer >= 1; default is 256
<region>.storage.snappercent	Additional space reserved on the flex volume to store automatic and manual snapshots created outside of Eucalyptus. The amount of space to be reserved is specified as a percentage of the flex volume.	Integer >= 0; default is 0
<region>.storage.aggregate	Aggregates that can be used to create and manage volumes and snapshots. If a list of aggregates is configured, Eucalyptus will pick one based on <region>.storage.uselargestaggregate strategy. If no aggregate is provided Eucalyptus will query the NetApp SAN for available aggregates and choose one based <region>.storage.uselargestaggregate strategy.	Comma-separated string
<region>.storage.uselargestaggregate	If set to “true” Eucalyptus will pick the largest available aggregate from a list of aggregates. If set to “false” the smallest available aggregate will be chosen.	"true" (default) or "false"

7-Mode Properties

The following properties are specific to 7-mode:

Eucalyptus Property	Description	Valid Values
<region>.storage.convertucode	Setting this option to "on" forces conversion of all directories to UNICODE format when accessed from both NFS and CIFS.	"on" (default) or "off"
<region>.storage.createucode	Setting this option to "on" forces UNICODE format directories to be created by default from NFS and CIFS.	"on" (default) or "off"

Eucalyptus Property	Description	Valid Values
<region>.storage.snapschedweeks	Number of weekly snapshots to keep online.	Integer >= 0; default is 0
<region>.storage.snapscheddays	Number of daily snapshots to keep online.	Integer >= 0; default is 0
<region>.storage.snapschedhours	Number of hourly snapshots to keep online.	Integer >= 0; default is 0
<region>.storage.nosnap	Disable automatic snapshots. If set to “true”, snapshot scheduling properties <region>.storage.snapschedweeks and <region>.storage.snapscheddays and <region>.storage.snapschedhours are ignored, and the SC transmits the default value (0) in their place to the NetApp SAN.	"true" (default) or "false"

Cluster Mode Properties

The following properties are cluster mode specific:

Eucalyptus Property	Description	Valid Values
<region>.storage.snapshotpolicy	Snapshot retention policy determines how long the scheduled snapshots in the reserve are kept before being deleted automatically. This applies to automatic snapshots only.	String; default is "none"
<region>.storage.autosnapshots	Disable automatic snapshots. If set to “false” snapshot scheduling policy defined by <region>.storage.snapshotpolicy is igonred and SC transmits the default value (“none”) in its place to the NetApp SAN.	"true" (default) or "false"
<region>.storage.deduppolicy	Name of the sis policy to be attached to flex volumes in cluster-mode. <region>.storage.enablededup must be true before configuring the policy. Either the schedule or the policy can be configured for the flex volume. Both cannot be configured together.	Default value: n/a
<region>.storage.portset	Name of the portset to bind to an igroup in cluster-mode. Port sets are collections of iSCSI ports/LIFs. A port set can be used to restrict access to the LUN by making it visible only through target ports that are contained in the port set definition.	Default value: n/a

OSG Advanced Configuration

The following properties are for tuning the behavior of the Object Storage service and Gateways; the defaults are reasonable and changing is not necessary, but they are available for unexpected situations.

Property	Description
objectstorage.bucket_creation_wait_interval_seconds	The interval, in seconds, during which buckets in a 'creating' state are valid. After this interval, the operation is assumed failed. Valid values: integer > 0 Default: 60
objectstorage.bucket_naming_restrictions	The S3 bucket naming restrictions to enforce. Use dns_compliant for non-US region S3 names. Use extended for US-Standard Region naming. For more information, see *Bucket Restrictions and Limitations* in the Amazon S3 documentation. Valid values: dns-compliant | extended Default: extended
objectstorage.cleanuptaskintervalseconds	The interval, in seconds, at which background cleanup tasks are run. The background cleanup tasks purge the backend of overwritten objects and clean object history. Valid values: integer > 0 Default: 60
objectstorage.dogetcopyputonfail	When this property is enabled (true), the OSG attempts to perform a manual copy (performing a GET operation on the source, followed by a PUT operation on the destination) whenever the copy operation fails against the upstream provider. Because manual copies can be slow and memory-intensive, this capability is disabled (false) by default. Valid values: true | false Default: false
objectstorage.failedputtimeouthours	The time, in hours, after which an uncommitted object upload is considered to be failed. This allows cleansing of metadata for objects that were pending upload when an OSG fails or is stopped in the middle of a user operation. This should be kept at least as long as the longest reasonable time to upload a single large object in order to prevent unintentional cleanup of uploads in-progress. The S3 maximum single upload size is 5GB. Valid values: integer > 0 Default: 168

Property	Description
`objectstorage.max_buckets_per_account`	Maximum number of buckets per account. For more information, see *Bucket Restrictions and Limitations* in the Amazon S3 documentation. Valid values: integer > 0 Default: `100` (the AWS limit)
`objectstorage.max_total_reporting_capacity_gb`	Total object storage capacity for objects, used solely for reporting usage percentage. Not a size restriction. No enforcement of this value. Valid values: integer > 0 Default: `2147483647` (maximum value of an integer)
`objectstorage.queue_size`	The size, in chunks, of the internal buffers that queue data for transfer to the backend on a per-request basis. A larger value will allow more buffering in the OSG when the client is uploading quickly, but the backend bandwidth is lower and cannot consume data fast enough. Too large a value may result in out-of-memory (OOM) errors if the JVM does not have sufficient heap space to handle the concurrent requests * queue_size. Valid values: integer > 0 Default: `100`
`objectstorage.s3provider.s3usebackenddns`	Use DNS virtual-hosted-style bucket names for communication to service backend. Valid values: `true` | `false` Default: `false`
`objectstorage.s3provider.s3usehttps`	Whether or not to use HTTPS for the connections to the backend provider. If you configure this, be sure you can use the backend properly with HTTPS (certs, etc.) or the OSG will fail to connect. For RiakCS, you must configure certificates and identities to support HTTPS; it is not enabled in a default RiakCS installation. Valid values: `true` | `false` Default: `false`

Index