Eucalyptus 4.2.2 Administration Guide

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 3221 (2016-07-14 22:01:24)

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, in Managed mode (only), Eucalyptus uses VLANs at Layer 2 in the networking stack to ensure isolation.
		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_MODE`	The networking mode in which to run. The same mode must be specified on all CCs and NCs in your cloud. Valid values: `EDGE`, `MANAGED`, `MANAGED-NOVLAN`,	All
`VNET_PRIVINTERFACE`	The name of the network interface that is on the same network as the NCs. In Managed and Managed (No VLAN) modes this must be a bridge for instances in different clusters but in the same security group to be able to reach one another with their private addresses. Default: `eth0`	Edge (on NC) Managed
`VNET_PUBINTERFACE`	**On a CC**, this is the name of the network interface that is connected to the "public" network. **On an NC**, this is the name of the network interface that is connected to the same network as the CC. Depending on the hypervisor's configuration this may be a bridge or a physical interface that is attached to the bridge. Default: `eth0`	Edge (on NC) Managed Managed (No VLAN)
`VNET_SUBNET, VNET_NETMASK`	These options control the internal private network used by instances within Eucalyptus. Eucalyptus assigns a distinct subnet of private IP addresses to each security group. This setting dictates how many addresses each of these subnets should contain. Specify a power of 2 between 16 and 2048. This is directly related, though not equal, to the number of instances that can reside in each security group. Eucalyptus reserves eleven addresses per security group.	Managed, Managed (No VLAN)

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
```

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.

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:

- `walrusbackend.storagemaxbucketsizeinmb`: max bucket size enforced by Walrus; should be larger than a Windows image
- `walrusbackend.storagemaxtotalsnapshotsizeingb`: if a Windows image is a type of EBS-backed EMI, this should be large enough to store all EBS backed images

- `ZONE.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 `walrusbackend.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 `walrusbackend.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
{}
```

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 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 Regions

This section provides information about regions and identity federation.

Regions Overview

Eucalytpus provides support for the notion of federation of identity.

Federation of identity information means that a Cloud Administrator can create a federation of (otherwise independent) Eucalyptus "clouds" where a Cloud User, using the same credentials as always, can use any of these federated Eucalyptus cloud regions. For the parts of Identify Access Management (IAM) and Security Token Service (STS) that Eucalyptus implements, the experience exposed to the Cloud User is the same as that seen by an AWS user working across AWS regions.

A user can interact with any region using the same credentials, subjected to the same policies, and having uniformly accessible and structured principals (Accounts, Users, Groups, Roles, etc.). The globality also includes the STS service functionality, the temporary credentials produced by the STS service also work globally.

Notably, this feature is restricted to IAM/STS and does not include other services which have pseudo-global characteristics, such as global bucket name space for S3. The following are general principles associated with regions:

- A region needs to be Registered as a federated region
- Registered regions should be discoverable via the EC2 DescribeRegions response
- A cloud user's credentials should be accepted by any federated cloud
- There is a global IAM service (identities and policies are global for all registered regions)

Region Configuration File Format

This section describes the necessary configuration properties that need to be addressed.

For federation to be successfully configured, each cloud (i.e. region) that will be part of the federated cloud needs to have the following properties set (at a minimum):

Property Name	Description
`region.region_name`	This cloud property identifies the local region. This is required and should be valid for use in a DNS name.
`region.region_configuration`	This property is a JSON document that will be the same for all federated regions.

Elements

Below are the elements that should be contained in a region configuration file. The elements are listed in a general order as they should appear in the configuration file. However, order does not matter, for example, the `Name` element can come after the `CertificateFingerprintDigest` element.

- Regions
- Name
- CertificateFingerprintDigest
- CertificateFingerprint
- IdentifierPartitions
- Services
- Type

- Endpoints

Regions

The `Regions` element is required and is the main element for the configuration file. It can include multiple elements (see the subsequent sections in this section). The `Regions` element contains an array of individual regions. Each individual region is a JSON block enclosed in braces `{ }`.

```
"Regions":[{...},{...},{...}]
```

Name

The `Name` element is required and identifies the local region. This value should be unique across all regions and should match the cloud property `region.region_name` on the local region. The value of the `Name` element should follow the *label* format described in the 'Conventions' section mentioned in *RFC 1035 - Domain Names - Implementation and Specification.*. The conventions are as follows:

- `<label> ::= <letter> [ [ <ldh-str> ] <let-dig> ]`
- The labels must follow the rules for ARPANET host names. Per *RFC 1123*, Eucalyptus only supports lowercase characters, whereas DNS labels are *not* case-sensitive.

The `Name` element should be assigned to each region in the Regions array:

```
"Name":"region-1"
```

CertificateFingerprintDigest

The `CertificateFingerprintDigest` element is required and references the OpenSSL signature algorithm used for signing the Eucalyptus certificate *(/var/lib/eucalyptus/keys/cloud-cert.pem)*, with the default being SHA-256. For more information, go to *Oracle Java Documentation*. Values supported by Eucalyptus 4.2 are as follows:

- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

The `CertificateFingerprintDigest` element should be assigned to each region in the Regions array:

```
"CertificateFingerprintDigest": "SHA-256"
```

CertificateFingerprint

The `CertificateFingerprintDigest` element is required and references the OpenSSL fingerprint of the signed Eucalyptus certificate *(/var/lib/eucalyptus/keys/cloud-cert.pem)*. The value of `CertificateFingerprintDigest` should be used to help calculate the fingerprint. The default is SHA-256. The supported values can be found under the `CertificateFingerprintDigest` element. The `CertificateFingerprint` element should be assigned to each region in the Regions array.

```
"CertificateFingerprint":
"EC:E7:3D:DF:97:43:00:9E:FC:F0:2C:6D:98:D2:82:EB:AA:04:75:10:E7:C2:F2:6F:31:F1:F1:CA:A1:61:DE:41"
```

Currently there are two ways to calculate a fingerprint from the certificate using OpenSSL:

1. When the Eucalyptus cloud certificate is available.

   ```
   openssl x509 -inform PEM -in /var/lib/eucalyptus/keys/cloud-cert.pem -noout
   -fingerprint -sha256
   ```

2. Use the *-connect* option:

   ```
   openssl s_client -showcerts -connect 10.111.5.32:8773 < /dev/null 2>/dev/null
    | openssl x509 -noout -fingerprint -sha256
   ```

The `CertificateFingerprint` is mainly for WS-Security, it is often the same as the HTTPS certificate, but might not be (in which case, the second option for configuration will not work, i.e. "openssl s_client"). For more information about WS-Security, go to *WS-Security Wikipedia page*.

IdentifierPartitions

The `IdentifierPartitions` element is required and must include at least one value in the 1-999 range. This element is unique for each region. The `IdentifierPartitions` element should be assigned to each region in the Regions array:

```
"IdentifierPartitions": [1]
```

Services

The `Services` element is required and is an array that contains two type elements. Each type contains a `Type` and `Endpoints` element. These are the services that are associated with each region needed for federation. The `Services` element should be assigned to each region in the Regions array:

```
"Services": [
    {
        "Type": "[service type]",
        "Endpoints": [
            "[http/https service endpoint]..."
             ]
    },
    {
        "Type": "[service type]",
        "Endpoints": [
            "[http/https service endpoint]..."
            ]
    }
]
```

Type

The `Type` element is required and defines a service type under the `Services` element. Currently, the only types supported are *identity* and *compute*. If there are more than one *identity* and/or *compute* service types defined, the first one will be used. The `Type` element should be assigned to each service type in the Services array:

```
{
    "Type": "[identity|compute]",
    ....
}
```

Endpoints

The `Endpoints` element is required and is an array that defines the endpoint for each service type under the `Services` element. The endpoint format can have *either* of the following formats:

- IP format

  ```
  [http|https]://<IPv4 Address>:8773/services/[Identity|Compute]/
  ```

- DNS format

  ```
  [http/https]://[identity|compute]<Eucalyptus DNS subdomain>:8773/
  ```

Even though the IP address can be used for the endpoint, it is **highly recommended** to use the Eucalyptus DNS name of the service endpoint. If there are more than one `Endpoints` defined, the first one will be used. The `Endpoints` element should be assigned to each service type in the Services array.

```
{
    .....
    "Endpoints": [
```

```
        "[http/https DNS name endpoint or IP Address endpoint]"
    ]
}
```

Examples

In this example, there will be two clouds used (10.111.5.32 and 10.111.1.1). Before setting up federation, the clouds must meet the following requirements:

- Eucalyptus 4.2 installed
- Eucalyptus DNS enabled

Register a Region - Script-assisted

This procedure allows you to run a script to register a region:

1. Grab a copy of `/var/lib/eucalyptus/keys/cloud-cert.pem` from both Cloud Controllers.
2. Clone the following repository: *https://github.com/hspencer77/region-configuration-tool*.
3. Run the `region-config-tool.py` script, passing the cloud certificates for cloud (i.e. region). For example:

```
# ./region-config-tool.py
region_name=region-1,cloud_cert=at-long-last-asap-region.pem,domain_name=h-33.autoqa.qa1.eucalyptus-systems.com

region_name=region-2,cloud_cert=long-live-asap-region.pem,domain_name=g-22-07.autoqa.qa1.eucalyptus-systems.com
 -f test-region-config.json
# cat test-region-config.json
{
 "Regions": [
 {
 "CertificateFingerprint":
"ED:8F:9A:92:45:4D:37:F3:54:E4:2E:E7:26:28:EE:04:A1:DF:AD:82:87:60:A6:C3:4A:15:CB:D7:E9:F2:99:13",

 "CertificateFingerprintDigest": "SHA-256",
 "IdentifierPartitions": [
 1
 ],
 "Name": "region-1",
 "Services": [
 {
 "Endpoints": [
 "http://identity.h-33.autoqa.qa1.eucalyptus-systems.com:8773/"
 ],
 "Type": "identity"
 },
 {
 "Endpoints": [
 "http://compute.h-33.autoqa.qa1.eucalyptus-systems.com:8773/"
 ],
 "Type": "compute"
 }
 ]
 },
 {
 "CertificateFingerprint":
"3A:69:0F:B3:A5:03:92:50:39:F2:C6:EB:E5:77:94:36:F9:36:12:E2:01:CA:AB:75:B2:6E:71:9B:D0:5E:61:94",

 "CertificateFingerprintDigest": "SHA-256",
 "IdentifierPartitions": [
 2
 ],
 "Name": "region-2",
```

```
 "Services": [
 {
 "Endpoints": [
 "http://identity.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/"
 ],
 "Type": "identity"
 },
 {
 "Endpoints": [
 "http://compute.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/"
 ],
 "Type": "compute"
 }
 ]
 }
 ]
}
```

4. Change the region.region_name cloud property on both clouds:

```
[root@h-32 ~]# euca-modify-property -p region.region_name=region-1
PROPERTY region.region_name region-1 was {}

[root@b-01 ~]# euca-modify-property -p region.region_name=region-2
PROPERTY region.region_name region-2 was {}
```

5. Set `region.region_ssl_verify_hostnames` to *true* on both clouds and properly signed certificates should be added to the User Facing Service (UFS).
6. Add the region JSON file to both clouds by modifying the `region.region_configuration` cloud property.

Register a Region - Manual

This procedure allows you to manually register a region:

1. Obtain the certificate fingerprint for both clouds, for example:

```
## region-1
$ openssl s_client -showcerts -connect 10.111.5.32:8773 < /dev/null 2>/dev/null
 | openssl x509 -noout -fingerprint -sha256
SHA256
Fingerprint=53:AE:4C:2F:D4:2D:AB:41:B9:F9:0B:B0:3E:DE:5D:94:3B:81:FC:FB:CC:58:3D:42:71:13:01:94:97:23:23:DD

## region-2
$ openssl s_client -showcerts -connect 10.111.1.1:8773 < /dev/null 2>/dev/null
 | openssl x509 -noout -fingerprint -sha256
SHA256
Fingerprint=07:52:F3:50:07:FB:C3:B7:28:AA:ED:D4:19:17:D4:05:E8:92:DE:8A:85:18:2E:6C:11:A9:84:56:D8:A
```

2. Create region JSON configuration file, for example:

```
{
    "Regions": [
        {
            "Name": "region-1",
            "CertificateFingerprintDigest": "SHA-256",
            "CertificateFingerprint":
"53:AE:4C:2F:D4:2D:AB:41:B9:F9:0B:B0:3E:DE:5D:94:3B:81:FC:FB:CC:58:3D:42:71:13:01:94:97:23:23:DD",

            "IdentifierPartitions": [
                1
            ],
            "Services": [
                {
                    "Type": "identity",
```

```
                    "Endpoints": [

"http://identity.h-33.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                },
                {
                    "Type": "compute",
                    "Endpoints": [

"http://compute.h-33.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                }
            ]
        },
        {
            "Name": "region-2",
            "CertificateFingerprintDigest": "SHA-256",
            "CertificateFingerprint":
"07:52:F3:50:07:FB:C3:B7:28:AA:ED:D4:19:17:D4:05:E8:92:DE:8A:85:18:2E:6C:11:A9:84:56:D8:A3:82:03",

            "IdentifierPartitions": [
                2
            ],
            "Services": [
                {
                    "Type": "identity",
                    "Endpoints": [

"http://identity.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                },
                {
                    "Type": "compute",
                    "Endpoints": [

"http://compute.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                }
            ]
        }
    ]
}
```

3. Change the `region.region_name` cloud property on both clouds:

```
[root@h-32 ~]# euca-modify-property -p region.region_name=region-1
PROPERTY region.region_name region-1 was {}

[root@b-01 ~]# euca-modify-property -p region.region_name=region-2
PROPERTY region.region_name region-2 was {}
```

4. Set `region.region_ssl_verify_hostnames` to true, and properly signed certificates should be added to the User Facing Service (UFS).
5. Add the region JSON file to both clouds by modifying the `region.region_configuration` cloud property. Region section for both clouds look like the following after all the changes:

```
### region 1
[root@h-32 ~]# euca-describe-properties region.
PROPERTY region.region_configuration {
    "Regions": [
        {
            "Name": "region-1",
            "CertificateFingerprintDigest": "SHA-256",
            "CertificateFingerprint":
```

```
"53:AE:4C:2F:D4:2D:AB:41:B9:F9:0B:B0:3E:DE:5D:94:3B:81:FC:FB:CC:58:3D:42:71:13:01:94:97:23:23:DD",

            "IdentifierPartitions": [
                1
            ],
            "Services": [
                {
                    "Type": "identity",
                    "Endpoints": [

"http://identity.h-33.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                },
                {
                    "Type": "compute",
                    "Endpoints": [

"http://compute.h-33.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                }
            ]
        },
        {
            "Name": "region-2",
            "CertificateFingerprintDigest": "SHA-256",
            "CertificateFingerprint":
"07:52:F3:50:07:FB:C3:B7:28:AA:ED:D4:19:17:D4:05:E8:92:DE:8A:85:18:2E:6C:11:A9:84:56:D8:A3:82:03",

            "IdentifierPartitions": [
                2
            ],
            "Services": [
                {
                    "Type": "identity",
                    "Endpoints": [

"http://identity.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                },
                {
                    "Type": "compute",
                    "Endpoints": [

"http://compute.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                }
            ]
        }
    ]
}
PROPERTY region.region_enable_ssl true
PROPERTY region.region_name region-1
PROPERTY region.region_ssl_ciphers
RSA:DSS:ECDSA:TLS_EMPTY_RENEGOTIATION_INFO_SCSV:!NULL:!EXPORT:!EXPORT1024:!MD5:!DES
PROPERTY region.region_ssl_default_cas  true
PROPERTY region.region_ssl_protocols TLSv1.2
PROPERTY region.region_ssl_verify_hostnames true

## region 2
[root@b-01 ~]# euca-describe-properties region.
PROPERTY region.region_configuration {
    "Regions": [
        {
```

```
            "Name": "region-1",
            "CertificateFingerprintDigest": "SHA-256",
            "CertificateFingerprint":
"53:AE:4C:2F:D4:2D:AB:41:B9:F9:0B:B0:3E:DE:5D:94:3B:81:FC:FB:CC:58:3D:42:71:13:01:94:97:23:23:DD",

            "IdentifierPartitions": [
                1
            ],
            "Services": [
                {
                    "Type": "identity",
                    "Endpoints": [

"http://identity.h-33.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                },
                {
                    "Type": "compute",
                    "Endpoints": [

"http://compute.h-33.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                }
            ]
        },
        {
            "Name": "region-2",
            "CertificateFingerprintDigest": "SHA-256",
            "CertificateFingerprint":
"07:52:F3:50:07:FB:C3:B7:28:AA:ED:D4:19:17:D4:05:E8:92:DE:8A:85:18:2E:6C:11:A9:84:56:D8:A3:82:03",

            "IdentifierPartitions": [
                2
            ],
            "Services": [
                {
                    "Type": "identity",
                    "Endpoints": [

"http://identity.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                },
                {
                    "Type": "compute",
                    "Endpoints": [

"http://compute.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/"
                    ]
                }
            ]
        }
    ]
}
PROPERTY region.region_enable_ssl true
PROPERTY region.region_name region-2
PROPERTY region.region_ssl_ciphers
RSA:DSS:ECDSA:TLS_EMPTY_RENEGOTIATION_INFO_SCSV:!NULL:!EXPORT:!EXPORT1024:!MD5:!DES
PROPERTY region.region_ssl_default_cas true
PROPERTY region.region_ssl_protocols TLSv1.2
PROPERTY region.region_ssl_verify_hostnames true
```

Describe Regions

Using cloud administrator (i.e. eucalyptus/admin user) credentials on each cloud, use `execute DescribeRegions`. This is a sanity check to make sure the configuration is correct on both clouds.

Note: Reminder: the cloud administrator for each cloud can only see the resources for that cloud. The eucalyptus account is one of the system accounts, therefore it is not synced across all clouds.

```
# euca-describe-regions
REGION region-1 http://compute.h-33.autoqa.qa1.eucalyptus-systems.com:8773/
REGION region-2 http://compute.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/
```

Create a Non-system Account

After federation has been configuration correctly, create a non-system account on either cloud using the eucalyptus/admin user. In the example below, the non-system account *test1* will be created. The credentials from the test1/admin user will be downloaded and sourced. The user will run `DescribeAvailabilityZones` against both clouds to confirm federation is working as expected.

1. On region-2 cloud, using eucalyptus/admin user - create IAM account 'test1'

```
[root@b-01 ~]# euare-accountcreate -a test1
test1 002093902049

[root@b-01 ~]# euare-accountlist
eucalyptus 000163314767
(eucalyptus)objectstorage 000107497415
(eucalyptus)blockstorage 000831185453
(eucalyptus)loadbalancing 000744507680
(eucalyptus)aws-exec-read 000890823690
test1 002093902049
(eucalyptus)cloudformation 000993524712
(eucalyptus)database 000630877528
(eucalyptus)imaging 000789831484
```

2. Download credentials for test1/admin user:

```
# euca-get-credentials -a test1 -u admin test1-admin.zip
# unzip test1-admin.zip
Archive:  test1-admin.zip
To setup the environment run: source /path/to/eucarc
  inflating: eucarc
  inflating: iamrc
  inflating: cloud-cert.pem
  inflating: jssecacerts
  inflating: euca2-admin-af5f63d3-pk.pem
  inflating: euca2-admin-af5f63d3-cert.pem
```

3. Source test1/admin eucarc, then execute `DescribeAvailibilityZones` against each region:

```
# source eucarc
# euca-describe-regions
REGION region-1 http://compute.h-33.autoqa.qa1.eucalyptus-systems.com:8773/
REGION region-2 http://compute.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/
# euca-describe-availability-zones -U
http://compute.h-33.autoqa.qa1.eucalyptus-systems.com:8773/
AVAILABILITYZONE region1-az-one available
# euca-describe-availability-zones -U
http://compute.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/
AVAILABILITYZONE region2-az-one available
```

Federation Differences Between AWS and Eucalyptus

This section outlines the differences between AWS and Eucalyptus with respect to federation in the following platforms:

- Euca2ools vs. AWS EC2 API Tools
- Eucalyptus OSG vs. AWS S3
- Eucalyptus Resource-Level vs. AWS Resource-Level Permissions
- Global Cloud Administration (Local vs. Federated)

Euca2ools vs. AWS EC2 API Tools

Euca2ools uses the `--region` option to read information from a configuration file. For a user to be able to access resources from different regions using Euca2ools, the `-U URL`, `--url URL` option has to be used. This behavior is different when compared to AWS API tools. With the AWS API tools, the `--region` option is used to access resources in different regions. Examples are as follows:

Euca2ools

```
# euca-describe-regions
REGION region-1 http://compute.h-33.autoqa.qa1.eucalyptus-systems.com:8773/
REGION region-2 http://compute.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/
(Using --region to access resources from different regions; Notice that the
returning value is the same)
# euca-describe-availability-zones --region region-1
AVAILABILITYZONE region2-az-one available
# euca-describe-availability-zones --region region-2
AVAILABILITYZONE region2-az-one available
(Using -U URL, --url URL to access resources from different regions; Notice the
 difference in outputs)
# euca-describe-availability-zones -U
http://compute.h-33.autoqa.qa1.eucalyptus-systems.com:8773/
AVAILABILITYZONE region1-az-one available
# euca-describe-availability-zones -U
http://compute.g-22-07.autoqa.qa1.eucalyptus-systems.com:8773/
AVAILABILITYZONE region2-az-one available
```

AWS EC2 API Tools

```
$ ec2-describe-regions
REGION eu-central-1 ec2.eu-central-1.amazonaws.com
REGION sa-east-1 ec2.sa-east-1.amazonaws.com
REGION ap-northeast-1 ec2.ap-northeast-1.amazonaws.com
REGION eu-west-1 ec2.eu-west-1.amazonaws.com
REGION us-east-1 ec2.us-east-1.amazonaws.com
REGION us-west-1 ec2.us-west-1.amazonaws.com
REGION us-west-2 ec2.us-west-2.amazonaws.com
REGION ap-southeast-2 ec2.ap-southeast-2.amazonaws.com
REGION ap-southeast-1 ec2.ap-southeast-1.amazonaws.com
$ ec2-describe-availability-zones --region us-east-1
AVAILABILITYZONE us-east-1a available us-east-1
AVAILABILITYZONE us-east-1b available us-east-1
AVAILABILITYZONE us-east-1c available us-east-1
AVAILABILITYZONE us-east-1d available us-east-1
AVAILABILITYZONE us-east-1e available us-east-1
$ ec2-describe-availability-zones --region us-west-1
AVAILABILITYZONE us-west-1a available us-west-1
AVAILABILITYZONE us-west-1c available us-west-1
$ ec2-describe-availability-zones --region us-west-2
AVAILABILITYZONE us-west-2a available us-west-2
AVAILABILITYZONE us-west-2b available us-west-2
AVAILABILITYZONE us-west-2c available us-west-2
```

Eucalyptus OSG vs. AWS S3

Eucalyptus OSG for each region is a separate entity (i.e. if you want to have the same bucket across all regions, you need to create that bucket across each region). With AWS S3, once you create a bucket in one region, it is replicated to all regions. This is the same for objects as well.

ARN Resources

This behavior is extremely important when dealing with IAM access policies regarding S3 (OSG) resources. When defining a resource using an ARN:

```
arn:partition:service:region:namespace:relative-id
```

Per *AWS S3 documentation*, when specifying a resource in a policy, you don't specify region and namespace in the ARN. The S3 (OSG) ARN resource will look like the following:

```
arn:aws:s3:::bucket_name
arn:aws:s3:::bucket_name/key_name
```

If a user tries to specify a region and/or namespace in an ARN associated with an S3 (OSG) resource on Eucalyptus, the following error will be displayed:

```
error (MalformedPolicyDocument): Error in uploaded policy:
net.sf.json.JSONException: 'arn:aws:s3:region-1::*' is not a valid ARN
```

The only valid ARNs for S3 (OSG) resources on Eucalyptus are as follows:

```
"Resource": "arn:aws:s3:::*"
"Resource": "arn:aws:s3:::*/*
```

Eucalyptus vs. AWS Resource-Level Permissions

There are some services that AWS supports resource-level permissions to be used in AWS IAM access policies. A list of them can be found in the *AWS IAM Guide - AWS Services that work with IAM*. Two services that do not support resource-level permissions on AWS are the following:

- *Auto Scaling*
- *CloudWatch*

On Eucalyptus, these services **do support** resource-level permissions, as well as the implemented services that support resource-level permissions in AWS. Below are example IAM policies for each service:

Auto Scaling

Eucalyptus IAM policy to allow all Autoscaling actions against any AutoScaling group under a given account:

```
{
 "Version":"2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "autoscaling:*",
 "Resource": "arn:aws:autoscaling:::autoScalingGroup:*"
 }
 ]
}
```

CloudWatch

Eucalyptus IAM policy to allow all CloudWatch actions against any CloudWatch alarm under a given account:

```
{
 "Version":"2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "cloudwatch:*",
```

```
 "Resource": "arn:aws:cloudwatch:::alarm:*"
 }
 ]
}
```

Note: AWS services that do support resource-level permissions have different behaviors depending upon the service API call (for example, AWS ELB doesn't support any Describe* service API calls if a resource-level permission is defined). This wasn't done as part of a security feature, and may just mean that AWS has not yet implemented support for that API call. Eucalyptus supports resource-level permissions for all API service calls. This means that the more restrictive AWS IAM access policy for the service that is supported on Eucalyptus should work seamlessly.

Global Cloud Administration (Local vs. Federated)

On AWS, there isn't a concept of a global cloud administrator (on AWS public cloud, AWS is the 'global cloud administrator'). Eucalyptus has the concept of a global cloud administrator. This is represented by the 'eucalyptus' account. This account, along with other Eucalyptus system accounts, are default on each Eucalyptus cloud. Currently, these Eucalyptus system accounts cannot be 'synced' when setting up a federated cloud environment. In order to have the concept of a 'global cloud administrator', one must leverage cross-account roles. Regarding the Eucalyptus IAM policy associated with the global cloud administrator, you can leverage an existing policy that is used by the ResourceAdministrator privileged persona (i.e. role), which is present on each cloud. Below are the high-level steps:

1. Create a non-Eucalyptus account (i.e. global cloud account)
2. Under each 'eucalyptus' account on each cloud (i.e. region), do the following:
   a. Assume Eucalyptus system credentials:

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

   b. Grant user access to ResourceAdministrator role (e.g. admin user of development-operations account):

   ```
   # clcadmin-grant-admin-access -a development-operations -u admin
   ResourceAdministrator
   ```

3. Using the credentials of the user that has been granted access to the ResourceAdministrator role, there are a couple of ways to access the ResourceAdministrator role through Euca2ools or programmatically.

Using Euca2ools

Use `euare-assumerole` to assume the role of the ResourceAdministrator of that region (i.e. cloud).

Note: The ARN associated with the ResourceAdministrator will be the same for each cloud (e.g. eucalyptus:role/eucalyptus/ResourceAdministrator). To use the role for different regions (i.e. clouds), leverage the endpoints for that region. For example:

```
# euare-assumerole  eucalyptus:role/eucalyptus/ResourceAdministrator --region
devops-admin@asap-rocky-2013
# euare-assumerole  eucalyptus:role/eucalyptus/ResourceAdministrator --region
devops-admin@asap-rocky-2015
```

Programmatically

Once this is in place, the user can leverage python boto to do the following:

- Leverage boto configuration file that contains global admin credentials and defines the multiple endpoints associated with each region in Eucalyptus. For example:

```
# cat .boto
[Credentials]
aws_access_key_id = AKIABMYJ35W7GPDOZ2YJ
aws_secret_access_key = 5clOBWeAIwUia4PNVip1CX157nT2ymnjlu12Wn5l
[Boto]
is_secure = False
```

```
endpoints_path = /root/boto-federated-endpoints.json
# cat boto-federated-endpoints.json
{
 "autoscaling": {
 "asap-rocky-2013": "autoscaling.c-40.autoqa.qa1.eucalyptus-systems.com",
 "asap-rocky-2015": "autoscaling.a-41.autoqa.qa1.eucalyptus-systems.com"
 },
 "cloudformation": {
 "asap-rocky-2013": "cloudformation.c-40.autoqa.qa1.eucalyptus-systems.com",
 "asap-rocky-2015": "cloudformation.a-41.autoqa.qa1.eucalyptus-systems.com"
 },
 "cloudwatch": {
 "asap-rocky-2013": "cloudwatch.c-40.autoqa.qa1.eucalyptus-systems.com",
 "asap-rocky-2015": "cloudwatch.a-41.autoqa.qa1.eucalyptus-systems.com"
 },
 "ec2": {
 "asap-rocky-2013": "compute.c-40.autoqa.qa1.eucalyptus-systems.com",
 "asap-rocky-2015": "compute.a-41.autoqa.qa1.eucalyptus-systems.com"
 },
 "elasticloadbalancing": {
 "asap-rocky-2013": "loadbalancing.c-40.autoqa.qa1.eucalyptus-systems.com",
 "asap-rocky-2015": "loadbalancing.a-41.autoqa.qa1.eucalyptus-systems.com"
 },
 "iam": {
 "asap-rocky-2013": "euare.c-40.autoqa.qa1.eucalyptus-systems.com",
 "asap-rocky-2015": "euare.a-41.autoqa.qa1.eucalyptus-systems.com"
 },
 "s3": {
 "asap-rocky-2013": "objectstorage.c-40.autoqa.qa1.eucalyptus-systems.com",
 "asap-rocky-2015": "objectstorage.a-41.autoqa.qa1.eucalyptus-systems.com"
 },
 "sts": {
 "asap-rocky-2013": "tokens.c-40.autoqa.qa1.eucalyptus-systems.com",
 "asap-rocky-2015": "tokens.a-41.autoqa.qa1.eucalyptus-systems.com"
 },
 "swf": {
 "asap-rocky-2013": "simpleworkflow.c-40.autoqa.qa1.eucalyptus-systems.com",
 "asap-rocky-2015": "simpleworkflow.a-41.autoqa.qa1.eucalyptus-systems.com"
 }
}
```

- Leverage *python boto STS* for a given region to acquire Access Key ID, Secret Key, and Security Token that can be used with any AWS service API implemented by Eucalyptus. Make sure that the given service endpoint URL matches the region that provided the STS credentials. For example:

```
In [1]: import boto.sts
In [2]: sts_connection = boto.sts.connect_to_region('asap-rocky-2013',
port=8773)
In [3]: assumedRoleObject =
sts_connection.assume_role(role_arn="arn:aws:iam::000560243913:role/FederatedCloudAdministrator",
 role_session_name="FederatedDescribeELBPolicyTypes")
In [4]: assumedRoleObject.credentials.access_key
Out[4]: u'AKIACY7V4ZGDNKCEXLQK'
In [5]: assumedRoleObject.credentials.secret_key
Out[5]: u'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'
In [6]: assumedRoleObject.credentials.session_token
Out[6]:
[illegible]
# eulb-describe-lb-policy-types -I AKIACY7V4ZGDNKCEXLQK -S
3VWnfDyBrCqtUAAiZqfvQjACLpdRrReHSkX6gLFu --security-token
[illegible]
 -U http://loadbalancing.c-40.autoqa.qa1.eucalyptus-systems.com:8773/
POLICY_TYPE SSLNegotiationPolicyType Listener policy that defines the ciphers
```

```
 and protocols that will be accepted by the load balancer. This policy can be
 associated only with HTTPS/SSL listeners.
POLICY_TYPE LBCookieStickinessPolicyType Stickiness policy with session
lifetimes controlled by the browser (user-agent) or a specified expiration
period. This policy can be associated only with HTTP/HTTPS listeners.
POLICY_TYPE BackendServerAuthenticationPolicyType Policy that controls
authentication to back-end server(s) and contains one or more policies, such
as an instance of a PublicKeyPolicyType. This policy can be associated only
with back-end servers that are using HTTPS/SSL.
POLICY_TYPE ProxyProtocolPolicyType Policy that controls whether to include
the IP address and port of the originating request for TCP messages. This
policy operates on TCP/SSL listeners only
POLICY_TYPE AppCookieStickinessPolicyType Stickiness policy with session
lifetimes controlled by the lifetime of the application-generated cookie. This
 policy can be associated only with HTTP/HTTPS listeners.
POLICY_TYPE PublicKeyPolicyType Policy containing a list of public keys to
accept when authenticating the back-end server(s). This policy cannot be applied
 directly to back-end servers or listeners but must be part of a
BackendServerAuthenticationPolicyType.
```

Troubleshooting

This section is presented in a Q&A format to provide a quick reference to the most frequently asked questions.

1. **Q.** Can Cloud Administrators federate existing clouds (i.e. clouds that already have non-system Eucalyptus accounts)?

   A. No, this is currently not supported. If a cloud administrator wants to federate an Eucalyptus clouds, this must be done prior to any non-system Eucalyptus account/user/group creation.

2. **Q.** Is Eucalyptus DNS required for federating Eucalyptus clouds?

   A. No, however its highly recommended to enable it.

3. **Q.** Are *Amazon Resource Names* supported for more granular IAM access policies per region?

   A. As of 4.2, no. IAM policies apply globally (for all regions). In order to get more granular IAM access, use availability zone restrictions under each region. For more information, see *Restrict Image to Availability Zone*.

4. **Q.** What services/resources span globally? Which span regionally?

   A. Currently, only Eucalyptus IAM and STS are global services/resources. All other services/resources are region-based (i.e. Eucalyptus cloud-specific). The only resource that can be either global or regional are keypairs. This is because users can import the same keypair to each region, therefore, the keypair is globally accessible. For additional information, please refer to the AWS EC2 Documentation regarding *Resource Locations*.

5. **Q.** Are Eucalyptus system accounts global in a federated setup?

   A. No. Any Eucalyptus system account is limited to that region. Examples of Eucalyptus system accounts are as follows:

   - eucalyptus
   - (eucalyptus)blockstorage
   - (eucalyptus)aws-exec-read
   - (eucalyptus)cloudformation

6. **Q.** Is *CopySnapshot* and *CopyImage* supported?

   A. No. There have been no improvements associated with Object Storage Gateway (OSG) regarding cross-regional behavior similar to AWS.

7. **Q.** If a user uploads an object to an Object Storage Gateway in one region, will copies show up in other regions (similar to the behavior on AWS)?

A. No, this is currently unsupported.

8. **Q.** Is LDAP/AD integration supported once Eucalyptus clouds have been federated?

   A. No, this feature is not supported.

9. **Q.** Do federated Eucalyptus clouds follow the same *limitations as AWS IAM* globally?

   A. No, Eucalyptus IAM limitations are regionally scoped.

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 Best Practices

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 IAM 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 IAM 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*.

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-4.2/nginx.conf`.

   ```
   cp /usr/share/doc/eucaconsole-4.2/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.

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.

   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 8443	Port for getting user credentials on the CLC. Configurable with `euctl`.
TCP 8772	DEBUG ONLY: JMX port. This is disabled by default, and can be enabled with the `--debug` or `--jmx` options for `CLOUD_OPTS`.
TCP 8773	Web services port for the CLC, user-facing services (UFS), object storage gateway (OSG), Walrus SC; also used for external and internal communications by the CLC and Walrus. Configurable with `euctl`.

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.ini` file.
TCP 16514	TLS port on Node Controller, required for node migrations
UDP 7500	Port for diagnostic probing on CLC, UFS, OSG, Walrus SC
UDP 8773	Membership port for SC, any UFS, Walrus
UDP 8778	The bind port used to establish multicast communication
TCP/UDP 53	DNS port on UFS

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/4.2/centos/6/x86_64/eucalyptus-release-4.2-1.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-walrusbackend`	Add a Walrus Backend to Eucalyptus	No
`--deregister-walrusbackend`	Remove a Walrus Backend from Eucalyptus	No
`--register-sc`	Add Storage Controller to Eucalyptus	No

Option	Description	Required
`--deregister-sc`	Remove Storage Controller from Eucalyptus	No
`--list-walrusbackends`	List all registered Walrus Backends	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.

euctl

The euctl utility retrieves cloud state and allows cloud administrators to set cloud state. The state to be retrieved or set is described using a Management Information Base (MIB) style name, described as a dotted set of components. See usage notes and examples below.

Note: The `euctl` command is intended to replace the deprecated `euca-describe-properties` and `euca-modify-property` commands.

Syntax

```
euctl [-A] [-r] [-d] [-s] [-n] [-q] [--edit | --dump] [--format {json,yaml}]
            [-U URL] [--region USER@REGION] [-I KEY_ID] [-S KEY]
            [--security-token TOKEN] [--debug] [--debugger] [--version] [-h]
            [NAME[=VALUE|=@FILE] [NAME[=VALUE|=@FILE] ...]]
```

Options

Option	Description	Required
NAME[=*VALUE*|=*@FILE*] *name=value*	Output the specified variable, and where a value is given, attempt to set it to the specified value. Specify a filename to read the values from a file; for example: `myproperty=@myvaluefile`	Positional
`-A, --all-types`	List all the known variable names, including structures. Those with string or integer values will be output as usual; for the structured values, the methods of retrieving them are given.	No
`-r, --reset`	Resets the named property to the default value.	No
`-d`	Show variables' default values instead of their current values.	No
`-s`	Show variables' descriptions instead of their current values.	No
`-n`	Suppress output of the variable name. This is useful for setting shell variables.	No
`-q`	Suppress all output when setting a variable. This option overrides the behavior of the `-n` parameter.	No
`--edit`	Edit a structured variable interactively. Only one variable may be edited per invocation. When looking for an editor, the program will first try the environment variable VISUAL, then the environment variable EDITOR, and finally the default editor, vi.	No
`--dump`	Output the value of a structured variable in its entirety.	No
`--format` *format*	Try to use the specified format when displaying a structured variable. Valid values: `json` | `yaml` Default value: `json`	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.

Examples

When retrieving a variable, a subset of the MIB name may be specified to retrieve a list of variables in that subset. For example, to list all the dns variables:

```
euctl dns
```

When setting a variable, the MIB name should be followed by an equal sign and the new value:

```
euctl dns.enabled=true
```

To reset a varable to its default value, specify the -r option:

```
euctl -r dns.enabled
```

The information available from euctl consists of integers, strings, and structures. The structured information can only be retrieved by specialized programs and, in some cases, this program's `--edit` and `--dump` options.

euca-describe-properties

This command lists properties.

Note: This command is deprecated. Use the *euctl* command instead.

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`.
`-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.

Note: This command is deprecated. Use the *euctl* command instead.

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.
`-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`.
`-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
`-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

Option	Description	Required
`--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.
`--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`.
`-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.
`-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

Option	Description	Required
`-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.access_keys_limit	Limit for access keys per user
authentication.authorization_cache	Authorization cache configuration, for credentials and authorization metadata
authentication.authorization_expiry	Default expiry for cached authorization metadata
authentication.authorization_reuse_expiry	Default expiry for re-use of cached authorization metadata on failure
authentication.credential_download_generate_certificate	Strategy for generation of certificates on credential download (Never | Absent | Limited)
authentication.credential_download_host_match	CIDR to match against for host address selection
authentication.credential_download_port	Port to use in service URLs when 'bootstrap.webservices.port' is not appropriate.
authentication.default_password_expiry	Default password expiry time
authentication.max_policy_size	Maximum size for an IAM policy (bytes)
authentication.signing_certificates_limit	Limit for signing certificates per user
authentication.system_account_quota_enabled	Process quotas for system accounts
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	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
bootstrap.async.future_listener_debug_limit_secs	Number of seconds a future listener can execute before a debug message is logged.

Property	Description
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.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_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.

Property	Description
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).
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.

Property	Description
bootstrap.webservices.disabled_soap_api_components	List of services with disabled SOAP APIs.
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.log_requests	Enable request logging.
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 time-out.
bootstrap.webservices.pipeline_max_query_request_size	Maximum Query Pipeline http chunk size (bytes).
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 accommodate collisions.
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.unknown_parameter_handling	Request unknown parameter handling (default | ignore | error)
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.

Property	Description
bootstrap.webservices.ssl.server_ssl_protocols	SSL protocols for webservices.
bootstrap.webservices.ssl.user_ssl_ciphers	SSL ciphers for external use.
bootstrap.webservices.ssl.user_ssl_default_cas	Use default CAs with SSL for external use.
bootstrap.webservices.ssl.user_ssl_enable_hostname_verification	SSL hostname validation for external use.
bootstrap.webservices.ssl.user_ssl_protocols	SSL protocols for external use.
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.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.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.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.images.cleanupperiod	The period between runs for clean up of deregistered images.
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.images.maxmanifestsizebytes	The maximum allowed image manifest size in bytes

Property	Description
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 send from the cc to the clc
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_index_pending_timeout	Minutes before a pending index allocation timesout and is released.
cloud.network.network_tag_pending_timeout	Minutes before a pending tag allocation timesout and is released.
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_reachability_timeout	Amount of time (in minutes) before a VM which is not reported by a cluster will fail a reachability test.
cloud.vmstate.instance_subdomain	Subdomain to use for instance DNS.
cloud.vmstate.instance_timeout	Amount of time (default unit 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	Default prefix to use for instance / network interface 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).

Property	Description
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.
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.unknown_instance_handlers	Comma separated list of handlers to use for unknown instances ('restore', 'restore-failed', 'terminate', 'terminate-done')
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.
cloud.vmtypes.format_ephemeral_storage	Format first ephemeral disk by defaut with ext3
cloud.vpc.defaultvpc	Enable default VPC.
cloud.vpc.networkaclspervpc	Maximum number of network ACLs for each VPC.
cloud.vpc.routespertable	Maximum number of routes for each route table.

Property	Description
cloud.vpc.routetablespervpc	Maximum number of route tables for each VPC.
cloud.vpc.rulespernetworkacl	Maximum number of rules per direction for each network ACL.
cloud.vpc.rulespersecuritygroup	Maximum number of associated security groups for each network interface .
cloud.vpc.securitygroupspernetworkinterface	Maximum number of associated security groups for each network interface .
cloud.vpc.securitygroupspervpc	Maximum number of security groups for each VPC.
cloud.vpc.subnetspervpc	Maximum number of subnets for each VPC.
cloudformation.autoscaling_group_deleted_max_delete_retry_secs	The amount of time (in seconds) to wait for an autoscaling group to be deleted after deletion)
cloudformation.autoscaling_group_zero_instances_max_delete_retry_secs	The amount of time (in seconds) to wait for an autoscaling group to have zero instances during delete
cloudformation.instance_attach_volume_max_create_retry_secs	The amount of time (in seconds) to wait for an instance to have volumes attached after creation)
cloudformation.instance_running_max_create_retry_secs	The amount of time (in seconds) to wait for an instance to be running after creation)
cloudformation.instance_terminated_max_delete_retry_secs	The amount of time (in seconds) to wait for an instance to be terminated after deletion)
cloudformation.max_attributes_per_mapping	The maximum number of attributes allowed in a mapping in a template
cloudformation.max_mappings_per_template	The maximum number of mappings allowed in a template
cloudformation.max_outputs_per_template	The maximum number of outputs allowed in a template
cloudformation.max_parameters_per_template	The maximum number of outputs allowed in a template
cloudformation.max_resources_per_template	The maximum number of resources allowed in a template
cloudformation.region	The value of AWS::Region and value in CloudFormation ARNs for Region
cloudformation.request_template_body_max_length_bytes	The maximum number of bytes in a request-embedded template
cloudformation.request_template_url_max_content_length_bytes	The maximum number of bytes in a template referenced via a URL
cloudformation.security_group_max_delete_retry_secs	The amount of time (in seconds) to retry security group deletes (may fail if instances from autoscaling group)

Property	Description
cloudformation.subnet_max_delete_retry_secs	The amount of time (in seconds) to retry subnet deletes
cloudformation.swf_activity_worker_config	JSON configuration for the cloudformation simple workflow activity worker
cloudformation.swf_domain	The simple workflow service domain for cloudformation
cloudformation.swf_tasklist	The simple workflow service task list for cloudformation
cloudformation.url_domain_whitelist	A comma separated white list of domains (other than Eucalyptus S3 URLs) allowed by CloudFormation URL parameters
cloudformation.volume_attachment_max_create_retry_secs	The amount of time (in seconds) to wait for a volume to be attached during create)
cloudformation.volume_available_max_create_retry_secs	The amount of time (in seconds) to wait for a volume to be available after create)
cloudformation.volume_deleted_max_delete_retry_secs	The amount of time (in seconds) to wait for a volume to be deleted)
cloudformation.volume_detachment_max_delete_retry_secs	The amount of time (in seconds) to wait for a volume to detach during delete)
cloudformation.volume_snapshot_complete_max_delete_retry_secs	The amount of time (in seconds) to wait for a snapshot to be complete (if specified as the deletion policy) before a volume is deleted)
cloudformation.wait_condition_bucket_prefix	The prefix of the bucket used for wait condition handles
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.search	Comma separated list of domains to search, OS settings used if none specified (change requires restart)
dns.server	Comma separated list of nameservers, OS settings used if none specified (change requires restart)
dns.server_pool_max_threads	Server worker thread pool max.
dns.server_pool_max_threads	Server worker thread pool max.
dns.instancedata.enabled	Enable the instance-data resolver. **Note:** `dns.enable` must also be 'true'

Property	Description
dns.ns.enabled	Enable the NS resolver. **Note:** `dns.enable` must also be 'true'
dns.recursive.enabled	Enable the recursive DNS resolver. **Note:** `dns.enable` must also be 'true'
dns.services.enabled	Enable the service topology resolver. **Note:** `dns.enable` must also be 'true'
dns.split_horizon.enabled	Enable the split-horizon DNS resolution for internal instance public DNS name queries. **Note:** `dns.enable` must also be 'true'
dns.spoof_regions.enabled	Enable the spoofing resolver which allows for AWS DNS name emulation for instances.
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.
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.

Property	Description
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.queue_timeout	Channel buffer queue timeout (in seconds)
objectstorage.s3client.buffer_size	Internal S3 client buffer size
objectstorage.s3client.connection_timeout_ms	Internal S3 client connection timeout in ms
objectstorage.s3client.max_connections	Internal S3 client maximum connections
objectstorage.s3client.max_error_retries	Internal S3 client maximum retries on error
objectstorage.s3client.socket_read_timeout_ms	Internal S3 client socket read timeout in ms
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)

Property	Description
<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.
<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.cephconfigfile	Absolute path to Ceph configuration (ceph.conf) file. Default value is '/etc/ceph/ceph.conf'
<partition>.storage.cephkeyringfile	Absolute path to Ceph keyring (ceph.client.eucalyptus.keyring) file. Default value is '/etc/ceph/ceph.client.eucalyptus.keyring'
<partition>.storage.cephsnapshotpools	Ceph storage pool(s) made available to Eucalyptus for EBS snapshots. Use a comma separated list for configuring multiple pools. Default value is 'rbd'
<partition>.storage.cephuser	Ceph username employed by Eucalyptus operations. Default value is 'eucalyptus'
<partition>.storage.cephvolumepools	Ceph storage pool(s) made available to Eucalyptus for EBS volumes. Use a comma separated list for configuring multiple pools. Default value is 'rbd'
<partition>.storage.chapuser	User ID for CHAP authentication
<partition>.storage.dasdevice	Direct attached storage device location
<partition>.storage.maxconcurrentsnapshots	Maximum number of snapshots processed on the block storage backend at a given time
<partition>.storage.maxconcurrentsnapshottransfers	Maximum number of snapshots that can be uploaded to or downloaded from objectstorage gateway at a given time
<partition>.storage.maxconcurrentvolumes	Maximum number of volumes processed on the block storage backend at a given time

Property	Description
<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.ncpaths	iSCSI Paths for NC. Default value is 'nopath'
<partition>.storage.readbuffersizeinmb	Buffer size in MB for reading data from snapshot when uploading snapshot to objectstorage gateway
<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 SC. Default value is 'nopath'
<partition>.storage.shouldtransfersnapshots	Should transfer snapshots
<partition>.storage.snapexpiration	Time interval in minutes after which Storage Controller metadata for snapshots that have been physically removed from the block storage backend will be deleted
<partition>.storage.snapshotpartsizeinmb	Snapshot part size in MB for snapshot transfers using multipart upload. Minimum part size is 5MB
<partition>.storage.snapshottransfertimeoutinhours	Snapshot upload wait time in hours after which the upload will be cancelled
<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 milli seconds for storage operations
<partition>.storage.volexpiration	Time interval in minutes after which Storage Controller metadata for volumes that have been physically removed from the block storage backend will be deleted
<partition>.storage.volumesdir	Storage volumes directory.
<partition>.storage.writebuffersizeinmb	Buffer size in MB for writing data to snapshot when downloading snapshot from objectstorage gateway
<partition>.storage.zerofillvolumes	Should volumes be zero filled.
region.region_enable_ssl	Enable SSL (HTTPS) for regions.

Property	Description
region.region_name	Region name.
region.region_ssl_ciphers	Ciphers to use for region SSL
region.region_ssl_default_cas	Use default CAs for region SSL connections.
region.region_ssl_protocols	Protocols to use for region SSL
region.region_ssl_verify_hostnames	Verify hostnames for region SSL connections.
reporting.data_collection_enabled	Set this to false to stop reporting from populating new data
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_mins	How often the reporting system requests information from the cluster controller
services.database.appendonlyhost	host address of the backend database for append-only data
services.database.appendonlypassword	password of the backend database for append-only data
services.database.appendonlyport	port number of the backend database for append-only data
services.database.appendonlysslcert	ssl certificate to use when connecting to the backend database for append-only data
services.database.appendonlyuser	user name of the backend database for append-only data
services.database.worker.availability_zones	availability zones for database server
services.database.worker.configured	Configure DB service so a VM can be launched.If something goes south with the service there is a chance that setting it to false and back to true would solve issues
services.database.worker.expiration_days	days after which the VMs expire
services.database.worker.image	EMI containing database server
services.database.worker.init_script	bash script that will be executed before serviceconfiguration and start up
services.database.worker.instance_type	instance type for database server
services.database.worker.keyname	keyname to use when debugging database server
services.database.worker.ntp_server	address of the NTP server used by database server
services.database.worker.volume	volume containing database files
services.imaging.import_task_expiration_hours	expiration hours of import volume/instance tasks

Property	Description
services.imaging.import_task_timeout_minutes	expiration time in minutes of import tasks
services.imaging.worker.availability_zones	availability zones for imaging worker
services.imaging.worker.configured	Prepare imaging service so a worker can be launched.If something goes south with the service there is a big 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
services.imaging.worker.healthcheck	enabling imaging worker healthcheck
services.imaging.worker.image	EMI containing imaging worker
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
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 imaging wokrers
services.imaging.worker.log_server_port	UDP port that log server is listerning to
services.imaging.worker.log_server_port	UDP port that log server is listerning to
services.imaging.worker.ntp_server	address of the NTP server used by imaging worker
services.loadbalancing.dns_resolver_enabled	Enable the load balancing DNS resolver. **Note:** `dns.enable` must also be 'true'
services.loadbalancing.dns_subdomain	loadbalancer dns subdomain
services.loadbalancing.dns_ttl	loadbalancer dns ttl value
services.loadbalancing.restricted_ports	The ports restricted for use as a loadbalancer port. Format should be port(, port) or port-port
services.loadbalancing.vm_per_zone	number of VMs per loadbalancer zone
services.loadbalancing.worker.app_cookie_duration	duration of app-controlled cookie to be kept in-memory (hours)
services.loadbalancing.worker.backend_instance_update_interval	interval for updating backend instance state
services.loadbalancing.worker.cache_duration	duration of cached data delivered to workers
services.loadbalancing.worker.cw_put_interval	interval for updating CW metrics
services.loadbalancing.worker.expiration_days	the days after which the loadbalancer Vms expire
services.loadbalancing.worker.image	EMI containing haproxy and the controller
services.loadbalancing.worker.init_script	bash script that will be executed before service configuration and start up
services.loadbalancing.worker.instance_type	instance type for loadbalancer instances
services.loadbalancing.worker.keyname	keyname to use when debugging loadbalancer VMs

Property	Description
services.loadbalancing.worker.lb_poll_interval	interval for distributing ELB data to haproxy VM
services.loadbalancing.worker.ntp_server	the address of the NTP server used by loadbalancer VMs
services.simpleworkflow.activitytypesperdomain	Maximum number of activity types for each domain.
services.simpleworkflow.deprecatedactivitytyperetentionduration	Deprecated activity type retention time.
services.simpleworkflow.deprecateddomainretentionduration	Deprecated domain minimum retention time.
services.simpleworkflow.deprecatedworkflowtyperetentionduration	Deprecated workflow type minimum retention time.
services.simpleworkflow.openactivitytasksperworkflowexecution	Maximum number of open activity tasks for each workflow execution.
services.simpleworkflow.opentimersperworkflowexecution	Maximum number of open timers for each workflow execution.
services.simpleworkflow.openworkflowexecutionsperdomain	Maximum number of open workflow executions for each domain.
services.simpleworkflow.systemonly	Service available for internal/administrator use only.
services.simpleworkflow.workflowexecutionduration	Maximum workflow execution time.
services.simpleworkflow.workflowexecutionhistorysize	Maximum number of events per workflow execution.
services.simpleworkflow.workflowexecutionretentionduration	Maximum workflow execution history retention time.
services.simpleworkflow.workflowtypesperdomain	Maximum number of workflow types for each domain.
stats.config_update_check_interval_seconds	Interval, in seconds, at which the sensor configuration is checked for changes
stats.enable_stats	Enable Eucalyptus internal monitoring stats
stats.event_emitter	Internal stats emitter FQ classname used to send metrics to monitoring system
stats.file_system_emitter.stats_data_permissions	group permissions to place on stats data files in string form. eg. rwxr-x--x
stats.file_system_emitter.stats_group_name	group name that owns stats data files
storage.global_total_snapshot_size_limit_gb	Maximum total snapshot capacity (GB)
system.dns.dnsdomain	Domain name to use for DNS.
system.dns.nameserver	Nameserver hostname.
system.dns.nameserveraddress	Nameserver ip address.
system.dns.nameserveraddress	Nameserver ip address.
system.dns.registrationid	Unique ID of this cloud installation.
system.exec.io_chunk_size	Size of IO chunks for streaming IO

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.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.https_ciphers	SSL ciphers for HTTPS listener.
www.https_port	Listen to HTTPs on this port.
www.https_protocols	SSL protocols for HTTPS listener.

Advanced Storage Configuration

This section covers advanced storage provider configuration options.

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`

Administration Guide History

This section contains information about changes to the administration documentation in this release.

Section / Topic	Description of Change	Date Changed
Eucalyptus Configuration Properties	Added new properties introduced in 4.2 and removed some old ones.	October 22, 2015
Manage Users and Groups	Moved to IAM Guide	October 22, 2015

Index

C

E

F

T