EUCALYPTUS

Troubleshooting 3.4.2 Guide

2014-02-23 Eucalyptus Systems

Contents

Troubleshooting Overview....................3

Log Files....................4

Network Information....................7

Walrus and Storage....................8

Access and Identities....................9

Windows Images....................10

Instances....................12

High Availability....................13

Recovering from a Failure: Walrus....................14

Troubleshooting Overview

This section covers common management or configuration problems and solutions for fixing these problems.

To troubleshoot Eucalyptus, the administrator must know the location of the Eucalyptus components, that is, on which machine each component is installed. The administrator must have root access to each machine hosting the components and must understand the network configuration connecting the components.

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.

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/`.

Here are the relevant logs for each component:

Cloud controller (CLC), Walrus, Storage controller (SC), and VMware Broker

- `cloud-output.log`
- `euca_imager.log`

These components also include specialized developer log files. These are not relevant to troubleshooting a production system, and are not affected by any log level settings. These logs include the following:

- `cloud-debug.log`
- `cloud-error.log`
- `cloud-exhaust.log`
- `cloud-extreme.log`

Cluster controller (CC)

- `cc.log`
- `axis2c.log`
- `httpd-cc_error_log`

Node controller (NC)

- `nc.log`
- `axis2c.log`
- `httpd-nc_error_log`
- `euca_test_nc.log`

System Logs

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/`

Configuring Logging

For the cluster controller and the node controller, log level is configured 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, the log level can be configured by passing an appropriate `--log-level` argument in the init script. It can also be dynamically changed using `euca-modify-property` to 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 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 broker 1008 daemon ntpd
```

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

Walrus and Storage

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

Walrus decryption failed.

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

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

If you are running this kernel:

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

Walrus physical disk is not large enough.

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

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

Access and Identities

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

Need to verify an existing LIC file.

1. Enter the following command:

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

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

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

Windows Images

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

Properties

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

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

The properties that might affect registering Windows images are:

- `walrus.storagemaxbucketsizeinmb`: max bucket size enforced by Walrus; should be larger than a Windows image
- `walrus.storagemaxcachesizeinmb`: total size of all images that is cached in Walrus; should be larger than the sum of all images (Windows/Linux) in Walrus
- walrus.storagemaxtotalsnapshotsizeingb: if a Windows image is a type of EBS-backed EMI, this should be large enough to store all EBS backed images
- `{PARTITION}.storage.maxvolumesizeingb`: if a Windows image is a type of EBS-backed EMI, this should be large enough to store the image

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

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

Image Preparation

`euca-bundle-image` hangs

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

`euca-upload-bundle` fails

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

Instance Launch and Login

Instance stays in pending

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

Instance stay in pending and goes to shutdown

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

Instance is running, but not accessible using Remote Desktop.

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

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

Finding the login username and password

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

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

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

Instance is not attached to an Active Directory domain

- Make sure the parameters set in Windows integration service are correct. One way to verify them is to log in the instance using Administrator password and manually attach the instance to the domain (System Properties -> Computer Name) using the same parameters.
- Make sure `VNET_DNS` in eucalyptus.conf is set to the domain controller (refer to User Guide: Configure Active Directory).

Disk and Volume

Ephemeral disks are not visible in the Windows

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

EBS volume is attached, but not visible in the Windows

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

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

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

`euca-bundle-instance` fails

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

Instances

This topic contains information to help you troubleshoot your instances.

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

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

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

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

NC does not recalculate disk size correctly

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

For example:

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

High Availability

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

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

1. Stop the secondary CLC.

```
service eucalyptus-cloud stop
```

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

```
service eucalyptus-cloud start
```

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

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

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-walrus --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.

Index

F

fail *14*
recovering from *14*

T

troubleshooting *4*, *7–10*, *12–13*
access and identities *9*

troubleshooting *(continued)*
high availability *13*
instances *12*
log files *4*
network information *7*
Walrus and storage *8*
Windows images *10*