# FleSSR Project: Installing Eucalyptus Open Source Cloud Solution at Oxford e-Research Centre

Matteo Turilli, David Wallom

Eucalyptus is available in two versions: open source and enterprise. Within this document the term 'Eucalyptus' will be referring to the open source version.

Eucalyptus is provided by Eucalyptus Systems Inc as source code or as binaries already compiled and packaged for the RHEL-5, CentOS-5, OpenSuse-11, Debian "squeeze" and Fedora Linux-based distributions.

Canonical Ltd offers a repackaged version of the Eucalyptus source code with some added features and a tailored installer. In the following we will consider the installation of Eucalyptus binary packages on the CentOS-5 and Ubuntu-10.04/10.10 Operating Systems.

Eucalyptus is a modular cloud solution designed around five main components called 'controllers':

1. Cloud Controller (CLC): "The cloud controller is a Java program that offers EC2-compatible SOAP and "Query" interfaces, as well as a Web interface to the outside world. In addition to handling incoming requests, the cloud controller performs high-level resource scheduling and system accounting".[1]
2. Walrus Controller (Walrus): "Walrus, also written in Java, implements bucket-based storage, which is available outside and inside a cloud through S3-compatible SOAP and REST interfaces".[2]
3. Cluster Controller (CC): is written in C and runs as a Web service inside Apache. It provides "cluster-level scheduling and network control".[3]
4. Storage Controller (SC):  written in Java, it provides Elastic Block Storage (EBS) -style block-based storage.
5. Node Controller (NC): also an Apache-based Web service written in C. It drives the chosen hypervisor (KVM or Xen) on each node the VMs are run.

All the controllers communicate via SOAP with WS-security.

Eucalyptus offers a set of functionalities loosely comparable to those offered by Amazon Elastic Compute Cloud (EC2). Such functionalities include:

1. Security Groups: users can define groups with access rules indicating what port can be accessed from which source IP(s). Multiple Virtual Machines (VMs) can then be instantiated and associated to a defined group. In this way, a security group works analogously to a firewall put in front of one or more VM. Crucially, such a 'firewall' is managed directly by the owner of the VM(s).
2. Elastic IPs: one or more public IP can be associated to a running VM. It should be noted that the VM itself has always a single Network Interface to which is assigned a private IP. The public IP is associated to the VM by the CC via iptable or vlan, depending on the type of network configuration chosen for Eucalyptus.

---

1  http://open.eucalyptus.com/wiki/EucalyptusInstallation_v2.0
2  Ibid.
3  Ibid.

3. VM Isolation: VM network traffic is isolated for the traffic of other VMs belonging to a different security group. Isolation is obtained either via iptables or vlan, depending on the type of network configuration chosen for Eucalyptus.

## *Architectures*

- Single machine; Eucalyptus can be installed on a single physical machine but Security Groups, Elastic IPs and VM Isolation are not available because of network configuration limitations.
- Two machines; A more useful testing setup involves two physical machines, one running the CLC, Walrus, CC, SC while the other offering an hypervisor driven by the NC in order to run the VMs. While the NC controller and the hypervisor are usually run on a physical machine, CLC+Walrus+CC+SC can be run on a virtual machine offering appropriate specifications.
- Three to Five physical machines + hypervisor nodes; A pre or production infrastructure requires one machine dedicated to each controller, with the exception of CC and CS that can be run on a single machine. In this set up, each couple CC+CS can run up to 256 VMs. If the production conditions require more VMs, another CC+CS couple should be deployed on one or two physical machine. A single CLC can run multiple CC+CS and Walruses. Each NC is associated to a single CC and only one CS can be paired to a given CC. Eucalyptus 2.x does not support high availability for either CLC or Walrus but Eucalyptus 3 is supposed to introduce such capability. It is not clear (Jul 2011) whether high-availability will be limited to the Enterprise edition or will be made available also through the open source version.

## *Requirements*

Eucalyptus Systems do not provide in their guides minimal or suggested hardware requirements. Ubuntu community offers a page with some guidelines[4] and Intel published a case study[5] in which the hardware that has been used is specified. Our investigations confirmed that, for testing purposes, Eucalyptus 1.6.3 and 2.0.2 can effectively run on hardware with the minimal and suggested requirements indicated by the Ubuntu Community. For production purpose, 4GB of RAM should be considered a minimal requirement for the controllers written in Java (CLC, CS, Walrus). A production environment would greatly benefit also from a multi Gb network interconnection between SC, CC and NCs, fast I/O devices/configuration for the volume or file system in which the VMs are copied and run by the NC and redundant configurations for the Walrus and SC disk subsystems.

## *Implemented Architectures*

Through the FleSSR project three Eucalyptus infrastructures have been deployed, with one at each of Eduserv, University of Reading and University of Oxford. All the infrastructures had testing and development purposes. The development required by the two use cases has been conducted on the Reading and Oxford infrastructures.

Oxford has tested Eucalyptus both on CentOS-5 (Fig. 1) and Ubuntu 10.4 subsequently updated to Ubuntu 10.10 (Fig. 2).

## *Eucalyptus Installation on CentOS-5*

**Preparation**

---

4   https://help.ubuntu.com/community/UEC/CDInstall
5   http://www.intel.com/Assets/PDF/general/icb_ra_cloud_computing_canonical_ubuntu.pdf

1. OS installation

   a) OS should be installed on all the physical machines following the installation procedure of the chosen Linux distribution.

   b) KVM or Xen hypervisors must be installed on any physical machine running the NC.

2. Networking

   a) Eucalyptus requires Internet connectivity for CLC and Walrus and a private network connectivity for CC, SC and NC.

   b) Eucalyptus dynamically updates the iptables rules on the CC. Modification of the iptables rules on the CC are likely to interfere with Eucalyptus networking.

   c) Eucalyptus does not offer proxy/masquerading facilities for the NCs. Every NCs will have to be configured so to access the Internet independently from the Eucalyptus installation.

   d) Eucalyptus requires a dedicated subnet for its private network connectivity.

   e) Eucalyptus requires a dedicated number of public IPs that will be assigned to the served VMs. The modalities of such an assignment depend on the type of network configuration chosen for Eucalyptus.

   f) If the MANAGED type of Eucalyptus networking configuration is chosen, switches must be configured with a number of vlans or so to 'transparent' to vlan tagging. The number of vlans should be equal to or greater of the number of security groups allowed. The number of security group allowed is a function of the size of the subnet dedicated to Eucalyptus and the value given to the configuration parameter VNET_ADDRSPERNET. The range of the number of the vlans tags can be set in the Eucalyptus web interface, under 'Configuration', at the voice 'Use VLAN tags'.

**Installation**

Eucalyptus Systems provides detailed and updated instructions for the following Linux-based distributions:

- RHEL-5, CentOS-5: http://open.eucalyptus.com/wiki/EucalyptusInstallationCentos_v2.0

- OpenSuse-11: http://open.eucalyptus.com/wiki/EucalyptusInstallationOpensuse_v2.0

- Debian "squeeze": http://open.eucalyptus.com/wiki/EucalyptusInstallationDebian_v2.0

- Fedora-12: http://open.eucalyptus.com/wiki/EucalyptusInstallationFedora_v2.0

It should be noted that the CLC installation set the admin password to 'admin' by default. If the installation is done on a machine that can be reached from the Internet, such policy is not safe.

## *Eucalyptus Installation on Ubuntu 10.10*

The Eucalyptus version installed by Ubuntu is called Ubuntu Enterprise Cloud (UEC). The installation of UEC Ubuntu 10.4 and then 10.10 offers:

- a dedicated, Eucalyptus-aware OS installer;

- an automatic and transparent registration system for Walruses, CCs, SCs and NCs. Where the default installation procedure for Eucalyptus requires to proceed with such registrations by hand, Ubuntu takes advantage of the Avahi implementation of Zeroconf in order to automate this process.

- a semi-automated installation procedure for the NCs;

- an automated set up of the eucalyptus user credentials;

- an automated set up of the admin user on each subsequent to that of the CLC.


**Preparation**

The preparation steps described for the Eucalyptus installation on CentOS-5 apply to the installation of UEC. The following should also be taken into account:

- The installation of Walrus(es), CC(s), SC(s) and NC(s) require access to the public interface of the CLC. This implies that if the CLC is not installed behind a proxy, public network access will be required in order to install Walrus(es), CC(s), SC(s) and NC(s).

- If the Eucalyptus controllers are not installed behind a proxy, the Avahi daemon will advertise (broadcast) the availability of the controllers to the whole broadcast subnet. Because of the Zeroconf properties, each advertised Eucalyptus controller will be picked up and registered by any CLC listening on the broadcasting subnet. The CLC will try to authenticate each registered controller and, in presence of a failure, will behave unpredictably, often becoming unable to use the other registered and authenticated controllers.

- The dedicated installer for UEC does not install the atp demon. This demon is likely to be required by the latest security patch applied to Eucalyptus because of the strict time-synchronisation policies it introduces.

**Installation**

The procedure to install UEC is described in the document "Getting Started with Ubuntu Enterprise Cloud".[6]

It should be noted that the procedure "Advanced installation using the packages on separate servers" is outdated and will not produce a working installation. The document provided by Intel and titled "Intel® Cloud Builder Guide to Cloud Design and Deployment on Intel® Platforms - Ubuntu Enterprise Cloud"[7] should instead be used as a starting point.

For an example of an architecture based on separate server see Fig.1, 2 and 3 in the paper above.

**Issues**

- The upgrade following a successful installation of UEC on Ubunt10.10 fails hanging forever. It is necessary to:

    o kill manually the installation process of the Eucalyptus package;

    o reissue the upgrading command (i.e. apt-get dist-upgrade);

    o restart every Eucalyptus process after the successful upgrade.

- Installing CC and SC into a dedicated machine leads to an error in starting SC that, in turn, makes the CC behaving unpredictably. Installing CC and SC on two separate machines solved the problem.

- When attaching an Elastic Block Storage (EBS) volume to a VM, KVM will fail to load the scsi device if it is named /dev/sd*. The device should be names /dev/vd* starting from /dev/vda.

---

[6] https://help.ubuntu.com/community/UEC
[7] http://www.intel.com/Assets/PDF/general/icb_ra_cloud_computing_canonical_ubuntu.pdf

## *Brokering layer*

The development of the two use cases of the FleSSR project required the use of a cloud brokering layer. The EoverI[8] team has worked closely with the other project members in order to develop a Eucalyptus module for their eZeel[9] cloud infrastructure system. The adoption and deployment of eZeel did not require any modification to the three Eucalyptus infrastructures of the project.

## *Management, Monitor and Accounting*

Linux OS management and Eucalyptus management are separated issues. In order to manage and monitor Eucalyptus we have used:

- a Firefox add-on called Hybridfox;

- shell scripts for parsing and monitoring the logs stored in /var/log/eucalyptus, especially cc.log, cloud.log, cloud-error.log and nc.log;

- euca-tools, even if almost all their functionalities have been replaced by Hybridfox;

- Ganglia;

- Nagios with dedicated probes scripts (it should be noted that the Nagios sample scripts provided with the Eucalyptus source are outdated and basically useless for Eucalyptus version 2.*).

Accounting and per-user resource quota allocation are provided only with the Enterprise version of Eucalyptus. An in-house solution has been developed for the accounting.

The accounting solution for Eucalyptus has been designed keeping in mind three requirements:

1. The source code of Eucalyptus should not be modified;

2. The records should be consistent with the OGF Usage Record recommendation;[10]

3. The records should be updated to the current National Grid Service (NGS) accounting facilities.

A proof of concept of the accounting system has been implemented in Perl. It consists of three modules and one database:

- Aggregator module, aggregates the Eucalyptus log files by filtering the available log entries storing only the relevant data. Eucalyptus destructively rotates its own log files based on the space they use so that only a precise amount of log files are retained at any time. The aggregator module guarantees that the original log entries are saved and made available for late use and/or review.

- Parser module, parses the aggregated log files in order to populate the Usage Record Database (URDb).

- Usage Record Database (URDb), stores all the users and usage data. It is built so that each registered Eucalyptus user can be associated to her/his VM, EBS and Walrus usage. The list of registered users is built as needed and no external input is required.

- Interface module, polls the URDb compiling the Usage Records. The format of the Usage Record is defined in this module following what is required by the adopted standard.

This implementation satisfies the three given requirements:

---

[8] http://www.eoveri.com/
[9] http://www.eoveri.com/products/zeel-i.html
10 http://*www.ogf.org/documents/GFD.98.pdf*

- Uses only the log files produced by a default Eucalyptus installation so that it does not require any change to the Eucalyptus source code.

- The Usage Records produced by the interface module can be consistent with the OGF Usage Record recommendations.

- The records produced by the interface module can be uploaded to the served developed and adopted by NGS in order to manage their resource records.[11]

## *Conclusions*

The installation process of Eucalyptus is well documented and of straight implementation for system administrator with robust knowledge of the Linux operating system and of network infrastructures. The major problems derive from the lack of robustness of Eucalyptus, especially for what concern the CLC stability. Monitoring and investigation of misbehaviour can be complex and require dedicated effort/resources. During six months of testing the following recurrent problems have been noted:

- Eucalyptus 1.6.3. Corruption of the public IP database. Clean restart of CC/SC required;

- Eucalyptus 16.3/2.0/2.0.2: quiet failure of the CLC. Restart needed.

- Eucalyptus 2.0/2.0.2: quiet failure on the e-scsi EBS attachment + corruption of the EBS volumes. Shutdown of all the running VMs + CC CLEAN=1/SC restart needed. Loss of EBS data.

On the base of this testing and experience, Eucalyptus 2.* is not ready for delivering a (pre)production service to the research community. The same should be concluded for UEC, especially when considering the issues introduced by the modification implemented on the standard distribution of Eucalyptus.

---

11 http://vidar.ngs.manchester.ac.uk/rus.html
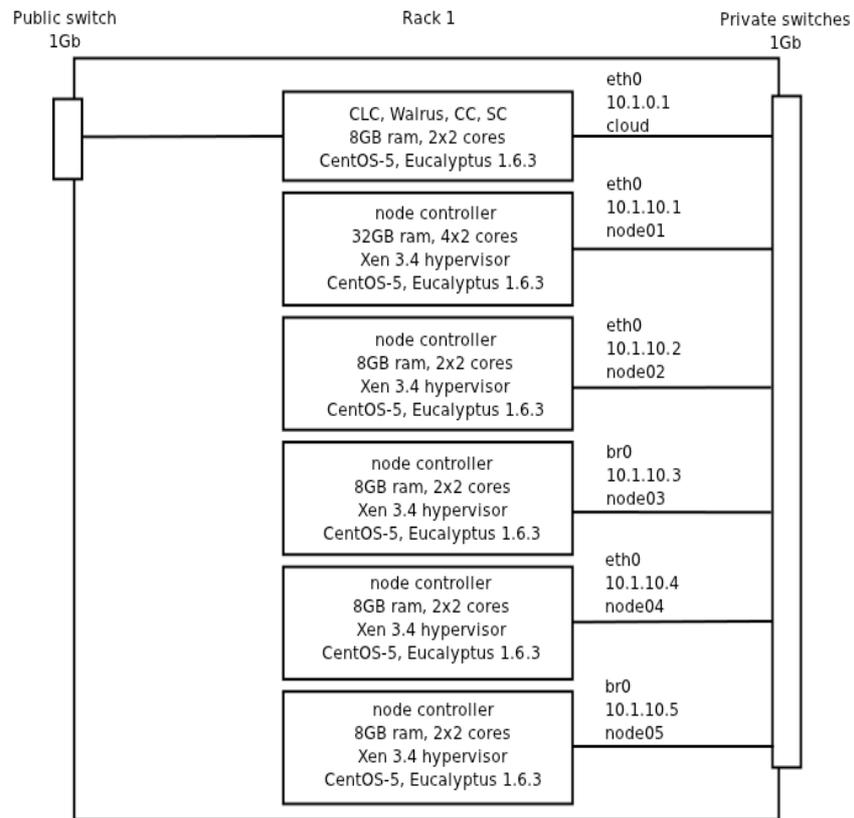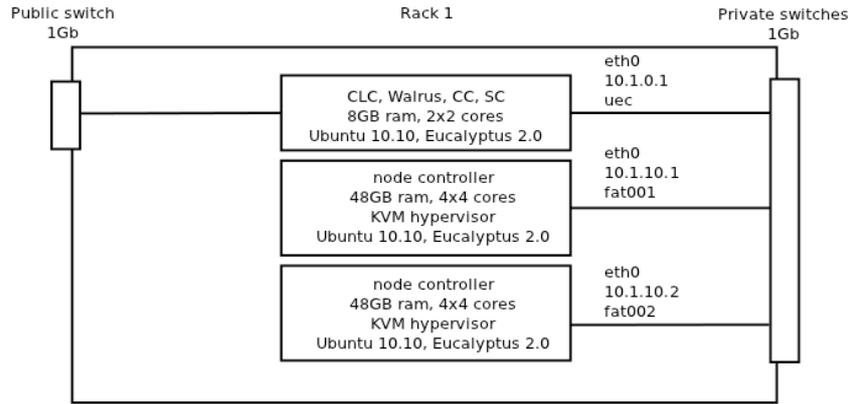
# Figures



*Figure 1: Eucalyptus infrastructure based on CentOS-5*

*Figure 2: Eucalyputus 2.0.2 infrastructure based on Ubuntu 10.10*