THE EUCALYPTUS
OPEN-SOURCE
PRIVATE CLOUD

By Yohan Wadia
Eucalyptus is a Linux-based open-source software architecture that implements efficiency-enhancing private and hybrid clouds within an enterprise’s existing IT infrastructure.

Eucalyptus is an acronym for “Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems.”
A Eucalyptus private cloud is deployed across an enterprise’s “on premise” data center infrastructure and is accessed by users over enterprise intranet. Thus, sensitive data remains entirely secure from external intrusion behind the enterprise firewall.

Initially developed to support the high performance computing (HPC) research of Professor Rich Wolski’s research group at the University of California, Santa Barbara, Eucalyptus is engineered according to design principles that ensure compatibility with existing Linux-based data center installations. Eucalyptus can be deployed without modification on all major Linux OS distributions, including Ubuntu, RHEL, Centos, and Debian. And Ubuntu distributions now include the Eucalyptus software core as the key component of the Ubuntu Enterprise Cloud.

TECHNOLOGY

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

The external interface to Eucalyptus can also be leveraged to become compatible with multiple public clouds (Amazon EC2, Sun Cloud, etc.).

EUCALYPTUS COMPONENTS

Each Eucalyptus service component exposes a well-defined language agnostic API in the form of a WSDL document containing both the operations that the service can perform and the input/output data structures. Inter-service authentication is handled via standard WS-Security mechanisms. There are five high-level components, each with its own Web-service interface, that comprise a Eucalyptus installation (Fig a). A brief description of the components within the Eucalyptus system follows.

![Eucalyptus Components Architecture - An Overview](image-url)
CLOUD CONTROLLER

Cloud Controller (CLC) is the entry-point into the cloud for administrators, developers, project managers, and end-users. The CLC is responsible for querying the node managers for information about resources, making high level scheduling decisions, and implementing them by making requests to cluster controllers. The CLC, as shown in Figure 1, is also the interface to the management platform. In essence, the CLC is responsible for exposing and managing the underlying virtualized resources (servers, network, and storage) via a well-defined industry standard API (Amazon EC2) and a Web-based user interface.

Functions:

1. Monitor the availability of resources on various components of the cloud infrastructure, including hypervisor nodes that are used to actually provision the instances and the cluster controllers that manage the hypervisor nodes.
2. Resource arbitration – deciding which clusters will be used for provisioning the instances.
3. Monitoring the running instances.

In short, CLC has a comprehensive knowledge of the availability and usage of resources in the cloud and the state of the cloud.

CLUSTER CONTROLLER

Cluster Controller (CC) generally executes on a cluster front-end machine or any machine that has network connectivity to both the nodes running NCs and to the machine running the CLC. CCs gather information about a set of VMs and schedules VM execution on specific NCs. The CC also manages the virtual instance network and participates in the enforcement of SLAs as directed by the CLC. All nodes served by a single CC must be in the same broadcast domain (Ethernet).

Functions:

1. To receive requests from CLC to deploy instances.
2. To decide which NCs to use for deploying the instances.
3. To control the virtual network available to the instances.
4. To collect information about the NCs registered with it and report it to the CLC.

NODE CONTROLLER

Node Controller (NC) is executed on every node that is designated for hosting VM instances. A UEC node is a VT-enabled server capable of running KVM as the hypervisor. UEC automatically installs KVM when the user chooses to install the UEC node. The VMs running on the hypervisor and controlled by UEC are called instances. Eucalyptus supports other hypervisors like Xen apart from KVM, but Canonical has chosen KVM as the preferred hypervisor for UEC.

The NC runs on each node and controls the life cycle of instances running on the node. The NC interacts with the OS and the hypervisor running on the node on one side and the CC on the other side.

NC queries the operating system running on the node to discover the node’s physical resources – the number of cores, the size of memory, and the available disk space. It also learns about the state of VM instances running on the node and propagates this data up to the CC.

Functions:

1. Collection of data related to the resource availability and utilization on the node and reporting the data to CC.
2. Instance life cycle management.

STORAGE CONTROLLER

Storage Controller (SC) implements block-accessed network storage (e.g., Amazon Elastic Block Storage -- EBS) and is capable of interfacing with various storage systems (NFS, iSCSI, etc.). An elastic block store is a Linux block device that can be attached to a virtual machine but sends disk traffic across the locally attached network to a remote storage location. An EBS volume cannot be shared across instances but does allow a snapshot to be created and stored in a central storage system such as Walrus, the Eucalyptus storage service.

Functions:

1. Creation of persistent EBS devices.
2. Providing the block storage over AoE or iSCSI protocol to the instances.
3. Allowing creation of snapshots of volumes.
Walrus (put/get storage) allows users to store persistent data, organized as eventually-consistent buckets and objects. It allows users to create, delete, list buckets, put, get, and delete objects, and set access control policies. Walrus is interface compatible with Amazon’s S3, and supports the Amazon Machine Image (AMI) image-management interface, thus providing a mechanism for storing and accessing both the virtual machine images and user data. Using Walrus, users can store persistent data, which is organized as buckets and objects. WS3 is a file-level storage system, as compared to the block-level storage system of Storage Controller.

For using Walrus to manage Eucalyptus VM images, you can use Amazon’s tools to store/register/delete them from Walrus. Other third-party tools can also be used to interact with Walrus directly.

**THIRD-PARTY TOOLS FOR INTERACTING WITH WALRUS**

1. S3curl: a command line tool that is a wrapper around curl.
   http://open.eucalyptus.com/wiki/s3curl
2. S3cmd: a tool that allows command line access to storage that supports the S3 API.
   http://open.eucalyptus.com/wiki/s3cmd
3. S3fs: a tool that allows users to access S3 buckets as local directories.
   http://open.eucalyptus.com/wiki/s3fs

**MANAGEMENT PLATFORM**

Management Platform provides an interface to various Eucalyptus services and modules. These features can include VM management, storage management, user/group management, accounting, monitoring, SLA definition and enforcement, cloud-bursting, provisioning, etc.

**EUCA2OOL**

Euca2ools are command-line tools for interacting with Web services that export a REST/Query-based API compatible with Amazon EC2 and S3 services. The tools can be used with both Amazon’s services and with installations of the Eucalyptus open-source cloud-computing infrastructure. The tools were inspired by command-line tools distributed by Amazon (api-tools and ami-tools) and largely accept the same options and environment variables. However, these tools were implemented from scratch in Python, relying on the Boto library and M2Crypto toolkit.

**Features:**
1. Query of availability zones (i.e., clusters in Eucalyptus).
2. SSH key management (add, list, delete).
3. VM management (start, list, stop, reboot, get console output).
4. Security group management.
5. Volume and snapshot management (attach, list, detach, create, bundle, delete).
7. IP address management (allocate, associate, list, release).

**KEY BENEFITS**

- Build and manage self-service heterogeneous on-premise IaaS clouds using either existing infrastructure or dedicated compute, network and storage resources.
- Support high-availability IaaS for the most demanding cloud deployments.
- Gain precise control of private cloud resources via enterprise-ready user and group identity management along with resource quotas.
- Pool dynamic resources with built-in elasticity, allowing organizations to scale up and down virtual compute, network and storage resources.
- Integrate robust storage, enabling IT to easily connect and manage existing storage systems from within Eucalyptus clouds.
- Build hybrid clouds between on-premise Eucalyptus clouds and AWS and AWS-compatible public clouds.
- Run Eucalyptus or Amazon Machine Images as virtual cloud instances on Eucalyptus and AWS-compatible clouds.
- Leverage vibrant AWS ecosystem and management tools to manage Eucalyptus IaaS clouds.

**REFERENCES**

- http://open.eucalyptus.com/
- http://www.eucalyptus.com/resources/cloud-myths-dispelled