

Implementación de un prototipo de Cloud Computing de modelo privado para ofrecer Infraestructura como Servicio (IaaS)

Elisa K. Mena, Ana C. Guerrero e Iván M. Bernal
Escuela Politécnica Nacional

Resumen— El artículo presenta el proceso realizado para la implementación de un prototipo de Computación en Nube (*Cloud Computing*), utilizando el *middleware* de código abierto *Eucalyptus*, el cual permite desplegar modelos de *Cloud Computing* privado e híbrido. Para la implementación del prototipo se describen los pasos de configuración fundamentales, se utilizan tres computadores físicos correspondientes a un *front-end* y dos nodos. Adicionalmente, se explica el manejo de la Nube a través de *Elasticfox* y las pruebas de funcionamiento del prototipo por parte del usuario.

Palabras clave— *Cloud Computing*, *Eucalyptus*, *Infraestructura como Servicio*, *Virtualización*, *Elasticfox*, *Ubuntu Enterprise Cloud*.

I. INTRODUCCIÓN

EN los últimos años, el desarrollo de las Tecnologías de Información (TI) ha dado paso a que los datos, la música, el video, etc. se encuentren almacenados y distribuidos a través del Internet, de tal manera que dichos recursos puedan ser accesibles en cualquier momento y prácticamente, desde cualquier lugar.

Los sistemas operativos han sido siempre el corazón de los servidores y de los computadores personales, y a lo largo del desarrollo de la tecnología, los nuevos elementos de software han ido adaptándose a los componentes de hardware para contar con un rendimiento cada vez más alto en el uso de aplicaciones. Sin embargo, los computadores no son utilizados todo el tiempo, a menos que se trate de brindar un servicio específico que demande una alta disponibilidad (24x7), e incluso en este caso realizan tareas que no requieren de toda su capacidad disponible.

Por otro lado, existen servidores antiguos, con sistemas operativos antiguos, y que ejecutan aplicaciones antiguas. Para migrar a las nuevas aplicaciones que exige el desarrollo de las TI es necesario seguir un largo proceso. “*La mayoría de departamentos de TI se ven obligados a dedicar una buena parte de su tiempo a la tarea de implementar, mantener y actualizar proyectos que, con demasiada frecuencia, no suponen un valor añadido en el balance final de la empresa*”

[1] y que en un determinado momento se verán obsoletos.

Diversos estudios muestran que aproximadamente el 75% del presupuesto de TI de una compañía se gasta en mantenimiento de la estructura existente, mientras que tan sólo el 25% se dedica a innovación. [2]

Es así que, de acuerdo a la necesidad de acelerar los procesos y además de aprovechar al máximo los recursos en los que se ha invertido, nace lo que se conoce como la virtualización.

La virtualización es un método que se utiliza para ejecutar sistemas operativos, múltiples e independientes en una sola máquina física. [3]

La tecnología de virtualización aprovecha los recursos de hardware subutilizados, y *Cloud Computing* la toma como base para poder ofrecer servicios de infraestructura, plataforma y software, a clientes que solamente requieren de estos servicios bajo demanda, permitiendo al usuario pagar al proveedor de este servicio, únicamente por lo que consume.

Es así como *Cloud Computing* se incorpora a las TI, minimizando el tiempo empleado en actividades de implementación y centrándose en actividades más estratégicas, que tienen un mayor impacto en los procesos comerciales.

Cloud Computing da al usuario acceso a los recursos de cómputo (máquinas, almacenamiento, sistemas operativos, entornos de desarrollo de aplicaciones, programas) a través de una red (Internet o Intranet) valiéndose de servicios web. “*La organización física de los equipos que ofrecen estos recursos, puede hallarse ya sea en la habitación contigua o distribuida alrededor del mundo, ya que los recursos aparecen ante el usuario como si se encontraran alojados en la Nube*”. [4]

“*El término Nube se refiere a la organización de infraestructura física, la cual es transparente para el usuario final*”. [4]

La virtualización y *Cloud Computing* son tecnologías que pueden adoptarse juntas o cada una de forma individual. La virtualización es parte fundamental de *Cloud Computing* ya que gracias a ella es posible disponer de los servicios que se ofrecen.

“*En la actualidad, la virtualización y Cloud Computing están siendo aceptados y adoptados por un número creciente de usuarios*”. [5]

II. DEFINICIÓN DE CLOUD COMPUTING

Cloud Computing es un modelo de computación que permite ofrecer, de manera escalable, cualquier tipo de recurso virtualizado como un servicio a través de una red.

Cloud Computing puede ofrecer servicios utilizando Centros de Datos (*data centers*) propios dentro de una empresa, o se pueden utilizar los servicios ofrecidos por la Nube de un proveedor que es quien tiene el control de los elementos físicos de su infraestructura.

“*Cloud Computing es la entrega de recursos de cómputo a través de una interfaz web (SOAP¹ o REST²) según las necesidades*”. [5]

III. MODELOS DE INFRAESTRUCTURA EN CLOUD COMPUTING

A. Cloud Computing de modelo público

El proveedor de los servicios de *Cloud Computing* es dueño de la infraestructura física y pone a disposición del cliente los servicios de la Nube a través del Internet; ésta es su característica esencial pues es lo que permite que el usuario pueda acceder a dichos servicios en cualquier momento y lugar.

B. Cloud Computing de modelo privado

Es una emulación de una Nube pública, pero en una red privada, ya que ofrece los mismos servicios que una Nube pública con la ventaja de que el usuario cuenta con sus propios recursos, lo que le permite tener el control de seguridad y calidad de servicio sobre ellos.

C. Cloud Computing de modelo híbrido

Las Nubes híbridas combinan los modelos público y privado. Este modelo tiene la ventaja de contar con los beneficios de ambos modelos, lo cual permite aumentar la capacidad de una Nube privada con los recursos de una Nube pública para poder mantener niveles de servicio adecuados, frente a rápidas fluctuaciones de carga de trabajo.

IV. MODELOS DE SERVICIOS DE CLOUD COMPUTING

A. Infraestructura como Servicio (IaaS)

En este servicio, la capacidad suministrada a los clientes es el abastecimiento de procesamiento, espacio de almacenamiento, equipos de red y otros recursos computacionales importantes para que los clientes puedan desplegar y ejecutar software de forma arbitraria, lo cual puede incluir sistemas operativos y aplicaciones.

La infraestructura se brinda, normalmente, mediante una plataforma de virtualización.

B. Plataforma como Servicio (PaaS)

Este servicio brinda a los clientes la capacidad de desplegar sus aplicaciones en la infraestructura de la Nube, utilizando diferentes lenguajes y herramientas de programación que el proveedor del servicio soporte.

Los clientes no gestionan ni controlan la infraestructura de la Nube, pero tienen el control sobre las aplicaciones desplegadas y su configuración.

C. Software como Servicio (SaaS)

La facilidad proporcionada con este servicio a los clientes es la utilización de aplicaciones del proveedor, que se ejecutan en la infraestructura de la Nube.

Adicionalmente, el usuario no se preocupa donde está instalado el software, qué tipo de sistema operativo utiliza o el lenguaje en el que cada aplicación está escrita.

V. MODELO EMPLEADO

El prototipo de *Cloud Computing* de modelo privado se construye en base a tres máquinas físicas, un *front-end* y dos nodos (Fig. 1).

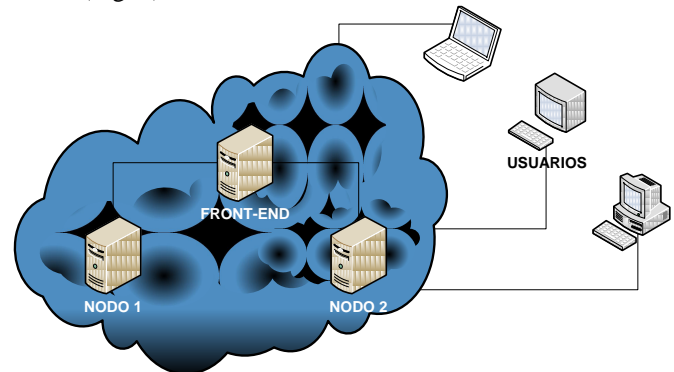


Fig. 1. Diagrama de conexiones de las máquinas físicas.

Este prototipo se implementa utilizando el *middleware Eucalyptus (Elastic Utility Computing for Linking your Programs to Useful Systems)*, que es un software de código abierto, para ofrecer Infraestructura como Servicio (IaaS) dentro de un modelo de *Cloud Computing* privado o híbrido.

La arquitectura de *Eucalyptus* es simple, flexible y modular, con un diseño jerárquico, como se indica en la Fig. 2.

El *Cloud Controller* (CLC) virtualiza los recursos subyacentes de los componentes del *front-end* y permite la comunicación de la Nube con el usuario. El *Cluster³ Controller* (CC) recopila información de estado de los *Node Controllers*, gestiona la red y las peticiones de ejecución de las máquinas virtuales. Los *Node Controllers* (NCs) son los equipos en los cuales se ejecutan las máquinas virtuales. El *Storage Controller* (SC) ofrece un servicio de almacenamiento en bloque similar a *Amazon Elastic Block Store (EBS)*, mientras que el sistema de almacenamiento *Walrus*, permite almacenar datos persistentes, organizados como *buckets⁴* y

¹ SOAP (*Simple Object Access Protocol*) es una especificación de protocolo de intercambio de información estructurada en la aplicación de Servicios Web en redes informáticas. [6]

² REST (*Representational State Transfer*) es una técnica de arquitectura de software para sistemas hipermedia distribuidos como la *World Wide Web*. [7]

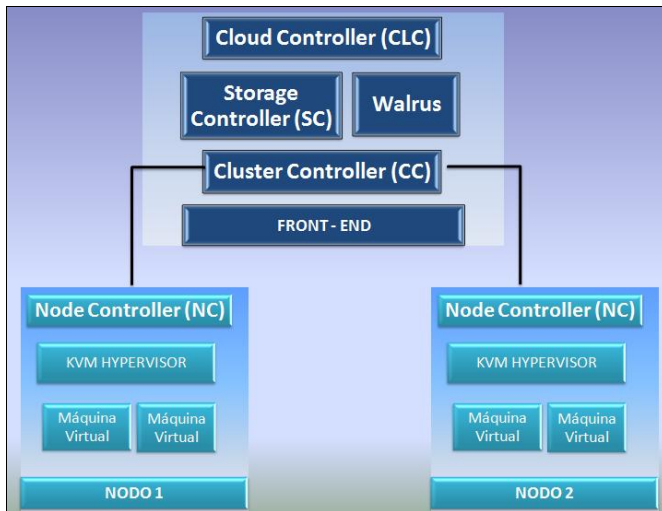
³ Cluster es un conjunto de computadores hacia los cuales se distribuye la carga (peticiones) que recibe el *Cluster Controller*.

⁴ Bucket es un contenedor que almacena objetos que se pueden agregar o quitar con los comandos *put* y *get*.

objetos consistentes, *Walrus* se extiende por la Nube entera y es similar a *Amazon Simple Storage Service (S3)*. [8]

A. Especificaciones técnicas del prototipo

En la Tabla I y Tabla II se especifican los requerimientos de hardware para el *front-end* y cada nodo.



Fuente: Basada en [9]

Fig. 2. Componentes de *Eucalyptus* instalados en cada equipo físico del prototipo de Nube implementado.

TABLA I
REQUERIMIENTOS DE HARDWARE PARA EL FRONT-END

Hardware	Mínimo	Sugerido	Utilizado	Notas
CPU (GHz)	1	2 x 2	3	Para front-ends donde corren varios componentes de <i>Eucalyptus</i> , es de gran ayuda utilizar un procesador, por lo menos, del tipo Dual Core.
Memoria RAM (GB)	2	4	4	La interfaz web basada en Java, necesita grandes cantidades de memoria.
Espacio en disco (GB)	40	200	200	40GB es sólo el espacio necesario para que una única imagen, caché, etc., <i>Eucalyptus</i> no trabaja sin espacio en disco.
Disco (rpm)	5400 IDE	7200 SATA	7200 SATA	Discos lento fundarán, pero una instancia tardará mucho más tiempo en iniciar.
Red (Mbps)	100	1000	100	Las imágenes de MV constan de cientos de MB, y se deben copiar a través de la red a los nodos.

Fuente: [10]

En la Tabla III se muestran las direcciones IP asignadas dentro de la red de la Escuela Politécnica Nacional correspondientes a la VLAN 172.31.9.X, y la red privada de los componentes de *Eucalyptus* correspondientes a la red 10.0.0.X.

TABLA II
REQUERIMIENTOS DE HARDWARE PARA LOS NODOS

Hardware	Mínimo	Sugerido	Utilizado N (1,2)	Notas
CPU	Extensión de Virtualización	Extensión de Virtualización, Multicore de 64 bits	Extensión de Virtualización, Cuad core de 64 bits	Instancias de 64-bits se pueden ejecutar tanto en i386 como en amd64, por defecto, <i>Eucalyptus</i> sólo ejecutará una VM por núcleo de CPU en un nodo.
Memoria RAM (GB)	1	4	4	Memoria adicional, significará <i>guests</i> más grandes.
Espacio en disco (GB)	40	100	100	
Disco (rpm)	5400 IDE	7200 SATA o SCSI	7200 SATA	Los nodos de <i>Eucalyptus</i> utilizan el disco intensivamente; operaciones de E/S tendrán que esperar probablemente para mejorar el rendimiento en cuellos de botella.
Red (Mbps)	100	1000	100	Las imágenes de MV constan de cientos de MB, y se deben copiar a través de la red a los nodos.

Fuente: [10]

TABLA III
DIRECCIONES IP ASIGNADAS A CADA MÁQUINA DENTRO DEL SISTEMA

Máquina/Interfaz	eth0	eth1	eth0:priv	eth0:pub
Front-End	10.0.0.1 (asignada después de la instalación)	172.31.9.21/192.168.1.57.170 (asignada durante la instalación)	192.168.1.1 (asignada por <i>Eucalyptus</i>)	172.31.9.22 (asignada por <i>Eucalyptus</i>)
Nodo1 (br0)	10.0.0.2 (asignada al bridge que utiliza eth0)	X		
Nodo2 (br0)	10.0.0.3 (asignada al bridge que utiliza eth0)	X		
Administrador	172.31.9.X	X		
Usuario	172.31.9.X	X		
Máquinas Virtuales	192.168.1.X	172.31.9.22 - 172.31.9.23		

B. Diseño de la topología de red y ubicación de los componentes físicos

En la Fig. 3 se pueden observar las redes a las que pertenecen los componentes de la Nube, los usuarios y las máquinas virtuales. Se tiene la nube Internet desde la cual se puede acceder a los servicios de la Nube implementada con *Eucalyptus*. Tanto el administrador como el usuario tienen acceso con diferentes permisos.

El círculo, en la parte superior de la Fig. 3 (POLIRED), representa a la red privada de la Escuela Politécnica Nacional a través de la cual el CLC puede salir a Internet mediante NAT en la IP privada: 172.31.9.21/24 a una IP pública.

El círculo en la parte inferior de la Fig. 3 (RED PRIVADA EUCALYPTUS), representa a la red privada de la Nube basada en *Eucalyptus* (10.0.0.X/24), a través de la cual sus tres componentes (*front-end* y dos nodos) se comunican.

Las máquinas virtuales (MVs) se encuentran en la parte inferior de la Fig. 3, éstas se ejecutan en otra red privada (RED PRIVADA MVs) a través de la cual se pueden comunicar entre ellas y entre máquinas de otros *clusters* de la misma Nube; las máquinas virtuales pueden tener dos interfaces de red una pública (pub) y una privada (priv).

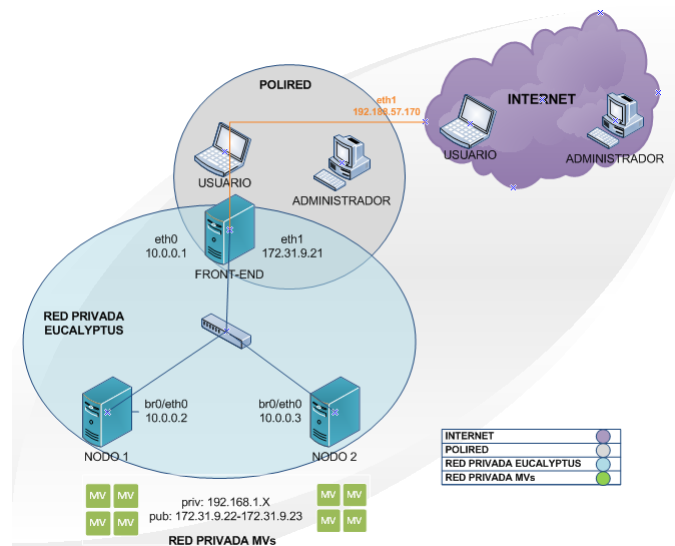


Fig. 3. Configuración de los componentes de Eucalyptus.

VI. IMPLEMENTACIÓN DEL PROTOTIPO

A. Instalación y configuración de los componentes de UEC

Ubuntu Enterprise Cloud (UEC) es un conjunto de aplicaciones, incluidas en *Ubuntu Server Edition* junto con un gran número de otras aplicaciones de código abierto; *Eucalyptus* es su componente principal. Facilita la instalación y configuración de los componentes dentro del entorno de *Cloud Computing*.

Eucalyptus permite crear un entorno propio de *Cloud Computing* con el objetivo de maximizar los recursos informáticos (*clusters*, *workstations*, varios servidores y máquinas de escritorio). [11]

Después de iniciar la instalación, el instalador detectará si actualmente existe algún componente de *Eucalyptus*, de esta forma se instala el *front-end* y los nodos según corresponda, se configurarán sus *hostnames* y se instalan el *Cloud/Cluster/Storage/Walrus Controller* en el *front-end*, y el *Node Controller* en los nodos, se crearán las cuentas de usuario para el *front-end* y los nodos, se configurará el servidor de correo electrónico *Postfix* y se especificará el rango de direcciones IP públicas a utilizarse en la implementación.

En los nodos es muy importante utilizar hardware que soporte extensión de virtualización (Intel-VT⁵ o AMD-V⁶) ya que son los que van a ejecutar las máquinas virtuales y el uso del *hypervisor* KVM⁷, el cual viene instalado por defecto en *Ubuntu Server Edition*.

B. Configuración de los componentes de Eucalyptus

Después de la instalación de UEC, en el *front-end* se configura una interfaz de red para que tenga comunicación con la red privada de la EPN, y también se configura una interfaz para que el *front-end* se comunique con cada uno los nodos. Lo más importante en este punto es recordar que en los nodos se configura un *bridge*⁸ que utiliza como su interfaz primaria a la interfaz *eth0*.

Para obtener la comunicación entre los componentes de la Nube, se registran el *Walrus*, *Cluster Controller*, *Storage Controller* y *Node Controller* con el *Cloud Controller* para que este último sepa la ubicación de cada uno de los otros componentes dentro de la red.

Para interactuar con el CLC se deben descargar las credenciales de acceso de un determinado usuario, esto se lo puede hacer ejecutando un script desde la línea de comandos (Fig. 4) o utilizando la interfaz web con la URL: <https://cloud.epn.edu.ec:8443>.

```
#credentials script
rm -rf ~/.euca/credenciales.zip
rm -rf ~/.euca/*.pem
rm -rf ~/.euca/eucarc
mkdir -p ~/.euca
chmod 700 ~/.euca
cd ~/.euca
euca_conf --get-credentials credenciales.zip
unzip credenciales.zip
ln -s ~/.euca/eucarc ~/.eucarc
```

Fig. 4. Archivo *credentials.sh*.

Una vez que las credenciales de un usuario fueron descargadas, se crea el archivo “*eucarc*” donde se almacenan las variables de entorno que *Eucalyptus* utiliza; es decir, se almacena información acerca de las URLs para la conexión de la Nube con el servicio de almacenamiento, información sobre las claves de acceso, clave secreta e identificación de un determinado usuario.

Se deben instalar las *euca2ools*⁹, para interactuar con los servicios web de *Amazon* y *Eucalyptus*, para consultar las zonas de disponibilidad¹⁰ y administrar llaves SSH, máquinas

⁵ Intel-VT es conjunto de mejoras de hardware para plataformas de servidores y clientes Intel que proporcionan soluciones de virtualización basadas en software. [12]

⁶ AMD-V incluye un conjunto de extensiones de hardware a la arquitectura de sistema x86 para aumentar el rendimiento, fiabilidad y seguridad de los entornos de virtualización actuales y futuros. [13], [14]

⁷ KVM (*Kernel-based Virtual Machine*) es software de código abierto, que representa una solución de virtualización completa para arquitecturas de hardware x86 que contengan extensión de virtualización (Intel-VT o AMD V).

⁸ *Bridge* interconecta segmentos de red haciendo el cambio de tramas entre las redes de acuerdo con una tabla de direcciones. [15]

⁹ *euca2ools* son herramientas de línea de comandos para la interacción de un usuario con los servicios web de una Nube (*Eucalyptus* o *Amazon*).

¹⁰ Zonas de disponibilidad (*availability zones*) representan una colección lógica de máquinas que tienen varios NCs y un solo CC. [4]

virtuales, grupos de seguridad, volúmenes lógicos, imágenes, direcciones IP, etc.

VII. PRUEBAS DE FUNCIONAMIENTO

A. Inserción de imágenes en Eucalyptus

Como se muestra en la Fig. 5, una de las opciones para insertar imágenes en el sistema, es utilizar la interfaz web donde se pueden ver las imágenes disponibles.

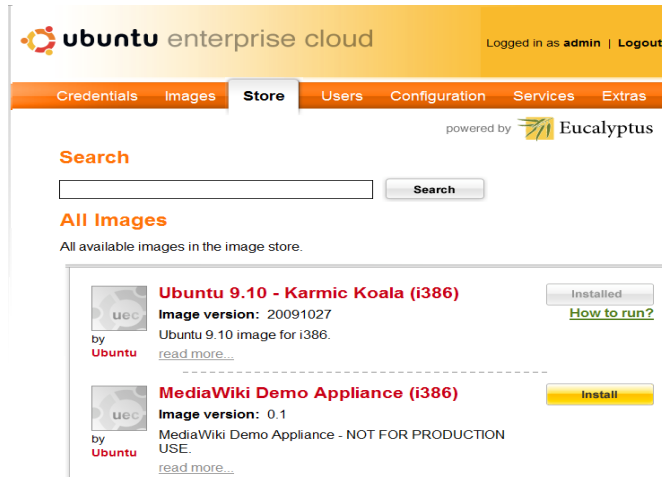


Fig. 5. Imágenes disponibles en Store de UEC.

Alternativamente se pueden insertar imágenes de diferentes sistemas operativos compatibles con *Eucalyptus* a través de las herramientas de línea de comandos, para lo cual primero se debe descargar el archivo que contiene la imagen que se desea instalar.

El comando *euca-bundle-image* permite “preparar” la imagen, separándola en múltiples partes y generando un archivo *manifes.xml*, que contiene la *metadata*¹¹ acerca de la imagen; estas partes son almacenadas en el directorio */tmp* hasta que sean cargadas en el *Walrus* a través del comando *euca-upload-bundle*. El comando *euca-register* permite registrar el *kernel*¹² insertado con *Eucalyptus*.

Como ejemplo se va a utilizar una imagen de Centos 5.3; en la Fig. 6 se muestran los comandos para la inserción del *kernel* de la imagen; en la Fig. 7 se añade la imagen del sistema de archivos *root* al *Walrus* y en la Fig. 8 se realiza la inserción de la *ramdisk*¹³ de la imagen.

```
euca-bundle-image -i ~/euca-centos-5.3-i386/kvm-kernel/vmlinuz-2.6.28-11-server --kernel true
euca-upload-bundle -b centos-kernel-bucket -m /tmp/vmlinuz-2.6.28-11-server.manifest.xml
euca-register centos-kernel-bucket/vmlinuz-2.6.28-11-server.manifest.xml
```

Fig. 6. Imagen Centos, inserción del *kernel*.

```
euca-bundle-image -i ~/euca-centos-5.3-i386/kvm-kernel/initrd.img-2.6.28-11-server --ramdisk true
euca-upload-bundle -b centos-ramdisk-bucket -m /tmp/initrd.img-2.6.28-11-server.manifest.xml
euca-register centos-ramdisk-bucket/initrd.img-2.6.28-11-server.manifest.xml
```

Fig. 7. Imagen Centos, inserción de la imagen al sistema de archivos.

```
euca-bundle-image -i ~/euca-centos-5.3-i386/centos.5-3.x86.img --kernel eki-91EB1792 --ramdisk eri-F87118D2
euca-upload-bundle -b centos-image-bucket -m /tmp/centos.5-3.x86.img.manifest.xml
euca-register centos-image-bucket/centos.5-3.x86.img.manifest.xml
```

Fig. 8. Imagen Centos, inserción de la *ramdisk*.

B. Ejecución de instancias (máquinas virtuales)

Antes de ejecutar una instancia de una imagen, se deben crear primero las *keypairs*, que son claves para la autenticación de usuarios. (Fig. 9)

```
mkdir -p -m 700 ~/.euca
touch ~/.euca/llavesAdmin.priv
chmod 0600 ~/.euca/ llavesAdmin.priv
euca-add-keypair llavesAdmin > ~/.euca/ llavesAdmin.priv
```

Fig. 9. Creación de *keypairs*.

Para ejecutar la máquina virtual se utiliza el comando *euca-run-instances* con el identificador *emi*¹⁴ correspondiente a la imagen a utilizarse, la opción *-k* especifica la llave pública del usuario, y la opción *-t* especifica el tamaño de máquina virtual deseado. (Fig. 10)

```
cd ~/.euca
euca-run-instances emi-E62F151C -k llavesAdmin -t c1.medium
```

Fig. 10. Ejecución de una máquina virtual.

Finalmente, se puede obtener una descripción completa de la instancia. En la Fig.11, se puede ver el usuario y grupo al cual le pertenece, un identificador de la instancia *-i*, el estado de la máquina virtual, la llave pública *llavesAdmin*, el tamaño de la máquina virtual, la fecha de creación, el nombre del *cluster* en donde se está ejecutando, el identificador *eki*¹⁵ y *eri*¹⁶.

¹¹*Metadata* son datos estructurados que describen las características de un recurso. [16]

¹²*Kernel* es el componente central de un sistema operativo, facilita la funcionalidad de las aplicaciones, representa el puente entre los procesos de las aplicaciones y el procesamiento de datos subyacente, y maneja funciones que se realizan a nivel del hardware. [17]

¹³*Ramdisk* es una unidad virtual creada en la RAM, que se utiliza como si fuera una unidad de disco.

¹⁴ *emi* (*Eucalyptus Machine Image*). [13]

¹⁵ *eki* (*Eucalyptus Kernel Image*). [13]

¹⁶ *eri* (*Eucalyptus Ramdisk Image*). [13]

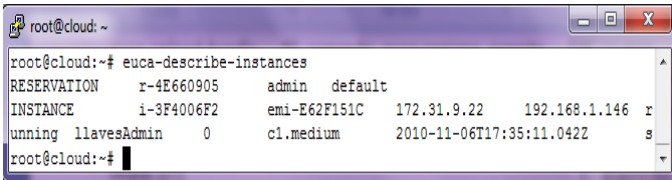


Fig. 11. Descripción de una instancia.

C. Elasticfox para la administración de la Nube

Elasticfox es un plug-in¹⁷ del navegador web Mozilla Firefox que permite administrar, a través de un entorno gráfico, los recursos de una Nube privada como Eucalyptus. [18]

Una vez que *Elasticfox* se ha inicializado, se deben configurar obligatoriamente las credenciales obtenidas de una cuenta de usuario de la Nube privada (Fig. 12). También se deben configurar las regiones en las que se van a ejecutar las instancias (Fig. 13).

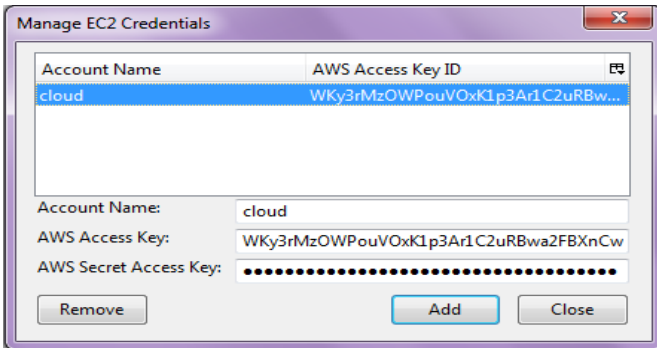


Fig. 12. Configuración de credenciales en *Elasticfox*.

Finalmente, para establecer la conexión entre *Elasticfox* y la Nube (Fig. 14) se debe establecer la variable `ec2ui.url` de éste para que al iniciarlo se conecte al CLC de la Nube, esta variable estará definida como la URL del servicio *Eucalyptus* almacenada en el archivo `eucarc`.

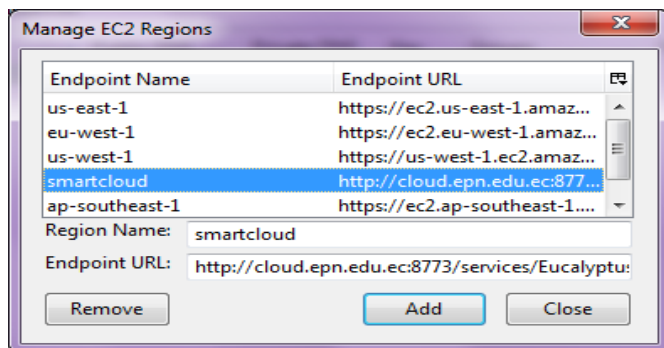


Fig. 13. Configuración de regiones en *Elasticfox*.

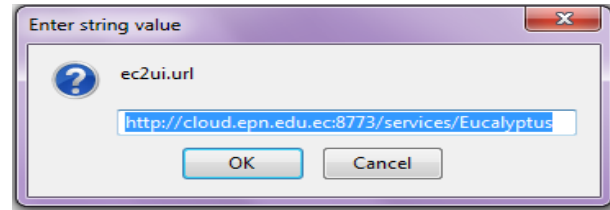


Fig. 14. `ec2ui.url`.

En la Fig. 15 se observan las imágenes con las que el usuario puede trabajar una vez que éste tiene acceso a la administración de la Nube.

Desde el navegador web, utilizando *Elasticfox*, se podrán ver y manejar todos los elementos de la Nube, incluyendo instancias creadas, imágenes que se pueden utilizar para la creación de las instancias, diferentes configuraciones de seguridad, direcciones IP elásticas¹⁸ asignadas a las diferentes cuentas de usuario, etc., de la misma manera que si se estuviera administrando la Nube a través de un cliente SSH utilizando las *euca2ools*.

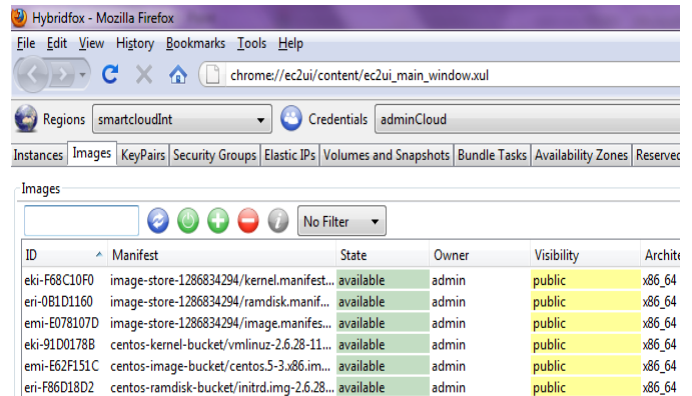


Fig. 15. Imágenes disponibles vistas desde *Elasticfox*.

Como ejemplo, se discute a continuación el manejo de volúmenes dentro del prototipo de *Cloud Computing*.

Para la creación de un volumen se escoge la opción “*Volumes and Snapshots*” y como se muestra en la Fig. 16, se definen las características del mismo.

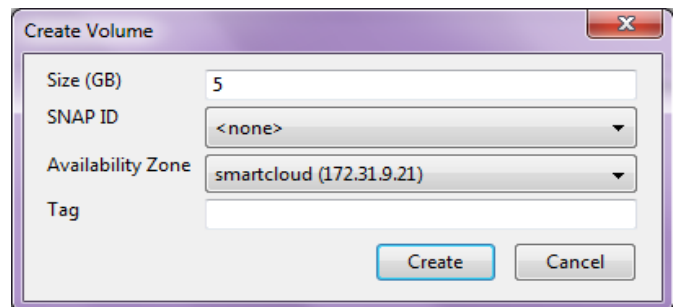


Fig. 16. Creación de un volumen en *Elasticfox*.

¹⁷*Plug-in* es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. [19]

¹⁸*Direcciones IP elásticas* son direcciones IP asignadas dinámicamente a partir de un pool de direcciones fijas. [20]

Para asociar un volumen a una instancia, ésta debe estar en ejecución, se selecciona dicha instancia y se escogen ciertas características como: identificación de la instancia, la cual está dentro de las especificaciones de la instancia en ejecución, identificación del volumen con el cual se la va a asociar, que es información que se muestra en la sección “volumes” de *Elasticfox*, y dispositivo con el que se desea asociar el volumen, cada instancia muestra el tipo de dispositivo que soporta. (Fig. 17)

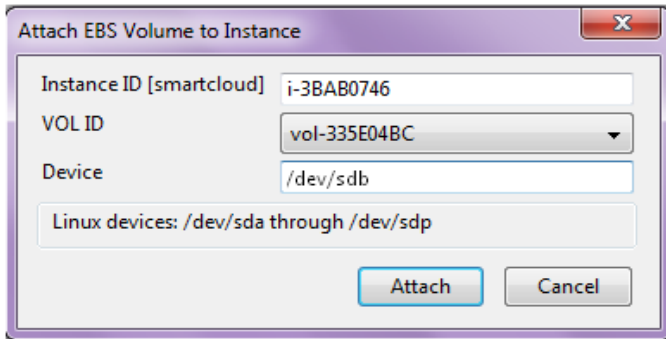


Fig. 17. Asociar volumen a una instancia.

Al momento de instalar cualquier tipo de servicio dentro del volumen, éste debe tener formato, por lo que primero el usuario debe acceder a una instancia en ejecución a través de un cliente SSH como se muestra en la Fig. 18.

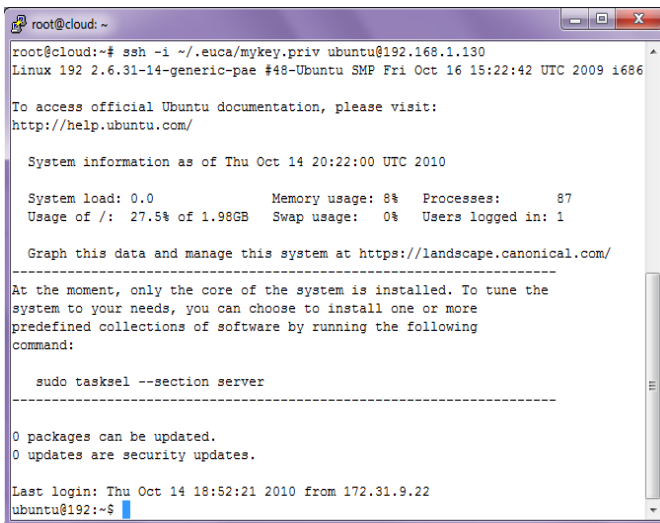


Fig. 18. Acceso a una instancia.

A continuación, con los permisos de *root*, se da formato al volumen creado, es decir se crea un sistema de archivos y un punto de montaje para el volumen; finalmente, se lo monta en dicho punto (Fig. 19).

```
mkfs.ext3 /dev/sdb1
mkdir -m 000 /vol
mount /dev/sdb1 /vol
```

Fig. 19. Dar formato a un volumen desde la línea de comandos.

Todo lo que se tenga almacenado en */vol*, incluyendo los archivos de los diferentes servicios, van a persistir más allá del tiempo de vida de la instancia en ejecución, y el usuario puede usar *snapshots*¹⁹ del volumen como copia de seguridad o réplica.

Para brindar un servicio, dentro de la instancia se debe instalar el software necesario; en este caso como ejemplo, se va a utilizar un motor de base de datos *mysql* y un servidor web (*apache*). Una vez instalados todos los paquetes requeridos, los servicios deben estar en ejecución, así también se puede ingresar y registrar una contraseña segura de *root* para *mysql*.

Es importante configurar cada uno de los servicios de tal forma que éstos se inicialicen cada vez que la máquina arranque.

Para configurar el servicio y usar el volumen, lo primero que se hace es parar el servicio de *mysql* y *apache*.

Al configurar el servicio de *mysql*, se mueven al volumen los archivos de una base de datos existente en la máquina virtual, para esto se debe considerar que *mysql* debe apuntar correctamente a los archivos de base de datos en el volumen usando el punto de montaje anteriormente descrito. (Fig. 20)

```
mkdir /vol/etc /vol/lib /vol/log
mv /etc/mysql /vol/etc
mv /var/lib/mysql /vol/lib
mv /var/log/mysql /vol/log

mkdir /etc/mysql
mkdir /var/lib/mysql

mount /etc/mysql
mount /var/lib/mysql
mount /var/log/mysql
```

Fig. 20. Mover los archivos de base de datos al volumen.

Finalmente, se debe tener el servicio de *mysql* en ejecución con todos los datos y registro de archivos almacenados de forma persistente.

Al igual que con el servicio de *mysql*, al configurar el servidor web, se copian los archivos del sitio web al volumen y se crea un link simbólico a éste. Como paso final, se inicializa el servicio de *apache*. (Fig. 21)

¹⁹*Snapshot* es una copia de un volumen creado a partir de un tiempo determinado en el tiempo. Basado en: [21], [22]

```

# Parar el servicio httpd:
service httpd stop

# Mover los archivos de http al volumen:
mv /etc/httpd /vol/httpd

# Crear un link simbólico al volumen:
ln -s /vol/httpd /etc

# También hacer esto para root:
mv /var/www /vol/www
ln -s /vol/www /var/

# Reiniciar el servicio apache:
service httpd start

```

Fig. 21. Configuración del servidor web.

Con toda la configuración anterior se tienen los servicios especificados funcionando. Utilizando la IP de la interfaz pública asignada a la máquina virtual, y a través del navegador web, se puede visualizar el sitio publicado en la máquina virtual (Fig. 22).



Fig. 22. Sitio web publicado en una instancia.

VIII. CONCLUSIONES

Cloud Computing es un nuevo modelo de computación que se está desarrollando a pasos agigantados, las grandes compañías están implementando políticas que redirigen sus modelos de negocio hacia la Nube, lo que les está permitiendo aumentar su productividad en corto tiempo y con una inversión inicial de bajo costo.

Cloud Computing ofrece como servicio virtualización y optimización de recursos, tanto en capacidad de procesamiento como almacenamiento, esto significa que las aplicaciones y datos del usuario no se encuentran dentro de sus equipos físicos, ni dependen del sistema operativo que estén ejecutando, sino que dependen de los recursos que la Nube proporcione.

La virtualización y *Cloud Computing* tienen un impacto directo y positivo sobre la investigación e inversión de los departamentos de TI, debido a que con el pasar del tiempo se está intentando reducir el presupuesto de equipos a utilizarse.

Ubuntu Enterprise Cloud y *Eucalyptus* han formado una unión estratégica la cual ha resultado en un sistema modular que permite incorporar varios servicios web, los cuales junto con la virtualización ofrecen servicios de Infraestructura. A partir de este sistema se puede escalar a capas superiores de *Cloud Computing* como Plataforma y Software, lo que implica que dicho sistema podría representar la base para la implementación de servicios en la Nube.

La implementación de este prototipo ha permitido el desarrollo de un modelo de *Cloud Computing* privado, el cual puede ser utilizado por cualquier tipo de usuario a través de la interfaz web de *Eucalyptus*, y utilizando la herramienta de gestión *Elasticfox* o un cliente SSH.

El prototipo utiliza una serie de formas de seguridad para garantizar la buena prestación de los servicios ofrecidos por la Nube, maneja claves de seguridad y certificados X.509²⁰.

Se ha comprobado que se pueden instalar imágenes de diferentes sistemas operativos basados en software libre, y compatibles con el *Eucalyptus*, para crear máquinas virtuales que se ejecuten dentro del prototipo.

Para realizar pruebas de funcionamiento del prototipo se instaló el motor de base de datos *mysql* y un servidor web (apache), servicios que están en funcionamiento dentro de una máquina virtual en ejecución asociada a un volumen.

Una de las características relevantes del prototipo es que se puede aumentar el tamaño del volumen con facilidad cuando el espacio sea insuficiente, simplemente modificando las características de dicho volumen.

REFERENCIAS

- [1] <http://www.salesforce.com/es/cloudcomputing>
"¿Qué es Cloud Computing?"
- [2] <http://eprints.ucm.es/9452/>
"Jorge Lástras, Javier Lázaro, Jonatan Mirón, Arquitecturas de red para servicios en Cloud Computing, 2007, España". PDF
- [3] http://articles.techrepublic.com.com/5100-10878_11-6074941.html
"Introduction to server virtualization"
- [4] <http://open.eucalyptus.com/EucalyptusUserGuide.v1.final.03.23.pdf>
"Eucalyptus Systems Inc., Eucalyptus Cloud Computing Platform, User Guide v 1.6"
- [5] <http://www.mkm-pi.com/mkmpi.php?article3924>
"Virtualización y Servicios distribuidos en Red (Cloud Computing): impacto en los proveedores de servicio"
- [6] <http://en.wikipedia.org/wiki/SOAP>
"SOAP"
- [7] <http://en.wikipedia.org/wiki/REST>
"REST"
- [8] http://open.eucalyptus.com/wiki/EucalyptusInstall_v1.6
"Installing Eucalyptus (1.6)"
- [9] http://cssoss.files.wordpress.com/2010/06/book_eucalyptus_beginners_guide_uec_edition1.pdf
Johnson D, Murari Kiran, Raju Murthy, Suseendran RB, Yogesh Girikumar, "Eucalyptus Beginner's Guide UEC Edition" (Ubuntu Server 10.04 - Lucid Lynx), v1.0, 25 mayo 2010.

²⁰**Certificado X.509** es un estándar UIT-T para infraestructuras de claves públicas (*Public Key Infrastructure* o PKI). X.509 especifica, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación.

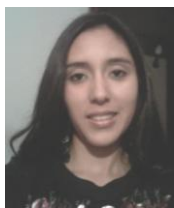
- [10] <http://en.wikipedia.org/wiki/Xen>
"Xen"
- [11] <http://110n.ubuntu.tla.ro/ubuntu-docs/jaunty/html/serverguide/es/eucalyptus.html>
"Eucalyptus"
- [12] <http://sites.amd.com/es/business/it-solutions/virtualization/Pages/amd-v.aspx>
"Tecnología AMD de Virtualización"
- [13] <http://www.amd.com/es/products/desktop/processors/phenom/Pages/AMD-phenom-processor-X3-features.aspx>
"Características básicas de arquitectura de los procesadores AMD Phenom™ X3 Triple-Core"
- [14] <https://help.ubuntu.com/community/UEC/CDInstall>
"Ubuntu Documentation, UEC CD Install"
- [15] <http://es.wikipedia.org/wiki/Ethernet>
"Ethernet"
- [16] <http://www.library.uq.edu.au/iad/ctmeta4.html>
"Chris Taylor, An Introduction to Metadata"
- [17] <http://linuxemb.wikidot.com/tesis-c2>
"Emiliano P. López, GNU/Linux embebido"
- [18] <http://blogs.techrepublic.com.com/networking/?p=1533>
"Rick Vanover, Tool for the cloud: Elasticfox"
- [19] http://es.wikipedia.org/wiki/Complemento_%28inform%C3%A1tica%29
"Complemento, plug-in"
- [20] <http://www.alegsa.com.ar/Dic/gnu%20grub.php>
"Definición de GNU /GRUB"
- [21] [http://en.wikipedia.org/wiki/Snapshot_\(computer_storage\)](http://en.wikipedia.org/wiki/Snapshot_(computer_storage))
"Snapshot (computer storage)"
- [22] <http://www.faqs.org/docs/evms/whatissnapshotting.html>
"EVMS User Guide, What is a snapshot?"

en 1992. Obtuvo los títulos de M.Sc. (1997) y Ph.D. (2002) en Computer Engineering en Syracuse University, NY, USA. Ha realizado cursos especializados en varios países europeos, latinoamericanos, Estados Unidos y en Corea del Sur. Actualmente trabaja en la EPN, en el Departamento de Electrónica, Telecomunicaciones y Redes de Información.



Elisa K. Mena, nació en Quito, Ecuador el 8 de octubre de 1986, sus estudios superiores los realizó en la Escuela Politécnica Nacional en la Carrera de Ingeniería Electrónica y Redes de Información.

En enero del 2009 realiza sus pasantías en Sonda de Ecuador y en octubre del mismo año se inicia como docente en el Instituto Superior Tecnológico para el Desarrollo, impartiendo clases de programación en Visual Basic .NET y de diseño de Bases de Datos con SQL. En marzo del año 2010 ingresa como Ayudante Técnico-Administrativo a la Facultad de Ciencias de la Escuela Politécnica Nacional donde se encarga de asesoría técnica, mantenimiento de computadores y desarrollo e implementación de sistemas web para la facultad. Actualmente se encuentra concluyendo la redacción de su proyecto de Titulación y trabaja en TATA Consultancy Services (TCS), desarrollando proyectos relacionados con actividades bancarias.



Ana C. Guerrero, nació en Quito, Ecuador, el 18 de octubre de 1986. Estudiante de Ingeniería Electrónica y Redes de Información de la Escuela Politécnica Nacional, cursó los estudios correspondientes a Linux en la Academia Linux y cursa sus estudios para certificarse como CCNA (Cisco Certified Network Associated) en la Academia Cisco, en la Escuela Politécnica Nacional. Actualmente se encuentra concluyendo la redacción de su Proyecto de Titulación.

Ha realizado prácticas pre-profesionales en la Dirección Nacional de Comunicaciones de la Policía Nacional del Ecuador – División Redes, donde realizó estudios para la implementación de sistemas de seguridad, manejo de antenas, análisis del proyecto de videovigilancia "Ojos de Águila Quito", entre otros.



Iván M. Bernal, graduado del Instituto Nacional Mejía. Obtuvo el título de Ingeniero en Electrónica y Telecomunicaciones en la Escuela Politécnica Nacional