

Trabajo - Instalación, Administración y Uso de un servicio en Linux

OpenNebula

Servicios Telemáticos Avanzados
Departamento de Ingeniería Telemática

Jon Senra Dearle

©LSC 2014/2015

ÍNDICE

1. Objetivos y Alcance.....	4
1.1. Introducción.....	4
1.2. Motivación y funcionalidad del servicio	4
1.3. Documentación bibliográfica.....	4
2. Base Teórica	4
2.1. Descripción del servicio y conceptos implicados	4
2.2. Análisis de protocolos.....	4
2.2.1. Funcionamiento y estructura del protocolo	4
2.2.2. Uso del protocolo en el servicio	4
3. Implementación estudiada para ofrecer el servicio	5
3.1. Referencias y características de la implementación estudiada	5
3.1.1. Descripción	5
3.1.2. Equipamiento necesario	5
3.1.3. Características y funcionalidades	5
3.2. Comparativa de soluciones existentes en el mercado	5
3.3. Clientes para el servicio	5
3.3.1. Referencias y características del cliente adoptado	5
3.3.2. Comparativa de clientes existentes en el mercado	5
4. Proceso de instalación/administración del cliente	6
4.1. Obtención del software del cliente.....	6

4.2. Instalación del cliente	6
4.3. Configuración del cliente	6
5. Proceso de instalación/administración del servidor.....	6
5.1. Obtención del software del servidor	6
5.2. Instalación del servidor	6
5.2.1. Primera instalación del servicio	6
5.2.2. Actualización del servicio	6
5.2.3. Desinstalación del servicio	7
5.3. Configuración del servidor	7
5.4. Puesta en funcionamiento del servicio.....	7
5.5. Arranque del servicio con el sistema	7
5.6. Administración y monitorización del funcionamiento	7
5.6.1. Arranque del servicio en modo detallado	7
5.6.2. Activación/configuración de los ficheros de registro	7
5.6.3. Seguridad del servicio	7
5.7. Tests y pruebas del servicio	7
6. Análisis del intercambio de mensajes por la red	8
7. Interfaz gráfica de gestión del servidor	8
8. Diseño de los escenarios de prueba	8
9. Deficiencias del servicio.....	8
10. Ampliaciones/mejoras del servicio.....	8
11. Incidencias y principales problemas detectados.....	8
12. Resumen y conclusiones.....	8

1. Objetivos y Alcance

1.1. Introducción

La computación en nube es un sistema informático basado en Internet y centros de datos remotos para gestionar servicios de información y aplicaciones. La computación en nube permite que los consumidores y las empresas gestionen archivos y utilicen aplicaciones sin necesidad de instalarlas en cualquier computadora con acceso a Internet. Esta tecnología ofrece un uso mucho más eficiente de recursos, como almacenamiento, memoria, procesamiento y ancho de banda, al proveer solamente los recursos necesarios en cada momento.

El término “nube” se utiliza como una metáfora de Internet y se origina en la nube utilizada para representar Internet en los diagramas de red como una abstracción de la infraestructura que representa.

Un ejemplo sencillo de computación en nube es el sistema de documentos y aplicaciones electrónicas Google Docs / Google Apps. Para su uso no es necesario instalar software o disponer de un servidor, basta con una conexión a Internet para poder utilizar cualquiera de sus servicios.

El servidor y el software de gestión se encuentran en la nube (Internet) y son directamente gestionados por el proveedor de servicios. De esta manera, es mucho más simple para el consumidor disfrutar de los beneficios. En otras palabras: la tecnología de la información se convierte en un servicio, que se consume de la misma manera que consumimos la electricidad o el agua.

En el contexto creciente de la computación distribuida, nace el proyecto OpenNebula se consolidó como proyecto de investigación en el 2005 por Ignacio M Llorente y Rubén S. Montero. Desde su primer lanzamiento público de software en marzo de 2008, a evolucionado a través de lanzamientos open-source y opera actualmente como proyecto open source. OpenNebula es el resultado de muchos años de investigación y desarrollo sobre gestión eficiente y escalable de máquinas virtuales en infraestructuras altamente distribuidas en colaboración cercana con su comunidad de usuarios y los principales participantes en el mundo del cloud computing.

La tecnología OpenNebula ha madurado gracias a una comunidad activa y comprometida de usuarios y desarrolladores. OpenNebula es software descargado varios miles de veces por mes, además de disponer del código en su repositorio de software. Además de un crecimiento exponencial en el número de sus usuarios, diferentes proyectos, grupos de investigación y compañías han creado nuevos componentes de virtualización y cloud para complementar y potenciar la funcionalidad que aporta este ampliamente usado toolkit open source para la programación cloud.

En marzo de 2010, los principales autores de OpenNebula fundaron C12G Labs (conocido actualmente como OpenNebula Systems) para proveer el valor añadido de servicios profesionales que muchas tiendas de IT requieren para adopción interna y para permitir al proyecto OpenNebula desligarse exclusivamente de la financiación pública, contribuyendo a su sostenibilidad a largo plazo. OpenNebula.org es un proyecto actualmente gestionado y soportado por OpenNebula Systems. En septiembre de 2013, OpenNebula organizó su primera conferencia ante la comunidad, cuyas presentaciones fueron llevadas a cabo por organizaciones de todos los confines del mundo.

El servicio de computación y virtualización distribuida OpenNebula será el objeto de estudio del presente trabajo.

1.2. Motivación y funcionalidad del servicio

1.2.1 ¿Qué es la tecnología OpenNebula?

OpenNebula provee la solución **más simple pero rica en prestaciones** para la gestión comprensiva de CPD virtualizados para permitir nubes IaaS públicas, privadas y mixtas. La interoperabilidad de OpenNebula hace al cloud una evolución mediante el manejo de activos IT existentes, protegiendo las inversiones y evitando el lock-in a vendedores concretos.

OpenNebula es una solución llave en mano preparada para empresas que incluye todas las prestaciones requeridas para proveer soluciones de nube privada y ofrecer servicios de nube pública.

1.2.2 ¿Qué funcionalidades aporta OpenNebula?

OpenNebula provee prestaciones en las dos principales capas de virtualización de Data Centers e infraestructuras de nube:

- **Gestión de virtualización de Data Centers.** Muchos usuarios emplean OpenNebula para gestionar la virtualización de Data Centers, consolidar servidores e integrar activos IT existentes para computación, almacenamiento y networking. En este modelo de despliegue, OpenNebula se integra directamente con hipervisores (como KVM, Xen o VMware ESX) y tiene control completo sobre recursos virtuales y físicos, proveyendo funcionalidades avanzadas para la gestión de capacidad, optimización de recursos, alta disponibilidad y continuidad del negocio. Algunos de dichos usuarios también disfrutaban de la gestión del cloud de OpenNebula y prestaciones de aprovisionamiento cuando adicionalmente desean federar CPD, implementar cloudbursting o ofrecer portales de servicios propios para usuarios.
- **Gestión Cloud.** También existen usuarios que emplean OpenNebula para proveer una capa de aprovisionamiento, de tipo cloud, multi-alojada encima de una solución existente de gestión de infraestructuras. Dichos usuarios buscan características de aprovisionamiento, elasticidad y multialojamiento tales como aprovisionamiento de Data Centers, federación de Data Centers o computación cloud híbrida para conectar infraestructuras domésticas con nubes públicas, mientras que la infraestructura es manejada y gestionada por herramientas familiares para tales efectos.

1.2.3 Principios de diseño

La tecnología OpenNebula es el resultado de muchos años de investigación y desarrollo en la gestión escalable y eficiente de máquinas virtuales en infraestructuras distribuidas de alta escala. OpenNebula se diseñó para cumplir los requisitos de casos de uso de negocio de compañías líderes y a través de múltiples industrias, tales como hosting, telecomunicaciones, eGobierno, utilidades de computación... Los principios que han guiado el diseño de OpenNebula son:

- Arquitectura, interfaces y código **abiertos**.
- **Flexibilidad** para adaptarse a cualquier CPD.
- **Interoperabilidad y portabilidad** para prevenir lock-in a vendedores.
- **Estabilidad** para el uso en entornos de producción de clase empresarial.
- **Escalabilidad** para infraestructuras de alta escala.
- **Centrarse en el enfoque del administrador**, con control completo sobre el cloud.
- **Simplicidad**, facilidad de despliegue, operación y uso.
- **Ligereza** para alta eficiencia.

1.2.4 Beneficios

1.2.4.1 Para el gestor de infraestructura

- **Rápida respuesta para necesidades de servicios** con redimensión dinámica de la infraestructura física mediante la adición de nuevos host, y la partición dinámica de clusters para complacer requisitos de capacidad de servicios.
- **Gestión centralizada** de toda la infraestructura virtual y física.
- **Uso más eficiente de los recursos existentes** con la creación de una infraestructura que incorpora los recursos heterogéneos en el datacenter, y la compartición de infraestructuras entre diferentes departamentos gestionando cada uno sus clusters de producción.
- **Ahorro operacional** con la consolidación de servidores a un número reducido de sistemas físicos, reduciendo el espacio requerido, el esfuerzo de administración y los requisitos de potencia y refrigeración.
- **Menores costes de infraestructura** con la combinación de nubes locales y remotas, eliminando la sobre compra de sistemas para complacer demandas de pico.

1.2.4.2 Para el usuario de infraestructura

- **Entrega rápida y escalabilidad de servicios** para complacer demandas dinámicas de servicio por parte de usuarios finales.
- **Soporte para la ejecución heterogénea de entornos** con requisitos de software múltiples, incluso conflictivos, sobre la misma arquitectura compartida.
- **Control total** del ciclo de vida de gestión de virtualización de servicios.

1.2.4.3 Para integradores de sistemas

- **Encajar en cualquier CPD existente** gracias a sus componentes, interfaces y arquitectura abiertos, flexibles y extensibles.
- **Construir cualquier tipo de despliegue cloud.**
- **Software abierto**, licencia Apache.

- **Integración sencilla con cualquier producto y servicio en el ecosistema de virtualización/nubes o herramienta de gestión de CPD**, tales como proveedores cloud, gestores VM, gestores de imágenes virtuales, gestores de servicios, herramientas de gestión, planificadores...

1.2.5 ¿Por qué OpenNebula?

“Todo debería hacerse lo más simple posible, pero no más simple”, Albert Einstein

Se debería construir software que cualquier persona pueda usar.

No para uno mismo, no para otro, no para expertos en tecnologías emergentes, sino para cualquiera interesado en construir un cloud.

OpenNebula es la alternativa a OpenStack, que es demasiado complejo, y a VMware, que es demasiado caro e inflexible.

1.3. Documentación bibliográfica

<http://www.revistacloud.computing.com>

<http://opennebula.org/about>

2. Base Teórica

OpenNebula es una infraestructura cloud integrada en un solo componente, procedente del paquete opennebula. No obstante, sus herramientas de gestión (el CLI opennebula-tools y el gestor gráfico opennebula-sunstone) son independientes a dicha infraestructura. Una infraestructura cloud es una unión compleja de componentes, y OpenNebula, pese a ser la solución más sencilla para el despliegue de nubes, como veremos a continuación, no es una excepción.

2.1. Descripción del servicio y conceptos implicados

2.1.1 Componentes principales

- **Interfaces y APIs:** OpenNebula provee diversas interfaces que se pueden emplear para interactuar con las funcionalidades ofrecidas para manejar recursos físicos y virtuales. Hay dos maneras principales de manejar instancias OpenNebula: mediante interfaz de línea de comandos y el GUI Sunstone. También existen diversas interfaces cloud que pueden emplearse para crear nubes públicas: OCCI y EC3 Query, además de un portal de servicio propio para consumidores cloud. Como adición, OpenNebula ofrece potentes APIs de integración para permitir el desarrollo fácil de nuevos componentes.
- **Usuarios y grupos:** OpenNebula soporta cuentas de usuario y grupos, además de varios mecanismos de autenticación y autorización. Esta característica puede usarse para crear varios compartimentos aislados dentro de la misma nube, implementando multi alojamiento. Además, un potente mecanismo de lista de control de acceso permite distintas gestiones según roles, permitiendo otorgar permisos de manera muy granulada.
- **Hosts:** Varios hipervisores son soportados por el gestor de virtualización, con la habilidad de controlar el ciclo de vida de máquinas virtuales, además de monitorizarlas. Dicha monitorización también se aplica a los hosts físicos. Los hipervisores principales soportados son: Xen, KVM y VMWARE.
- **Redes:** Hay presente en OpenNebula un subsistema de red fácilmente adaptable y personalizable para permitir una mejor integración con los requisitos específicos de red de CPD existentes. Se ofrecen además soporte de VLANs y Open vSwitch.
- **Almacenamiento:** OpenNebula es suficientemente flexible para soportar diversas configuraciones de almacenamiento de imágenes. El soporte para múltiples sistemas de almacenamiento provee una flexibilidad extrema en la planificación del almacenamiento backend e importantes beneficios de rendimiento. Las principales configuraciones de almacenamiento son soportadas: datastore basado en sistema de archivos para almacenar imágenes de disco en forma de archivo y con transmisión de imágenes empleando ssh o sistemas de archivos compartidos (NFS, GlusterFs, Lustre...), iSCSI/LVM para almacenar imágenes de disco en forma de dispositivo de bloque y datastore VMWARE, especializado para el hipervisor VMware, que maneja el formato vmdk.

- **Clusters:** Los clusters son conjuntos de hosts que comparten datastores y redes virtuales. Los clusters se emplean para balanceo de carga, alta disponibilidad y computación de alto rendimiento.

2.1.2 Operación del sistema

OpenNebula orquesta almacenamiento, redes, virtualización, monitorización y tecnologías de seguridad para permitir el emplazamiento dinámico de servicios multicapa (grupos de máquinas virtuales interconectadas) en infraestructuras distribuidas, combinando tanto recursos de CPD como recursos de nubes remotas, de acuerdo a las siguientes políticas:

- **Manejo de red, computación y capacidad de almacenamiento:** Orquestado de almacenamiento, red y tecnologías de virtualización para permitir el emplazamiento dinámico de servicios distribuidos.
- **Manejo del ciclo de vida de VM:** Ejecución fluida de VM mediante la asignación de recursos requeridos por las mismas para operar y ofreciendo la funcionalidad requerida para implementar políticas de instanciación de VM.
- **Manejo de la asignación de carga de trabajo:** Soporte para la definición de la carga de trabajo y asignación mediante políticas pendientes de la disponibilidad de recursos, tales como la consolidación para eficiencia energética, balanceo de carga, consideración de compatibilidades, reserva de capacidad...
- **Manejo de redes virtuales:** Soporte para la definición de redes virtuales para interconectar VM.
- **Manejo de imágenes de VM:** Exposición de los mecanismos generales para transferir y clonar imágenes de VM. Las imágenes se pueden registrar antes de su ejecución. Cuando se registran, las imágenes de VM se transfieren al host y las imágenes de swap de disco se crean. Después de la ejecución, las imágenes pueden copiarse de vuelta al repositorio.
- **Manejo de información y contabilidad:** Provisión de indicadores que se pueden emplear para diagnosticar la correcta operación de los servidores y las VM y para soportar la implementación de políticas de ubicación dinámica de VM.
- **Manejo de la seguridad:** Definición de políticas de seguridad sobre los usuarios del sistema, garantizando que los recursos se utilizan únicamente por usuarios acreditados y el aislamiento entre cargas de trabajo.

- **Gestión de la capacidad de nubes remotas:** La extensión dinámica de capacidad local con recursos de proveedores remotos para construir despliegues híbridos o federados.
- **Manejo de servidores cloud públicos:** Exposición de las interfaces cloud más comunes para soportar despliegues públicos de computación en la nube.

2.2. *Análisis de protocolos*

Debido a la complejidad extensa que supone el conjunto de servicios y protocolos necesarios para el despliegue de una nube, se analizarán únicamente los protocolos que se consideren más relevantes o característicos del servicio OpenNebula, asumiendo que se entienden las tecnologías de menor nivel sobre las que dichos protocolos funcionan.

2.2.1. *Funcionamiento y estructura del protocolo SSH*

SSH (o Secure *SHell*) es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas.

SSH está diseñado para reemplazar los métodos más viejos y menos seguros para registrarse remotamente en otro sistema a través de la shell de comando, tales como **telnet** o **rsh**. Un programa relacionado, el **scp**, reemplaza otros programas diseñados para copiar archivos entre hosts como **rcp**. Ya que estas aplicaciones antiguas no encriptan contraseñas entre el cliente y el servidor, evite usarlas mientras le sea posible. El uso de métodos seguros para registrarse remotamente a otros sistemas reduce los riesgos de seguridad tanto para el sistema cliente como para el sistema remoto.

2.2.2. *Uso del protocolo SSH en el servicio*

SSH se emplea en OpenNebula mediante la compartición de ssh keys entre el frontend y los hosts para que el primero pueda acceder a la configuración y estadísticas de los segundos.

2.2.3. Funcionamiento y estructura del protocolo X.509

El X.509 es un estándar desarrollado en 1998 por el ITU-T (International Telecommunication Union-Telecommunication Standardization Sector) y el ISO/IEC (International Standards Organization/International Electrotechnical Commission) para definir el formato de los certificados digitales.

En este formato se incluye, además del par de claves del certificado, información adicional que permite delimitar el tiempo de validez, información sobre la Autoridad de Certificación que lo ha generado y otros datos que permiten aplicar directivas para su gestión y control.

2.2.4. Uso del protocolo X.509 en el servicio

X.509 se utiliza como mecanismo de seguridad adicional en las conexiones desde la interfaz de gestión CLI.

2.2.5 Funcionamiento y estructura del protocolo SSL

SSL significa "Secure Sockets Layer". Secure Sockets Layer es un protocolo diseñado para permitir que las aplicaciones puedan transmitir información de manera segura. Las aplicaciones que utilizan el protocolo Secure Sockets Layer disponen de un medio para dar y recibir claves de cifrado con otras aplicaciones, así como la manera de cifrar y descifrar los datos enviados entre los dos.

2.2.6 Uso del protocolo SSL en el servicio

SSL se emplea como el mecanismo para el acceso seguro al servidor de la interfaz de gestión gráfica Sunstone.

2.2.7 Funcionamiento y estructura del protocolo HTTP

Hypertext Transfer Protocol, o HTTP, es el protocolo usado en cada transacción de la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un spider) se lo conoce

como "user agent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (URL). El resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

2.2.8 Uso del protocolo HTTP en el servicio

Puesto que la interfaz de gestión gráfica de OpenNebula, Sunstone, es una interfaz web, se emplea el protocolo HTTP para obtener la información del servidor.

2.2.9 Funcionamiento y estructura del protocolo LDAP

LDAP son las siglas de Lightweight Directory Access Protocol (en español Protocolo Ligero de Acceso a Directorios) que hacen referencia a un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también se considera una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

2.2.10 Uso del protocolo LDAP en el servicio

LDAP se emplea en OpenNebula para desplegar sistemas de almacenamiento de imágenes VM distribuidos.

2.2.11 Funcionamiento y estructura del protocolo XML-RPC

XML-RPC es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.

Es un protocolo muy simple ya que solo define unos cuantos tipos de datos y comandos útiles, además de una descripción completa de corta extensión. La simplicidad del XML-RPC contrasta con la mayoría de protocolos RPC que tiene una documentación extensa y requiere considerable soporte de software para su uso.

2.2.12 Uso del protocolo XML-RPC

XML-RPC es muy importante en OpenNebula, pues se emplea para operar sobre los hosts, clusters, datastores y, en general, todos los elementos que componen la infraestructura física del cloud.

2.2.13 Funcionamiento y estructura del protocolo EC2

Amazon Elastic Compute Cloud (Amazon EC2) es una parte central de la plataforma de cómputo en la nube de la empresa Amazon.com denominada Amazon Web Services (AWS). EC2 permite a los usuarios rentar computadores virtuales en los cuales poder correr sus propias aplicaciones. Este tipo de servicio supone un cambio en el modelo informático al proporcionar capacidad informática con tamaño modificable en la nube, pagando por la capacidad utilizada. En lugar de comprar o alquilar un determinado procesador para utilizarlo varios meses o años, en EC2 se alquila la capacidad por horas.

2.2.14 Uso del protocolo EC2 en el servicio

OpenNebula emplea la API de EC2 para implementar nubes híbridas o federadas, combinando la infraestructura privada con la capacidad de servidores Amazon alquilada.

3. Implementación estudiada para ofrecer el servicio

La implementación estudiada es OpenNebula 3.4, aquella compatible con la versión Stable de Debian 7 (Wheezy) empleada por el departamento de telemática. Además, por enormes problemas de dependencias incumplidas/obsoletas/rotas para las diversas versiones testadas previo a la selección de la presente versión, se instruirá únicamente en el proceso de instalación mediante los repositorios oficiales de Debian, puesto que la descripción adicional de todo el proceso para la instalación desde código fuente supondría una extensión excesiva para el alcance del trabajo.

3.1. Referencias y características de la implementación estudiada

Muchos de los rasgos generales explicados en los anteriores apartados se aplican a la versión estudiada, se hará pues referencia en el presente apartado a los rasgos particulares de dicha versión.

3.1.1. Descripción

El frontend requiere que el equipo disponga de ruby con versión igual o superior a 1.8.7, asimismo, se requieren diversos módulos de ruby (gems) que se pueden instalar, como veremos posteriormente, con un script proporcionado con el software.

Los hosts requieren un hipervisor compatible con OpenNebula. Los principales hipervisores compatibles son KVM, Xen y VMware. El hipervisor que se empleará en este trabajo es Xen.

A continuación se muestra un diagrama que muestra todas las dependencias de los diversos componentes que forman OpenNebula:

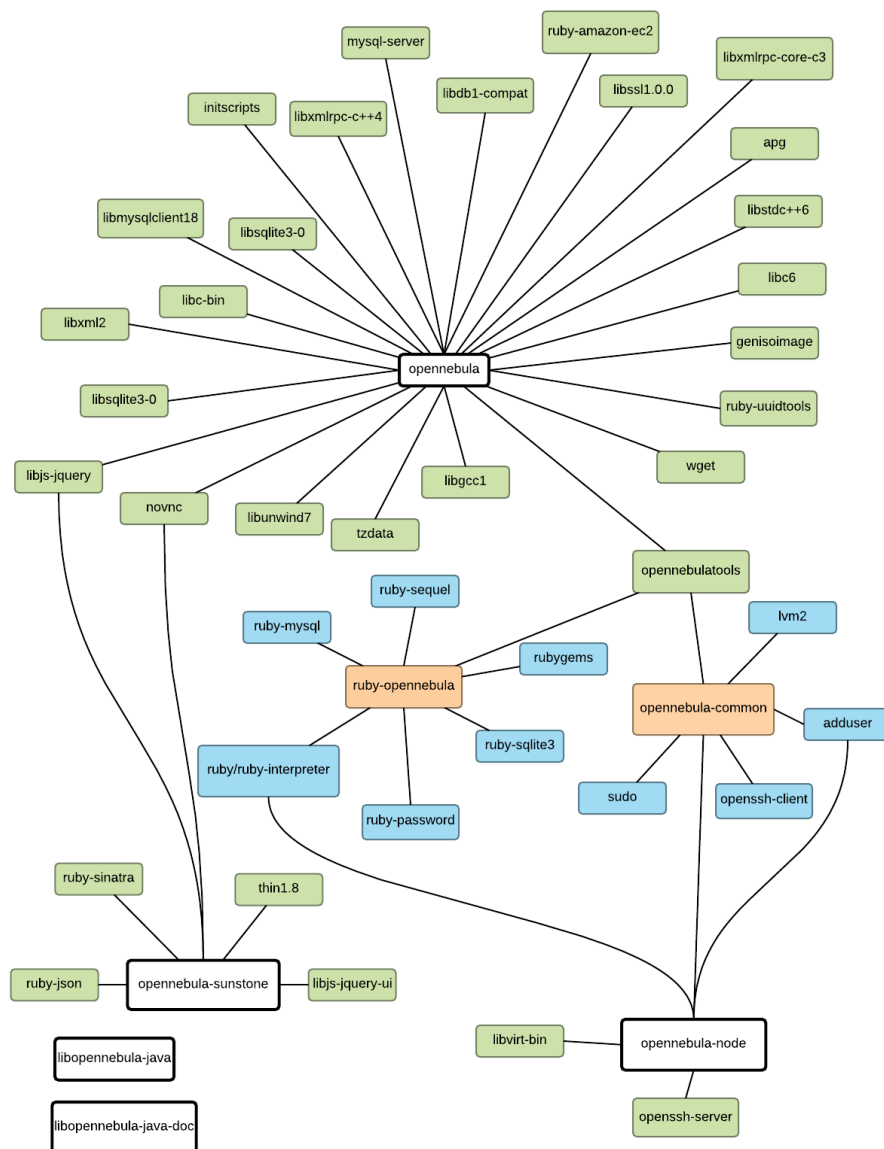


Gráfico 1 – Dependencias de paquetes del entorno opennebula.

3.1.2. Equipamiento necesario

El frontend de OpenNebula ocupa menos de 10MB una vez instalado, con lo cual puede instalarse en cualquier equipo, provisto que dispone de todos los paquetes para cumplir las dependencias del software recogidas en el apartado anterior.

Los host no requieren de preinstalación de ningún paquete. Sin embargo, el paquete `openebula-node`, ofrece una manera sencilla para preparar un equipo para convertirlo en un host `openebula`. Los hosts tan sólo deben disponer de un servidor `ssh` y tener configurado correctamente el hipervisor que se empleará para instanciar las imágenes de VM en el host.

Para una implementación real, se recomienda disponer de una red dedicada que interconecte los diversos equipos. Se recomienda además, por motivos de seguridad, que los elementos que forman la infraestructura del cloud se separen en equipos físicos distintos (`datastore`, `hosts`, `frontend`, etc.).

En particular, el proceso descrito se hace sobre equipos con SO Debian 7 (Wheezy).

3.1.3. Características y funcionalidades

A continuación detallaremos las principales funcionalidades y características clave de OpenNebula 3.4:

- Poderosa gestión de seguridad de usuarios.
- Gestión avanzada de multi alojamiento mediante grupos.
- Asignación bajo demanda de data centers virtuales.
- Control y monitorización avanzados de la infraestructura virtual.
- Configuración flexible y completa de máquinas virtuales.
- Control y monitorización avanzados de la infraestructura física.
- Soporte para plataformas empresariales, comodidad en general.
- Optimización de recursos distribuidos.
- Gestión centralizada de zonas múltiples.

- Alta disponibilidad.
- Computación de nube híbrida y cloudbursting.
- Interfaces cloud estándar y portal simple de servicios propios para consumidores cloud.
- Interfaz de comandos rica y extensiva e interfaces web para administradores cloud.
- Opciones de despliegue múltiples.
- Fácil extensión e integración.
- Seguridad, eficiencia y escalabilidad masivas.

3.2. Comparativa de soluciones existentes en el mercado

Los autores de OpenNebula la presentan como la alternativa para OpenStack, el coloso del despliegue de infraestructuras cloud basadas en proveedores. Frente a la complejidad de configuración y gestión, y el alto número de subproyectos de OpenStack, OpenNebula ofrece una interfaz de despliegue, gestión y operación cloud enfocada a la simplicidad de uso, tanto para usuarios, como para administradores. A continuación se recogen las principales diferencias entre los proyectos OpenNebula y OpenStack:

OpenNebula	OpenStack
Orientado a usuarios	Orientado a proveedores
Sólo un proyecto de integración de subsistemas. La infraestructura supone la unión de los diversos subsistemas.	Varios proyectos, uno por subsistema. Cada subsistema se despliega sobre una infraestructura común.
Modelo de gobernanza basado en “administrador solidario”: organización dedicada al interés y estrategia del proyecto	Modelo de gobernanza basado en una fundación constituida por los proveedores que son los principales colaboradores del proyecto.
Perfil principal de los contribuidores al proyecto: usuarios del mismo	Perfil principal de los contribuidores al proyecto: vendedores
Modelo de gestión centralizado con dirección individual	Modelo de gestión descentralizado con dirección colectiva

“Por usuarios para usuarios”	“Por proveedores para proveedores”
Modelo de nube: enterprise	Modelo de nube: public

3.3. *Clientes para el servicio*

Hay multitud de empresas líderes que emplean OpenNebula, entre las cuales podemos encontrar Telefónica, Blackberry, RedIris, CSIC, BBC, Unity, Ongrid, Runtastic, NASA y muchos otros. Existen diversos proyectos internacionales que emplean OpenNebula, uno en concreto es el proyecto BonFire, en el cual la empresa andaluza Wellness Telecom es participante. En concreto, analizaremos el caso particular de BBC.

3.3.1. *Referencias y características del cliente adoptado*

Existe una sección entera de la BBC dedicada al análisis e investigación de datos de usuarios, encuestas, intereses, etc. destinados a la mejora de las noticias y servicios que ofrecen. En concreto, existen departamentos para captura (procesamiento de información de imágenes y vídeos), producción (estudios en la optimización de costes), entrega (estudios en la optimización de los canales de entrega al usuario), descubrimiento (estudio de las tendencias de usuario en los últimos 70 años y elaboración de previsiones inteligentes para las tendencias futuras) y experiencia (mejora de la experiencia de usuario relativa a los diferentes programas de la BBC). Cada departamento cuenta con una serie de clusters que manejan balanceo de carga, alta disponibilidad, etc. a su vez, todos los clusters de los departamentos se interconectan mediante redes virtuales, con lo cual toda la información es accesible por cualquier punto del sistema.

Los principales retos para las tecnologías de computación de BBC radican en el cumplimiento de objetivos tales como la aportar flexibilidad y libertad a sus ingenieros, crear silos de conocimiento a través de equipos diferentes, optimización del tiempo, supresión de inconsistencias, optimización del manejo de recursos, cumplir los requisitos de demanda de recursos de computación y almacenamiento. En una conferencia de OpenNebula, el vocal de BBC comentó que, tras la implantación de OpenNebula, dichos objetivos se cumplían con un gran salto en cuanto a optimización de recursos invertidos.

3.3.2. *Comparativa de clientes existentes en el mercado*

Como se citó anteriormente, existen muchos clientes de diversos sectores que emplean la tecnología OpenNebula, a continuación citaremos las principales industrias y sectores en los que OpenNebula figura:

- Telecomunicaciones e Internet.
- Gobiernos.
- Analistas financieros, de banca y de riesgos.
- Medios de comunicación y Gaming.
- Proveedores de hosting y MSPs.
- SaaS y eCommerce.
- Aeroespacial.
- Supercomputación.
- Investigación.
- Académico.
- Tecnologías de la información.
- Productos Cloud.
- Proyectos internacionales.

4. Proceso de instalación/administración del cliente

En esta sección describiremos el proceso de instalación del hipervisor Xen en los nodos del cloud a desplegar y la preparación del nodo como host OpenNebula. Xen es un hipervisor tipo 1, lo cual hace posible correr diversas instancias de un sistema operativo o distintos sistemas operativos en una sola máquina o host.

4.1. Obtención del software del cliente

Antes de profundizar en el proceso de instalación del hipervisor, cabe destacar las dependencias que hay que cumplir previa instalación, a continuación, en el siguiente esquema se reflejan dichas dependencias:

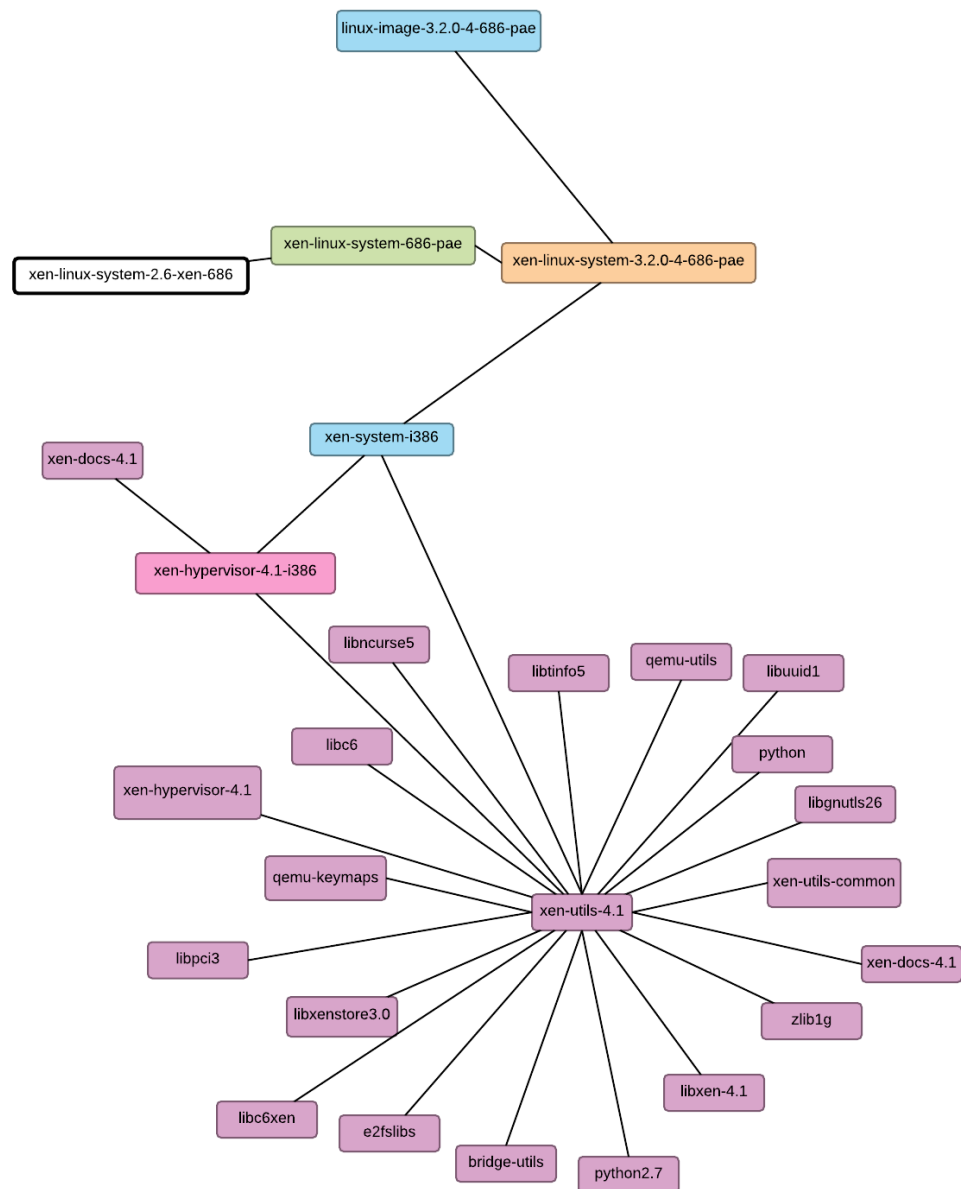


Gráfico 2 – Dependencias de paquetes del hipervisor Xen.

Nota: en el gráfico de dependencias anterior no se han recogido los paquetes de los que dependen los paquetes más inferiores de la jerarquía, puesto que sobrecargaría el diagrama.

4.2. *Instalación del cliente*

Los paquetes se instalan mediante el gestor apt-get vía conexión remota ssh indicando las órdenes como usuario root desde el script de configuración especificando la opción `-client`.

Los pasos a seguir son:

1. Invocación del script de instalación de OpenNebula (usuario root):

```
./install_OpenNebula.sh --client
```

2. Inserción de la dirección IP del host sobre el que se instalará el hipervisor.
3. Inserción de la contraseña del usuario root del equipo remoto.

El script procederá a instalar el meta paquete `xen-linux-system-2.6-xen-686`.

A continuación hay que instalar el paquete `opennebula-node`, el script realiza dicha tarea de forma automática tras las instalación del hipervisor Xen. No obstante, es preciso realizar ciertos cambios en la configuración del hipervisor para su posterior utilización, dichos cambios se detallan en el punto 4.3.

Una vez instalado `opennebula-node`, el host dispondrá de un usuario `oneadmin`, necesario para labores de administración. Sin embargo, es preciso que los id de usuario y grupos efectivos de dicho usuario sean los mismos que en el frontend, con lo cual se debe ejecutar:

```
id oneadmin
```

en el frontend, para obtener dichos id. Posteriormente se deben modificar los usuarios y grupos en el host para poder continuar con el proceso de configuración.

4.3. *Configuración del cliente*

Xen corre sobre un sistema operativo que denomina dominio 0. En general, se puede instalar un dominio 0 Debian como cualquier instalación Debian normal. La principal consideración a realizar es la distribución de las particiones puesto que ello tendrá impacto sobre las configuraciones de disco disponibles a los huéspedes. Si el dominio 0 Debian

será primariamente empleado para correr invitados, una buena regla es apartar 4GB para el sistema raíz del dominio 0 (/) y algo de espacio swap (swap=RAM si RAM<=2GB, swap=2GB si RAM>2GB). El espacio swap debería determinarse por la cantidad de RAM provista para el dominio0. Emplear el resto del espacio de disco para un volumen físico LVM.

4.3.1 Priorizar Grub por encima del SO nativo

A continuación hay que priorizar el kernel Xen como kernel principal en el listado de Grub 2. Se puede cambiar la configuración de Grub 2 desde su script de configuración (20_linux_xen) para que bootee Xen de manera prioritaria, ello se realiza dándole una prioridad mayor que la configuración estándar Linux (10_linux). Se puede emplear el comando dpkg-divert:

```
dpkg-divert --divert /etc/grub.d/08_linux_xen --rename  
/etc/grub.d/20_linux_xen
```

Para deshacer lo anterior:

```
dpkg-divert --rename --remove /etc/grub.d/20_linux_xen
```

Aplicar los cambios con:

```
update-grub
```

4.3.2 Configuración de red

Para poder dar acceso de red a dominios invitados, es necesaria la configuración de red adecuada del dominio 0. La configuración más común es usar un puente software.

Se recomienda la gestión de puente de red empleando el puente de red Debian. Si se dispone de un router que asigna direcciones ip mediante dhcp, la siguiente es un ejemplo funcional de configuración de /etc/network/interfaces empleando el software bridge-utils.

```
#The loopback network  
  
interface auto lo  
  
iface lo inet loopback  
  
iface eth0 inet manual  
  
auto xenbr0 iface xenbr0 inet dhcp
```

```
bridge_ports eth0

#other possibly useful options in a virtualized environment

#bridge_stp off

# disable Spanning Tree Protocol    #bridge_waitport 0

# no delay before a port becomes available

#bridge_fd 0

# no forwarding delay

## configure a (separate) bridge for the DomUs without giving Dom0 an IP
on it

#auto xenbr1

#iface xenbr1 inet manual

#    bridge_ports eth1
```

4.3.3 Instalación de dominio U (DomU o huésped)

xen-tools es un conjunto de scripts que permiten crear dominios huéspedes Xen totalmente configurados.

Una vez instalado el dominio 0, se puede instalar xen-tools en el host mediante

```
apt-get install xen-tools
```

El anterior paso se realiza de manera automatizada desde el script de instalación. Para configurar xen-tools, se puede editar el fichero /etc/xen-tools/xen-tools.conf, que contiene los valores por defecto que el script xen-create-image usará.

Para proveer un path diferente donde se guarden las imágenes de domU y permitir la contraseña de super usuario en la construcción inicial, se editará el fichero /etc/xen-tools/xen-tools.conf descomentando las siguientes líneas.

```
dir = /home/xen

passwd = 1
```

Entonces se pueden crear máquinas virtuales con el comando:

```
Xen-create-image --hostname <hostname> --ip <ip> --vcpus 2 --pygrub --  
dist <lenny|squeeze|maverick|loquesea>
```

Para iniciar la máquina virtual creada, se ejecuta el comando:

```
xm create /etc/xen/virtual_machine.cfg
```

Para borrar una vm:

```
xen-delete-image nombre_VM
```

4.3.4 Configuración de accesos secure Shell (Front-end)

Pese a que este paso sea de necesaria configuración en el frontend, está muy relacionado con el proceso de instalación del software en el host, con lo cual se detalla en esta sección. Es importantísimo que se realice este paso **posteriormente** a la instalación del software que se detalla en el punto 5.

Para realizar conexiones remotas sin la necesidad del uso de contraseñas, hay que generar claves ssh para el usuario oneadmin tanto para el frontend como para los hosts.

En el frontend:

1. Generar claves ssh para oneadmin (con dicho usuario ejecutar)

```
ssh-keygen
```

Cuando se requiera introducir una contraseña, pulsar intro para que la clave privada no sea encriptada.

2. Añadir la clave pública a ~/.ssh/authorized_keys para permitir al usuario oneadmin a acceder sin el uso de contraseña.

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

3. Asignar los permisos correctos.

```
chmod 700 ~/.ssh
```

```
chmod 600 ~/.ssh/id_rsa.pub
```

```
chmod 600 ~/.ssh/id_rsa
```



```
chmod 600 ~/.ssh/authorized_keys
```

4. Indicar al cliente ssh que no pregunte antes de añadir hosts al fichero known_hosts. Esto se configura en ~/.ssh/config, debería figurar algo como lo mostrado:

```
Cat ~/.ssh/config
```

```
Host *
```

```
StrictHostKeyChecking no
```

En el host:

1. Comprobar que el daemon sshd está ejecutándose. Eliminar cualquier opción banner en el fichero sshd_config en el host.
2. Copiar el directorio /var/lib/one/.ssh al host en la misma ruta.

Para verificar la configuración, se puede tratar de acceder vía ssh al host mediante el usuario oneadmin sin que se pida contraseña.

Tras los pasos anteriores se dispondrá del software de virtualización xen en cualquier host que se marque como objetivo de la instalación --client. Además, se dispondrá de todo el software necesario para que el host actúe como nodo OpenNebula y además se dispondrá de acceso directo ssh sin requisitos de contraseña puesto que se ha realizado una configuración de claves ssh. No obstante, cabe destacar que la instalación del hipervisor Xen no está completa, puesto que para la correcta configuración del dominio 0, tal y como se explicó antes, requiere de particionamiento específico dependiendo de las necesidades particulares de cada instalación. Se considera demasiado extenso el trabajo de configuración dedicado para tales efectos, se propone como futuro trabajo de la asignatura.

5. Proceso de instalación/administración del servidor

En esta sección describiremos el proceso de instalación configuración y administración del frontend de OpenNebula, la designación del centro de operaciones del cloud OpenNebula. Para consultar los requisitos del software, dirigirse a la sección 3 de este documento.

5.1. Obtención del software del servidor

Los paquetes y dependencias del servidor se instalan mediante el gestor apt-get a través del script de instalación `install_OpenNebula.sh`. Junto con el core de OpenNebula se instala el servidor gráfico Sunstone, que se discutirá en un punto posterior. El software se descarga de los repositorios oficiales de Debian 7 Wheezy.

5.2. Instalación del servidor

A continuación se detalla la estructura de directorios de la instalación de OpenNebula, para facilitar futuras referencias:

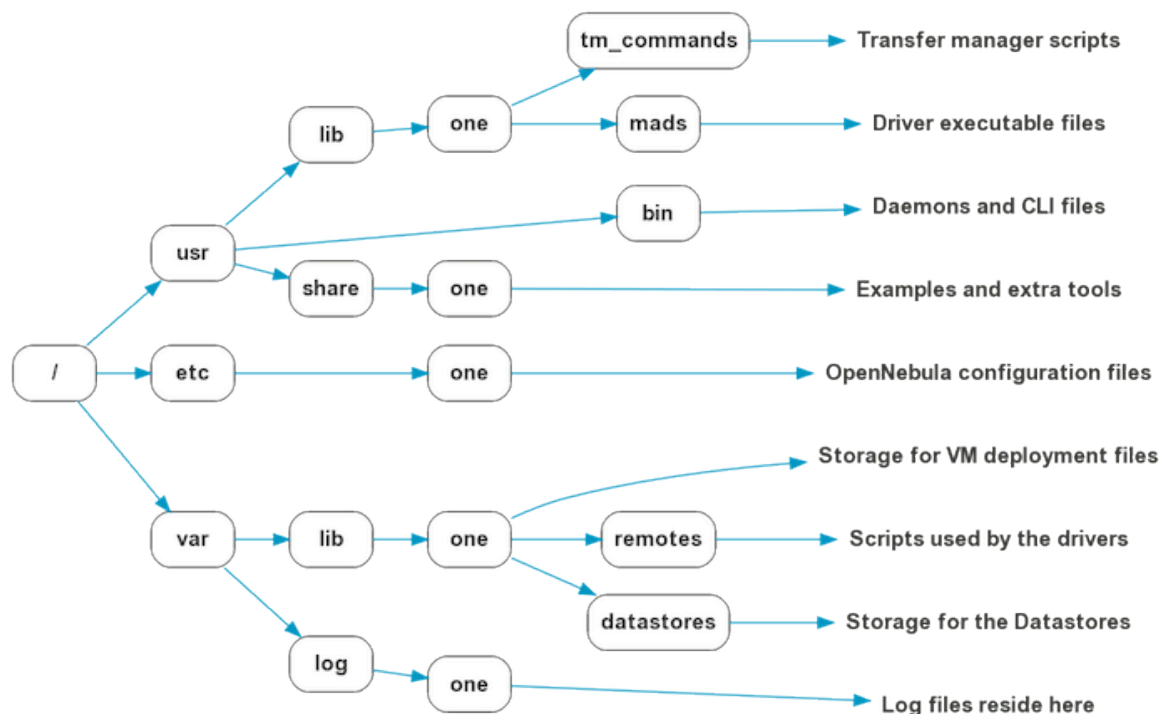


Gráfico 3 – Estructura de directorios del frontend OpenNebula.

5.2.1. Primera instalación del servicio

Los pasos a seguir son:

1. Ejecución del script de configuración, como root, bien sin argumentos o con el argumento `-v` para un modo detallado que, además de orientar al usuario durante el proceso de instalación, ameniza la espera durante dicho proceso.

```
./install_OpenNebula -v
```

2. Verificación de la instalación. Para ello, desde root ejecutar:

```
su - oneadmin
```

```
one start
```

```
onevm list
```

Debiendo obtener algo como:

ID	USER	GROUP	NAME	STAT	CPU	MEM	HOSTNAME
TIME							

5.2.2. Actualización del servicio

Como se indicó anteriormente, la versión de OpenNebula instalada (3.4) se trata de una versión relativamente antigua. Desde dicha versión se han lanzado las versiones 3.6, 3.8, 4.0, 4.2, 4.4, 4.6, 4.8 y 4.10. Se recomienda visualizar las guías “Upgrading from previous versions” de dichas versiones para actualizar OpenNebula. No se recomienda la actualización sobre el actual SO puesto que se trata de la versión Stable y para las últimas versiones es preciso realizar las instalaciones sobre distros Testing o incluso Unstable. Para tales casos, un buen punto de partida puede ser la página <http://archives.opennebula.org/documentation/archives:rel3.6:upgrade> que describe el proceso de actualización desde versiones antiguas para OpenNebula 3.6.

5.2.3. Desinstalación del servicio

Para desinstalar la versión actual del servicio, se puede invocar el script `install_OpenNebula.sh` con el argumento `--remove`, en tal caso el script desinstala los paquetes `libopennebula-java`, `libopennebula-java-doc`, `ruby-opennebula`, `opennebula-common`, `opennebulatools` y `opennebula`; en el orden indicado para evitar errores por dependencias incumplidas.

5.3. Configuración del servidor

El servidor principal de OpenNebula se puede configurar mediante el archivo `/etc/one/oned.conf`. Adicionalmente, se pueden configurar aspectos de contabilidad desde `/etc/one/acctd.conf`, aspectos del servidor econome y las llamadas a remoto xml-rpc desde `/etc/one/econome.conf`, aspectos relativos al servidor occi desde `/etc/one/occi-server.conf`,

similarmente, aspectos del servidor ozones desde `/etc/one/ozones-server.conf`, aspectos del planificador desde `sched.conf`, del servidor gráfico sunstone desde `/etc/one/sunstone-server.conf`.

5.4. *Puesta en funcionamiento del servicio*

El servicio se controla desde el frontend, bajo comandos desde el usuario `oneadmin`, los siguientes son ejemplos de inicio y parada de funcionamiento:

```
one start
```

```
one stop
```

Dichos comandos llaman al script de servicio `/usr/bin/one`, que a su vez inicia los ejecutables de servicio `/usr/bin/oned` y `/usr/bin/mm_sched`.

5.5. *Arranque del servicio con el sistema*

Para que el script de inicio para `oned` y `mm_sched` se inicie en el arranque del sistema, debemos copiarlo al directorio `init.d`:

```
cp /usr/bin/one /etc/init.d/one
```

A continuación debemos darle los permisos adecuados (755):

```
chmod 755 /etc/init.d/one
```

Por último, se debe incluir el script en los niveles de funcionamiento apropiados, la manera mas sencilla de realizar esto es mediante el comando `update-rc.d`:

```
update-rc.d one defaults
```

5.6. *Administración y monitorización del funcionamiento*

La monitorización de los diversos subsistemas se realiza desde diversos ejecutables que son controlados desde el usuario `oneadmin`, entre los que se destacan:

- `oneacl`: control de accesos para la gestión del cloud.
- `onecluster`: control y gestión de clusters del cloud.
- `onedatastore`: control y gestión de datastores para las plantillas de vm.

- onedb: gestión de la base de datos opennebula.
- onegroup: creación, eliminación, control de grupos de usuarios.
- onehost: creación, eliminación, control de hosts del cloud.
- oneimage: creación de imágenes a partir de archivo, control y gestión de las imágenes de vm.

Y otros como:

- onequota
- onetemplate
- oneuser
- onevdc
- onevm
- onevnet
- onezone

5.6.1. Arranque del servicio en modo detallado

Todos los comandos listados en 5.6 permiten el parámetro `-v` para la ejecución en modo detallado, el servidor no se puede arrancar en modo `-v` pero como `oned` es simplemente un core del sistema opennebula que se dedica a atender peticiones CLI, a gestionar los pool de host y todos los componentes del cloud, los eventos de su ejecución son volcados a un fichero de registro, con lo cual no tiene sentido incluir un comando de modo detallado para su ejecución.

5.6.2. Activación/configuración de los ficheros de registro

El fichero de registro por defecto para el core de OpenNebula es `/var/log/one/oned.log`, por defecto activado. El fichero de registro a usar, además de si se emplea o no, se puede

configurar desde `/etc/one/oned.conf`. La “verbosidad” de los logs se puede controlar con el nivel que se otorgue al parámetro `DEBUG_LEVEL` que puede ir desde 0 hasta 3.

5.6.3. Seguridad del servicio

La configuración de autenticación y seguridad del servicio se puede realizar mediante usuarios/grupos, acl, cuotas. El control y gestión de dichos elementos se puede realizar desde los subsistemas `oneuser`, `onegroup`, `oneacl` y `onequota`.

5.7. Tests y pruebas del servicio

Se proponen las siguientes pruebas de servicio:

Arranque del servicio: `su -; su - oneadmin; one start`

Añadir un host: `onehost create ip_host --im im_kvm --vm vmm_kvm --net dummy`

Listar hosts: `onehost list`

Mostrar información del host creado: `onehost show ip_host`

Deshabilitar el host creado: `onehost disable ip_host`

Eliminar el host creado: `onehost delete ip_host`

Parada del servicio: `one stop`

Creación de usuario y acceso desde sunstone (ver sección 7).

6. Análisis del intercambio de mensajes por la red

Por complejidad y extensión del trabajo, no procede.

7. Interfaz gráfica de gestión del servidor

La interfaz de gestión gráfica para OpenNebula es Sunstone GUI. Sunstone se debería haber instalado junto con oned y los demás componentes del sistema, adicionalmente se requieren ciertos pasos que se detallan a continuación:

Sunstone requiere ciertos gems de ruby, sin embargo, el entorno provee un script para la instalación de los mismos, para ejecutarlo escribir (es necesaria la ejecución desde root):

```
/usr/share/one/install_gems sunstone
```

Previo a dicho paso hay que instalar, desde root, el paquete dev de ruby:

```
apt-get install ruby-dev
```

A continuación, se propone como prueba, el inicio del servidor:

(Desde oneadmin): `sunstone-server start`

Para testarlo, se puede crear un usuario prueba:

```
oneuser create usuario pass
```

Y, al acceder desde un navegador web (sunstone corre por defecto en 127.0.0.1:9869) introduciendo las credenciales para el usuario, se debería mostrar el panel principal de gestión sunstone.

8. Diseño de los escenarios de prueba

Los ya recogidos en el apartado Tests y pruebas del servicio.

9. Deficiencias del servicio

- Los gráficos de monitorización de Sunstone no se muestran correctamente para oneadmin o usuarios en el grupo oneadmin.

Solución: Alterar la línea 268 de SunstoneServer.rb

```
filter[:uid] = opts[:uid] if opts[:gid]!="0"
```

por la siguiente

```
filter[:uid] = opts[:uid] if opts[:gid]!=0
```

y la línea 157 de `watch_client.rb`

```
filter[:uid]==0 ? (hosts = pool) : (return nil)
```

por

```
filter[:gid]==0 ? (hosts = pool) : (return nil)
```

- El driver de nube híbrida EC2 falla al intentar iniciarse.

Solución: Alterar la línea 196 y 363 del fichero `one_vmm_ec2.rb`, dejándolas como se detalla a continuación

```
196: deploy_exe = exec_and_log_ec2(:run, ec2_info, "", id)
```

```
363: def exec_and_log_ec2(action, xml, extra_params, id)
```

- El driver TM `qcow2` corrompe imágenes persistentes.

Solución: Alterar la línea 61 del fichero `mvds` dejándola como se detalla a continuación

```
if [ \ ( -L $SRC_PATH \) -a \ ( "\$SRC_READLN" = "\$DST_READLN"
\ ) ] ; then
```

10. Ampliaciones/mejoras del servicio

Se propone como mejora del servicio la compatibilización con el hipervisor de VirtualBox, Hyperbox, por tratarse de un proyecto de índole similar: OpenSource bajo licencia GPL2 (muy similar a la licencia Apache).

Como mejora adicional, se propone lo iniciado en el presente trabajo, la automatización de instalación y configuración mediante scripts habilitados especialmente para ello (haciendo especial hincapié en la configuración de los hipervisores compatibles con OpenNebula, puesto que es el aspecto que menos se explica en las guías de instalación), para cumplir

con la filosofía del proyecto OpenNebula “Crear un software que cualquiera interesado en desplegar un cloud, pueda instalar y manejar”.

11. Incidencias y principales problemas detectados

- Inicialmente se intentó instalar el software desde código fuente descargado desde el sitio web <http://opennebula.org> e incluso desde paquetes software provistos desde el mismo, lo cual resultó en una interminable ristra de errores de dependencias imposibles de cumplir debido a que se trataba de la versión Stable de Debian 7 Wheezy.
- Se trató de enviar las claves ssh para oneadmin mediante acceso sftp inverso desde un acceso ssh al host remoto, esto causaba un error por denegación de permisos que no se pudo resolver, se dejó en el script una indicación al usuario para que copie manualmente los ficheros al host remoto.
- La configuración completa del hipervisor Xen fue una tarea que inicialmente se abordó sin tener en cuenta la complejidad y dimensión del problema. Se resolvió no completarla por falta de tiempo y dejarla como propuesta de trabajo futuro.

12. Resumen y conclusiones

Para concluir el presente trabajo, querría realizar una serie de reflexiones acerca de la configuración y administración de un servicio, en concreto uno de gran envergadura como lo puede ser un gestor de cloud.

En los últimos años se ha podido observar de manera más o menos evidente la gran importancia que están cobrando los servicios distribuidos. Las aplicaciones de escritorio son cada vez más raras, cediendo paso a las populares “apps” web. Como ejemplo podemos citar el conocidísimo Microsoft Word del paquete Office, que ya dispone de versión online. Además, con la evolución de Internet y las tecnologías de la información, numerosas empresas comienzan a ofrecer servicios virtualizados, como lo puede ser la banca online.

En el clima del auge de la computación remota, en la cual la carga de trabajo se traslada de los equipos y dispositivos personales a los servidores de empresas e instituciones, la

computación distribuida o cloud computing emerge como solución eficaz para numerosos casos de uso.

En este trabajo se ha podido comprobar la utilidad que provee OpenNebula como sistema de gestión de nubes híbridas compuestas por cloud privadas incorporando servicios de nubes públicas como complemento. Las tareas de instalación y configuración del sistema gestor y la solución de problemas técnicos se han antojado retos de considerable dificultad dados los conocimientos que disponía a la hora de realizar el trabajo. Resulta inimaginable la carga de trabajo que supone la gestión de una nube de capacidades colosales como pueden ser los Web Services de Amazon, Azure de Microsoft o el Google Cloud.

Ha sido un proyecto complejo, extenso y exigente; sin embargo, creo que ha supuesto una de las tareas más interesantes que podría haber realizado, puesto que me ha puesto cara a cara con una tecnología de evidente emergencia y con gran futuro.

ANEXO A: Parámetros de configuración y comandos de gestión del servidor

A continuación se detallan los principales parámetros de configuración del fichero `/etc/one/oned.conf`:

MANAGER_TIMER: Tiempo en segundos entre funciones de evaluación periódicas del core.

HOST_MONITORING_INTERVAL y **VM_POLLING_INTERVAL** no pueden poseer temporizadores más pequeños que el **MANAGER_TIMER**.

HOST_MONITORING_INTERVAL: Tiempo entre segundos entre cada monitorización de host.

HOST_PER_INTERVAL: Número de hosts monitorizados en cada intervalo.

VM_POLLING_INTERVAL: Tiempo entre segundos entre cada monitorización de máquina virtual (usar 0 para deshabilitar la monitorización de VM).

VM_PER_INTERVAL: Número de VMs monitorizadas en cada intervalo.

SCRIPTS_REMOTE_DIR: Path remoto para almacenar los scripts de monitorización y gestión de VM.

PORT: Puerto donde oned escuchará llamadas remotas xml-rpc.

DB: Atributos de configuración para la bbdd del backend.

backend : puede ser sqlite or mysql (sqlite por defecto).

server : (mysql) nombre de host o dirección IP para el servidor MySQL.

port : (mysql) puerto para la conexión al servidor. Si se especifica 0 se emplea el puerto por defecto.

user : (mysql) el ID de login del usuario MySQL.

passwd : (mysql) la contraseña para el usuario.

db_name : (mysql) el nombre de la bbdd.

VNC_BASE_PORT: los puertos VNC para VMs pueden establecerse automáticamente iguales a VNC_BASE_PORT.

DEBUG_LEVEL: Nivel de “verbosidad” del registro de logs.

0 = ERROR, 1 = WARNING, 2 = INFO, 3 = DEBUG.

ANEXO B: Parámetros de configuración y comandos de gestión del cliente

No procede.