



UNIVERSIDAD
POLITECNICA
DE VALENCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

UNIVERSIDAD POLITÉCNICA DE VALENCIA

ESCUELA TÉCNICA SUPERIOR DE INFORMÁTICA APLICADA

**CONSTRUCCIÓN DE LABORATORIOS VIRTUALES PARA LA
ADMINISTRACIÓN DE SISTEMAS Y SERVIDORES**

PROYECTO FIN DE CARRERA

Autor: Marcos Martínez García
Director: José Ramón García Escrivá

Fecha del proyecto: 22/09/2010

ÍNDICE GENERAL

ÍNDICE GENERAL	3
CAPÍTULO 1 INTRODUCCIÓN Y NOCIONES PREVIAS	6
1.1 Motivaciones	6
1.2 Partes del sistema	6
1.3 ¿Cómo funciona?	7
1.4 Conocimientos y tecnologías implicadas	8
1.5 Introducción a la virtualización [41]	9
1.5.1 Ventajas e inconvenientes de virtualizar un sistema operativo	10
1.5.2 Programas interesantes	10
1.5.3 Tipos de Virtualización	10
1.5.4 Nuestro caso	11
1.5.5 Paravirtualización	11
1.6 Glosario de términos	11
CAPÍTULO 2 ANTECEDENTES	15
2.1 Software de administración de servidores y máquinas virtuales	15
2.1.1 Ovirt [1]	16
2.1.2 OpenQRM[2]	16
2.1.3 ProxMox VE [6]	23
2.2 Software de monitorización de la red	23
2.2.1 Zenoss [8]	23
2.2.2 EtherApe[11]	27
2.2.3 Wireshark [12] y Tshark [13]	28
CAPÍTULO 3 ANÁLISIS Y DESCRIPCIÓN DEL PRODUCTO	32
CAPÍTULO 4 DISEÑO	35
4.1 Asignando direcciones IP y configurando la red	35
Cómo funciona la red	37

4.2 Acceso a los elementos del sistema para su configuración	38
4.3 Instalación de Proxmox	40
4.4 Añadiendo nodos secundarios al nodo principal [18]	42
4.5 Crear una máquina virtual con ProxMox	43
4.6 Otras funciones de ProxMox	46
4.7 Crear y administrar una máquina virtual con instrucciones de OpenVZ	46
4.8 Copias de seguridad de las máquinas virtuales	48
4.9 Ejecución de máquinas virtuales creadas en Linux en sistemas Windows	50
4.10 Restauración rápida del sistema	51
Para el nodo principal	51
Para el nodo general o secundario	52
CAPÍTULO 5 RESULTADOS Y CONCLUSIONES	54
5.1 Simulación de uso	54
Creando una plantilla	55
Preparando el sistema	56
La parte del alumno	57
Monitorización y control	58
5.2 Valoración de resultados	59
5.3 Ampliaciones futuras	60
Futuros proyectos	60
Ejemplo de uso con interfaz web	60
Correcciones	62
CAPÍTULO 6 BIBLIOGRAFÍA	64
ANEXOS	68
A.1 Scripts	68
A.1.1 NodoPrincipal	68
A.1.2 NodoGeneral	68

Capítulo 1 INTRODUCCIÓN Y NOCIONES PREVIAS

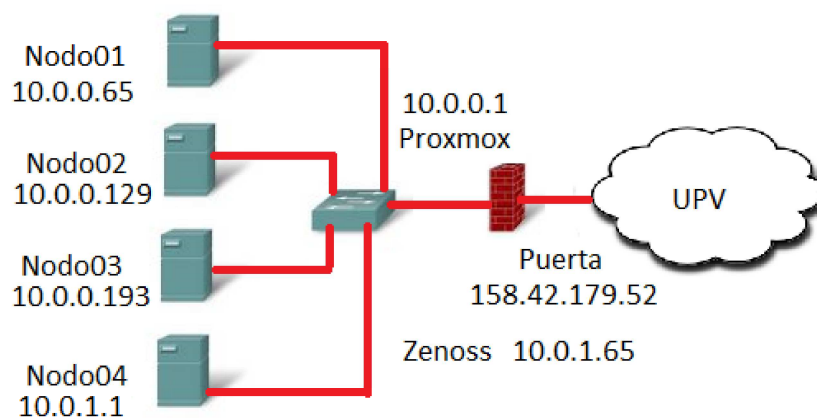
1.1 Motivaciones

Este proyecto viene dado por la necesidad de disponer de varios sistemas para cada alumno con el fin de realizar unas prácticas, para varias asignaturas, que necesitan de unos recursos que no podrían ser obtenidos de otra forma que no sea la virtualización por el coste que ello supone. Sobre todo si hablamos de prácticas con sistemas distribuidos. Por otro lado, estos recursos limitados no se ofrecen las veinticuatro horas del día lo cual resulta incómodo y poco flexible. Otro aspecto a tener en cuenta es que para realizar muchas prácticas interesantes es necesario contar con privilegios máximos, ya sea para tener acceso a modificar partes importantes del sistema operativo como para manipular la red o las relaciones entre varias máquinas diferentes, lo cual, hasta ahora, era una limitación y presentaba un gran problema de seguridad.

Con este proyecto hemos querido solucionar, entre otros, estos tres problemas: necesidad de un número mayor de recursos y de su disponibilidad y facilitar nuevas funciones y elementos configurables al alumno sin poner en riesgo la seguridad del sistema. Todo ello haciendo uso de software libre y reduciendo los costes únicamente a la compra de los equipos.

1.2 Partes del sistema

A continuación vamos a nombrar y definir los dispositivos que forman nuestro proyecto así como sus funciones:



Los dispositivos con los que contamos inicialmente serán: cuatro servidores con capacidad para almacenar un importante número de máquinas virtuales cada uno, un switch encargado de la interconexión de los servidores y un quinto nodo que hará las funciones de máster, firewall, monitor de red (gracias a Zenoss y Wireshark)... Finalmente cableado de red categoría 6 que conectará las tarjetas de red (dos por nodo) Gigabit Ethernet. Las partes más importantes del sistema son:

- 1- La entrada a nuestro sistema desde el exterior. Vendrá dada por el nodo principal del clúster llamado “puerta”. A él será donde lleguen las peticiones por parte de los alumnos que deseen tener acceso a los elementos que necesiten para sus prácticas. En este nodo se redireccionarán, gracias a IPTables, las conexiones entrantes hacia las máquinas virtuales correspondientes a cada alumno.
- 2- Vemos también que “puerta” tiene otra tarjeta de red que comunica con la parte interna de nuestra red y desde donde se mostrará con el nombre “proxmox”. Una de sus misiones principales es la de actuar como Gateway para los nodos secundarios.
- 3- Los nodos que vayamos añadiendo al sistema se nombrarán como nodo01, nodo02,... y serán los encargados de almacenar las MVs además de contar con un firewall por software que impedirá las conexiones entre MVs de diferentes alumnos. Todos ellos llevan el software ProxMox instalado.
- 4- Nuestro switch es un elemento que no admite ningún tipo de configuración y sólo realiza la función de permitir la comunicación Gigabit Ethernet entre los distintos elementos.

Todos los nodos son accesibles mediante SSH y mediante el puerto web seguro 443. Más tarde veremos cómo llevamos a cabo el reparto de direcciones IP.

1.3 ¿Cómo funciona?

Se describe, a continuación el funcionamiento básico del sistema:

- 1- Se registra al alumno en nuestro sistema asignándole varios puertos para que pueda repartirlos entre los diferentes servicios de sus máquinas virtuales.
- 2- El alumno conecta contra el puerto de la interfaz web proporcionada desde “puerta” una vez autenticado en la VPN de la UPV.
- 3- Desde allí es capaz de crear, administrar, guardar copias de seguridad o destruir sus máquinas virtuales.

- 4- Ahora, el alumno puede conectar a sus máquinas virtuales dirigiéndose a un puerto en concreto para cada MV dentro de “puerta” y poder así realizar las prácticas y las configuraciones oportunas.
- 5- El alumno puede descargar copias de seguridad de sus máquinas virtuales para poder trabajar en local después de formatearlas con el software adecuado.
- 6- Por último, un firewall impide que los alumnos puedan acceder a MVs que no les correspondan. En caso de que el profesor quiera comprobar que no hay fallos de seguridad en el sistema, es posible ver las conexiones abiertas en cada equipo gracias al software de monitorización.

1.4 Conocimientos y tecnologías implicadas

ProxMox[6]:

Es software libre capaz de crear y administrar máquinas virtuales basadas en plantillas así como de almacenar dichas plantillas. También tiene la capacidad de crear un clúster de varios nodos y distribuir la carga de trabajo entre ellos.

Veremos las características de este software con más detalle en el capítulo 2.

Open Virtuozzo:

Como todos los programas utilizados es libre. Su función es la de virtualizar y proporcionarnos una interfaz de comandos entre la MV y el usuario que desea administrarla.

Zenoss:

Este software con licencia GPL se encarga de monitorizar el tráfico de red así como de proporcionar alarmas programables mediante mensajes SNMP y de reaccionar a situaciones que puedan ocurrir en la red.

Lo veremos con más detalle en el capítulo 2 y 4.

Wireshark:

Es un popular sniffer que utilizaremos para escanear la red en busca de posibles funcionamientos incorrectos o conductas no permitidas entre los usuarios.

Lo veremos con más detalle en los próximos capítulos.

Etherape:

Monitor de red gráfico que cumple la misma función que Wireshark pero de forma más visual.

IPTables:

Firewall de Linux que tiene también funciones de NAT y PAT.

Otras tecnologías:

Navegadores web Firefox y Iceweasel, Putty, JAVA, VNC, y servidores de X.

1.5 Introducción a la virtualización [41]

La virtualización, en informática, consiste en abstraer recursos procedentes del hardware de la máquina física. Así podemos obtener una versión virtual de unos recursos que no tienen por qué ser los reales y sobre los que podemos instalar cualquier software, incluido un sistema operativo, de la misma forma que lo haríamos sobre una máquina real. Finalmente obtenemos una interfaz externa diferente a la del hardware real con lo que conseguimos ocultar detalles técnicos al usuario, entre otras cosas.

La virtualización de plataforma consiste en un programa que se encarga de simular un medio físico (aunque es virtual) con los recursos parciales de la máquina física, donde poder instalar un sistema operativo.

Vamos a hablar de los tipos diferentes de virtualización de plataforma:

- **Virtualización completa**

La máquina virtual muestra un hardware sobre el cual es posible instalar un sistema operativo sin modificarlo previamente, es decir, diseñado para el mismo procesador. Es posible crear varias máquinas virtuales de este tipo en la misma máquina física.

Aquí es donde podemos encontrar programas como los que vamos a utilizar para nuestro proyecto. Ejemplo: Openvz

- **Virtualización parcial**

La máquina virtual simula parte de los recursos hardware pero no del espacio de direcciones, permitiendo así compartir recursos y procesos. No es posible tener más de un sistema operativo invitado con este método.

- **Virtualización por S.O.**

Buscamos instalar un S.O dentro de otro. El servidor donde tenemos instalador el S.O. que conoce el Hardware real, puede acoger varios sistemas operativos diferentes instalados sobre varios entornos virtuales. Todo el hardware del servidor se encuentra virtualizado con el fin de poder asignar los recursos que deseemos a las MVs.

1.5.1 Ventajas e inconvenientes de virtualizar un sistema operativo

Instalando un sistema operativo virtualizado sobre otro físico estamos instalando dos sistemas operativos en el mismo ordenador que, de otra forma, sólo sería posible tenerlos instalándolos en diferentes particiones y utilizando un gestor de arranque. Esto puede ser una mala idea si necesitamos cambiar a menudo de sistema operativo ya que nos obliga a parar la máquina y volverla a encender. De la otra forma podemos estar trabajando con programas para diferentes SOs al mismo tiempo y de forma inmediata.

La virtualización reduce los costes de espacio físico y de consumo eléctrico, aísla los fallos ya que, si un SO virtualizado da problemas no afectará al resto del sistema, ahorro en piezas de hardware, es posible migrar las máquinas virtuales en caliente ahorrando tiempo en la pérdida de servicio además de evitar servidores ociosos o congestionados.

Por el contrario la opción de la virtualización nos proporciona un SO menos potente que uno instalado sobre el hardware real.

1.5.2 Programas interesantes

De pago podemos encontrar como mejor opción a VMWare mientras que de forma gratuita tenemos como referentes a Xen, OpenVZ y VirtualBox.

1.5.3 Tipos de Virtualización

- **Por Hardware**

Algunos procesadores (los modernos) incorporan modificaciones que aceleran la virtualización. Se añade un anillo más de privilegio (llamado anillo interior o -1) que utilizará el monitor de máquina virtual para aislar las capas superiores de software de las instrucciones que ejecuta la virtualización.

- **Virtualización de almacenamiento**

Los medios físicos de almacenamiento se agregan a una piscina de almacenamiento de la cual se crean almacenamientos lógicos.

- **Particionado**

División de un recurso en varios más pequeños y fáciles de manejar.

- **Máquina virtual**

Es un dispositivo virtual que se crea con unas características propias de CPU, tarjetas de red, discos... y que realmente obtienen sus recursos, que compartirán con el resto de máquinas virtuales, de un sistema físico real.

1.5.4 Nuestro caso

Debido a las exigencias del sistema que vamos a desarrollar, necesitaremos que los alumnos dispongan de máquinas virtuales con diferentes sistemas operativos, las cuales se alojarán en diferentes servidores físicos (varias por cada servidor), hemos optado por la virtualización completa. Además, ya que los procesadores de los nodos secundarios lo permiten, vamos a utilizar virtualización por hardware mejorando así el rendimiento de los sistemas operativos invitados.

1.5.5 Paravirtualización

La paravirtualización se diferencia de la virtualización común por ser la única en el que el sistema operativo que se está virtualizando es consciente de que está siendo virtualizado y por tanto aprovecha este conocimiento para combinarse, mediante una interfaz proporcionada por el programa paravirtualizador, con el sistema operativo anfitrión, de tal forma que, puede enviarle unas tareas específica a éste, las cuales, si se ejecutan en el anfitrión se harán más rápidamente debido a que son mucho más complejas de ejecutar en un sistema virtualizado que en uno real.

1.6 Glosario de términos

Agente SNMP: Es un software que se instala en un equipo para que proporcione información, del sistema donde está instalado, de forma remota.

Appliance o Plantilla[28]: El nombre o calificativo de appliance se aplica a cualquier sistema que se vende como «listo para ser usado». Presentado como una caja negra, en la que el aplicativo está preinstalado.

BackUp: Copia de seguridad.

Bit[29]: Bit es el acrónimo de Binary digit (dígito binario). Un bit es un dígito del sistema de numeración binario.

Byte: Conjunto de ocho bits

Certificado de seguridad[30]: Forma de garantizar que los sitios web y otros servicios de Internet, utilizando cifrado, son realmente quienes dicen ser.

Clúster[31]: El término clúster se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora.

Contenedor: Espacio en disco preparado para albergar una máquina virtual. Puede tener una serie de atributos que definan el tipo de MV.

Dominio[32]: Es un conjunto de ordenadores conectados en una red que confían a uno de los equipos de dicha red la administración de los usuarios.

Escalable[33]: Propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para extender el margen de operaciones sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida.

Firewall: Uno de los elementos encargados de la seguridad de un sistema. Se encarga de dejar pasar conexiones deseadas y de rechazar las no permitidas.

Gigabit Ethernet[34]: Ampliación del estándar Ethernet que consigue una capacidad de transmisión de 1 gigabit por segundo.

IPTables: Firewall incluido en el núcleo de Linux.

Imagen ISO[35]: Imagen ISO es un archivo donde se almacena una copia o imagen exacta de un sistema de ficheros.

Monitorizar: Observar o medir un proceso con el fin de detectar un funcionamiento incorrecto.

Máquina virtual[36]: Software que emula a un ordenador y puede ejecutar programas como si fuese un ordenador real.

Núcleo de un SO: Parte principal del sistema operativo.

Octeto: Byte.

Paquete: Unidad mínima de transporte de la red. Está formado por los datos que se desean enviar y una cabecera que dice cómo y por dónde deben enviarse.

Parche: Modificación de un programa con el fin de reparar algún error o extender sus funcionalidades.

Partición: División del disco duro en discos duros virtuales.

Plugin: Código que aporta una función nueva a una aplicación.

Router: Dispositivo hardware encargado de encaminar paquetes por las diferentes redes.

Servidor: Ordenador que da un servicio a otros ordenadores clientes.

Sistema distribuido[37]: Conjunto de ordenadores conectados entre sí de forma que la ejecución de aplicaciones en uno de ellos no depende de donde esté situada la misma.

Sniffer: Programa utilizado para ver los paquetes que circulan por la red y los datos de su interior.

Software[38]: Equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas; en contraposición a los componentes físicos del sistema, llamados hardware.

Swapping: Es el nombre que se le da al proceso que sucede cuando la memoria RAM de un ordenador está llena y la información se empieza a guardar en el disco duro. Este suceso ralentiza mucho el sistema.

Switch: Dispositivo hardware encargado de encaminar paquetes dentro de una misma red local.

Template: Es una plantilla. En términos de informática se entiende por template una imagen preparada para ser cargada y funcionar con varias funciones previamente configuradas.

VNC[39]: VNC es un programa de software libre basado en una estructura cliente-servidor el cual nos permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente.

VPN: O Red Privada Virtual, es una tecnología que permite a un usuario estar dentro de una red local, aunque la conexión atravesase una red pública, gracias a un túnel de datos.

Virtualización[40]: Capa abstracta que permite que múltiples máquinas virtuales con sistemas operativos heterogéneos puedan ejecutarse individualmente, aunque en la misma máquina física.

Capítulo 2 ANTECEDENTES

2.1 Software de administración de servidores y máquinas virtuales

Dado que este proyecto consta de una gran parte dedicada a la investigación con el fin de encontrar la solución más acertada para el problema que se nos propone, este capítulo, Antecedentes, nos ocupará un tiempo considerable de redacción y unas cuantas páginas de este documento. El otro porcentaje importante del escrito lo podremos encontrar en el cuarto capítulo, que está dedicado al diseño de nuestro producto.

Lo que andamos buscando es una herramienta versátil que nos ofrezca el máximo de posibilidades y combinaciones, gratuita, muy extendida (lo que nos permitirá encontrar mayor información sobre qué cosas puede hacer y cómo las puede hacer, además de ser más compatible), documentada, de fácil manejo, flexible, con multitud de opciones,...

Antes de decidirnos por la opción de utilizar ProxMox como software para administrar nuestro sistema, nos informamos sobre otras soluciones, igualmente válidas en un principio, que fuimos descartando por unas u otras razones que describiremos a continuación:

2.1.1 Ovirt [1]



Ésta es una de las alternativas por ser un programa de visualización y administración de máquinas virtuales, que es lo que buscamos. Para controlarlo podemos usar su interfaz web, conectando al puerto 443, como viene siendo habitual en este tipo de software. Para la virtualización usa KVM (que viene incluido en el núcleo de Linux y requiere que el procesador tenga incluida la virtualización por hardware) y aunque en su web prometen que pronto podrá integrarse también Xen en forma de parche, de momento no tenemos esa posibilidad. Esto ya es una limitación, ya que Xen nos interesa, entre otras cosas, por su capacidad de paravirtualización, es decir, que el sistema operativo huésped o invitado es consciente de que está siendo virtualizado. Este conocimiento tiene como ventaja un incremento de la velocidad. Aunque también es cierto que Ovirt ha evolucionado desde que empezamos con este proyecto hasta hoy y quizá, ahora, sea una opción más interesante que entonces.

2.1.2 OpenQRM[2]

Ésta fue una herramienta a la que dedicamos bastante tiempo. OpenQRM es otra alternativa para la gestión de máquinas virtuales en diferentes máquinas físicas o servidores. Una de las ventajas que posee es la enorme cantidad de cosas que es capaz de hacer. Tal vez demasiadas para este tipo de proyecto.

Para las pruebas con OpenQRM utilizamos una máquina constituida por un procesador Atom a 1,6 Ghz, 1 GB de memoria RAM y 80GB de capacidad de disco duro.

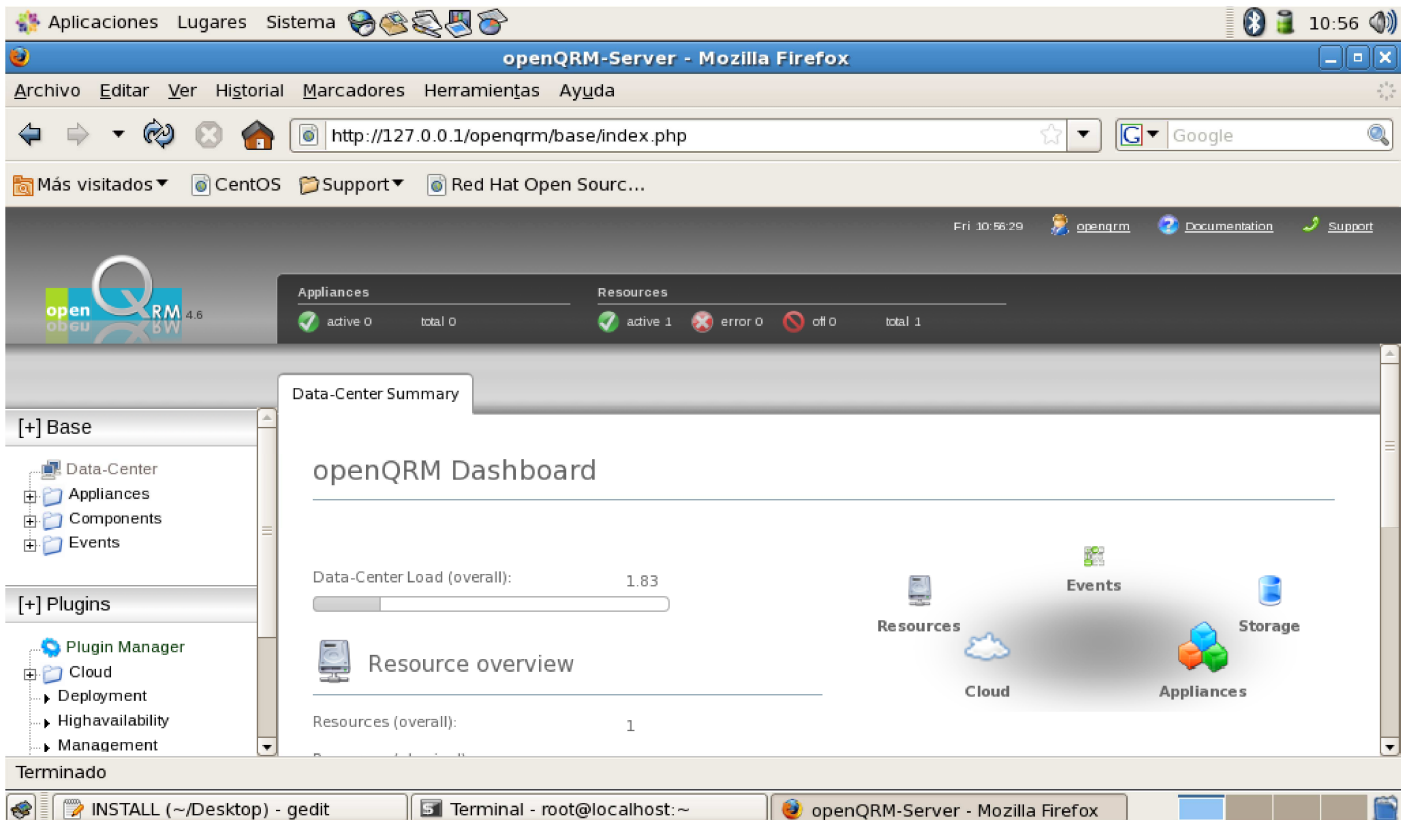
Instalación [7]

- 1- Empezamos por insertar el DVD de instalación de CentOS en el lector y dar formato al disco duro. Instalamos CentOS, configuramos la red y descargamos OpenQRM Server (A partir de la versión 6 no es necesario descargar los plugins, cómo FTP, DHCP,DNS,..., individualmente, sino que ya vienen integrados todos en un paquete “plugins”) para CentOS i386 de la página oficial.
- 2- Lo primero que descubrimos es que nos hace falta MySQL u otra base de datos (Lo instalaremos con “yum install mysqlserver”), ya que OpenQRM va a necesitar de éste elemento para almacenar archivos como la propia información de las máquinas virtuales.
- 3- Una vez configurado, en nuestro caso MySQL, ya podemos instalar OpenQRM. Escribimos yum install openqrm* (dentro del directorio donde hemos descargado El paquete de OpenQRM y el de los plugins) en la línea de comandos y vemos que el paquete de plugins no está firmado, por tanto, CentOS no nos va dejar instalarlo. Solución: editar el fichero “etc/yum.conf” y cambiamos el valor de la variable gpgcheck por un 0.[3] Guardaremos los mensajes que muestran por pantalla en la instalación para una posterior consulta.
- 4- Si todo ha ido bien y después de iniciar la base de datos con la orden `/etc/rc.d/init.d/mysql start`, podremos iniciar el servidor web de OpenQRM con la orden `etc/init.d/openqrm-server start` (también disponemos de la orden `stop` para pararlo y `restart` que sustituye a `stop + start`) con lo que ya podremos acceder a la interfaz de administración web desde un explorador.
- 5- En nuestro caso usaremos Firefox. En la línea de direcciones (URLs) escribiremos `http://127.0.0.1/openqrm`, se nos pedirá nombre de usuario y contraseña, que por defecto es `openqrm` para ambos campos.



The screenshot shows a CentOS desktop environment with a Firefox browser window open to `http://127.0.0.1/openqrm`. A dialog box titled "Identificación requerida" (Required Identification) is overlaid on the browser. The dialog contains the following text: "http://127.0.0.1 está solicitando un nombre de usuario y una contraseña. El sitio dice: 'openQRM-Server Login'". Below this text are two input fields: "Nombre de usuario:" with the value "openqrm" and "Contraseña:" with masked characters ".....". At the bottom right of the dialog are two buttons: "Cancelar" (Cancel) and "Aceptar" (Accept). The background shows the CentOS logo and the text "The Community Welcome" and "CentOS is an Enterprise Linux distribution...".

- 6- La primera vez que entramos tendremos que configurar los últimos detalles como la base de datos. Nótese que, en principio, si el firewall lo permite y hemos configurado correctamente la dirección IP en la instalación de OpenQRM podremos entrar a esta interfaz remotamente desde otros ordenadores que se encuentren en la red, a diferencia de ahora que lo estamos haciendo en modo local (sustituyendo 127.0.0.1 ó *localhost* por la dirección IP que hayamos configurado durante la instalación de OpenQRM).



- 7- A partir de aquí la información y documentación que podemos encontrar, tanto en la página oficial como en otras webs relacionadas, es bastante escasa, lo cual nos hará avanzar con extrema lentitud y finalmente abandonar OpenQRM como opción.

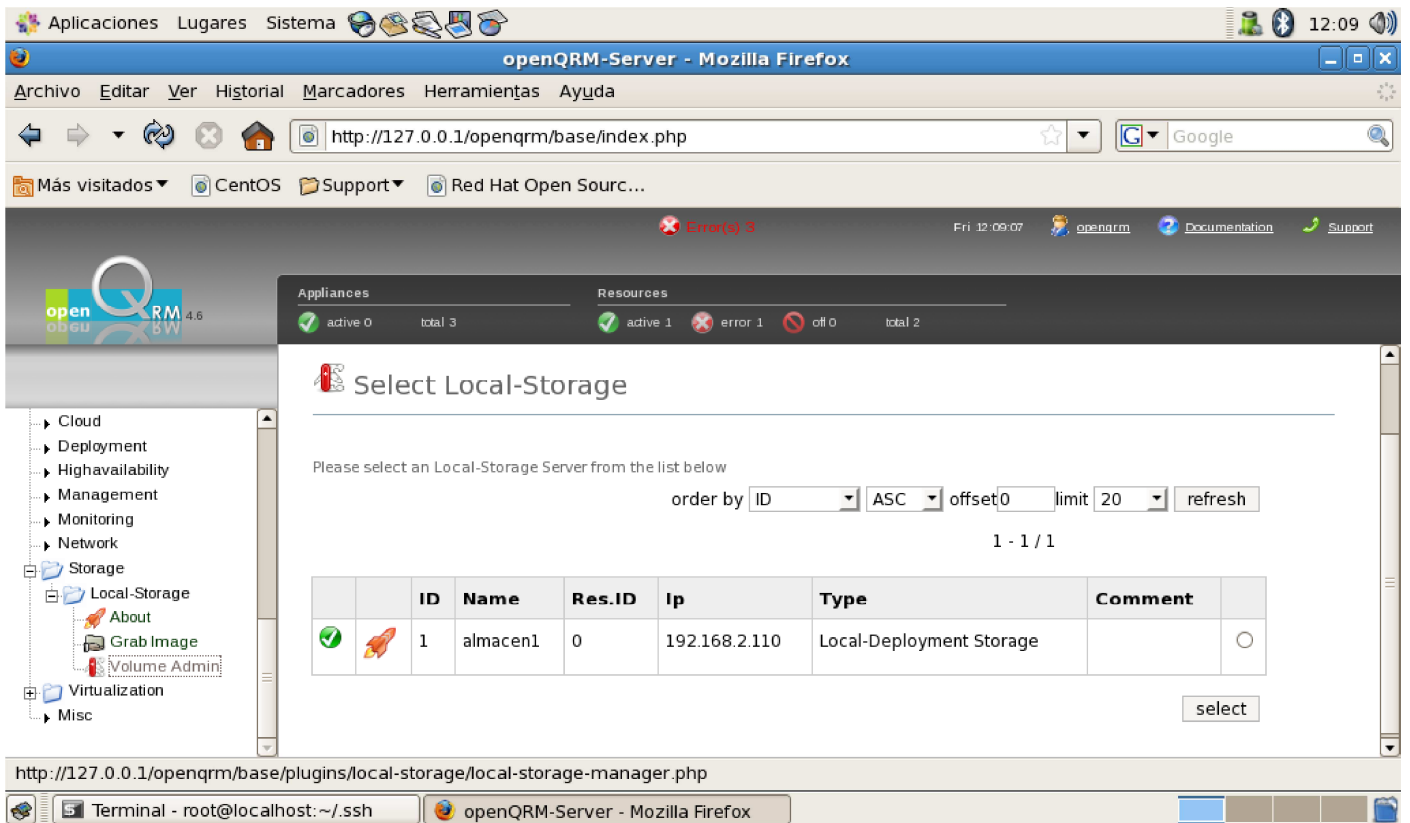
Aun así montamos un ordenador para que forme parte de nuestra red y poder empezar a virtualizar imágenes en él. Se trata de un Pentium 4 con 1GB de memoria RAM y 80GB de disco duro que hemos podido construir de retales. Así pues el ordenador que estábamos usando será el servidor, desde donde pretendemos controlar el estado de los nodos y las máquinas virtuales que en ellos se ejecutan. El Pentium 4 será el primero de los nodos.

- 8- Instalamos CentOS sobre el nuevo ordenador, configuramos el equipo con nombre de Host "nodo1" y dominio "prueba" (nodo1.prueba). Configuramos la red: El servidor tendrá la IP privada 192.168.1.2 mientras que al nodo1 le asignamos la 192.168.1.3 con máscaras de 24 bits. La puerta de enlace será el router con la IP 192.168.1.1 que estará conectado a un switch al cual están conectados ambos ordenadores. El router y el switch están integrados en el mismo dispositivo para uso personal de la marca Edimax. Después de hacer un "ping" vemos que, efectivamente, hay conexión entre ellos.

Creando espacio para una máquina virtual

Vamos a ver los pasos a seguir para llevar a cabo esta tarea:

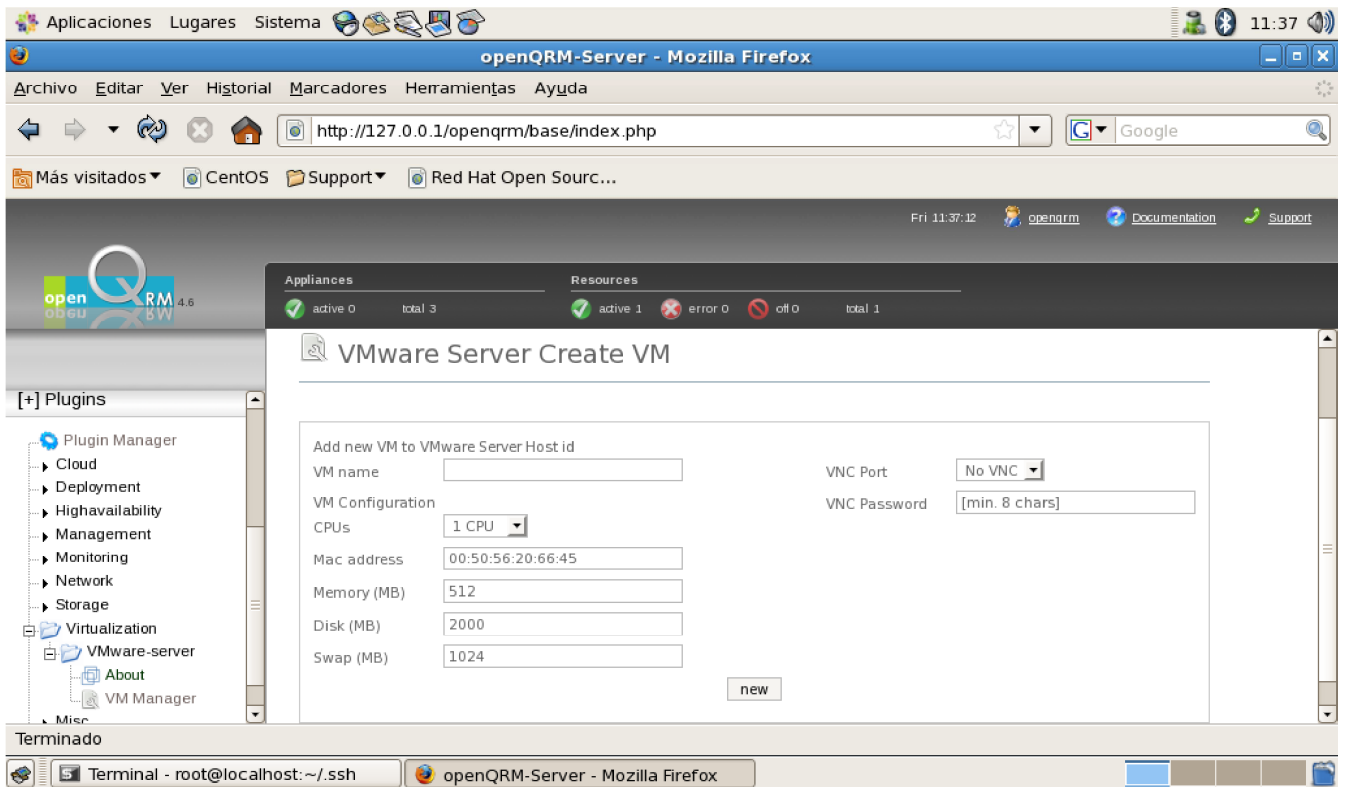
- 1- Dentro de la interfaz web de OpenQRM podemos encontrar un menú a la izquierda donde buscaremos la entrada Plugins/Plugin Manager. Seleccionamos VMWare para añadirlo, ya que esa es la opción que vamos a elegir, en esta primera prueba, como software para la virtualización. Pulsamos en “enable” y después en “Start”.
- 2- Ahora podemos ir a la sección Appliances del menú, y elegir la opción “Create”, elegimos el tipo openQRM Server y ya podremos seleccionar las características de las MV que podrán instalarse en este contenedor, tales como Nombre, número de CPUs,... También debemos de marcar como recurso la opción VMWare –Server Host. [4]
- 3- A continuación debemos crear un almacén donde dejar las imágenes que queremos montar en nuestras máquinas virtuales. En la pestaña “Base” del menú de la izquierda de la interfaz web de OpenQRM seleccionamos el directorio “Components/Create/Storage” para ver los plugins de almacenes que tenemos. En caso de no tener ninguno iremos al administrador de plugins y añadiremos una de las opciones que nos da OpenQRM. En mi caso “Local Storage”, es decir, guardaré las imágenes en el disco duro local del servidor OpenQRM.
- 4- Si volvemos a Storage veremos que ahora sí podemos seleccionar “Local Storage”. Seleccionamos de nuevo OpenQRM Server, click sobre “select” y ya podemos darle nombre y descripción a nuestro nuevo almacén. Guardamos y ya podemos verlo en “Components/storage”.
- 5- Lo siguiente es crear un nuevo volumen lógico en el almacén (podemos verlo si ejecutamos la orden `df -h` en el terminal) vamos a “Components/Create/Image” y seleccionamos una imagen o “virtual appliance”* (podemos descargar un paquete con imágenes de [5]) para llenar el volumen lógico que hemos creado.



*Nota: las virtual appliances se crean en el siguiente paso, Administración de las máquinas virtuales.

Administración de las máquinas virtuales

Para poder cargar una ISO o una “template” en nuestra MV o elegir las características de los recursos de ésta, deberemos dirigirnos, en el menú de la izquierda, a la pestaña Plugins donde veremos que la opción “virtualización” se puede seleccionar gracias a haber instalado el plugin de VMWare. Hacemos click en “VMWare-server” y en VM Manager. Seleccionamos el contenedor que queramos de los que hayamos creado previamente. Ahora si podemos darle nombre a la MV, número de CPUs que va a tener, la dirección MAC, Cantidad de Memoria RAM,...



Cambiar la dirección IP del servidor OpenQRM

Es interesante mencionar que cuando haya un cambio en nuestra topología de red (ya sea a nivel físico o lógico) cambiar la dirección IP que hayamos asignado a OpenQRM, durante la instalación, no es una tarea trivial. Si bien, es cierto, no deberíamos tener que cambiar la dirección de nuestro servidor muy a menudo. Las dos formas de proceder son las siguientes:

Opción primera:

Ya que la información de configuración básica de OpenQRM no se destruye aunque se desinstale el programa, si quisiéramos desinstalarlo y volverlo a instalar con el fin de que nos volviese a pedir una dirección IP el propio programa, no funcionaría. La solución pasa por borrar (o modificar) el archivo donde se almacena esa configuración. El nombre de ese archivo podemos obtenerlo de los mensajes que muestra OpenQRM por pantalla cuando se está instalando y será diferente en función del sistema operativo sobre el que instalemos, el árbol de directorios,...

Además, hay que recordar que si reinstalamos OpenQRM deberemos borrar las claves aprendidas de los archivos `"/root/.ssh/known_hosts"` y `"/var/lib/nxserver/home/.ssh/known_hosts"` de lo contrario no coincidirían con las guardadas en la anterior instalación y jamás funcionaría.

Opción segunda:

La otra forma de hacerlo es también la más elegante.

- 1- Tendremos que abrir el archivo `/usr/lib/openqrm/tftpboot/pxelinux.cfg/default` con un editor de texto (vi, kate, emacs,...) y cambiar la dirección IP manualmente.
- 2- Después iniciaremos MySQL ejecutando el comando `./mysqld start` en el directorio `/etc/rc.d/init.d/`.
- 3- Ahora entramos en el Shell de MySQL:

```
#mysql -u root
```

- 4- Elegimos la base de datos:

```
>use openqrm
```

- 5- Cambiamos el valor de la entrada para la dirección IP:

```
>UPDATE `resource_info`
```

```
SET `resource_openqrm server`=`192.168.x.x`,`resource_ip`=192.168.x.x`
```

```
WHERE `resorce_info_info`.`resource_id`=0
```

```
LIMIT 1;
```

- 6- Luego sólo nos quedará reiniciar OpenQRM.

Abandono de OpenQRM

Debido a la complejidad del programa y a la ausencia de la documentación necesaria para hacerlo funcionar en nuestro sistema y por otros motivos como que la gestión de las máquinas virtuales está separada del programa principal (a diferencia de ProxMox que sí que las incluye mediante Xen y OpenVirtuozzo) y la cantidad de fallos que genera y que no son fáciles de solventar con la pobre documentación de la página oficial y los pocos textos que se pueden encontrar en la red a fecha de hoy, finalmente acabaremos por abandonar la idea de utilizar OpenQRM para nuestro proyecto.

2.1.3 ProxMox VE [6]

Después de instalar y probar ProxMox hemos decidido, debido a la simplicidad de su interfaz, a la excelente documentación y a su fácil implantación y gestión de las MVs, que es la mejor opción para emplear en este sistema. Lo veremos con más detalle en el apartado de “Diseño”.

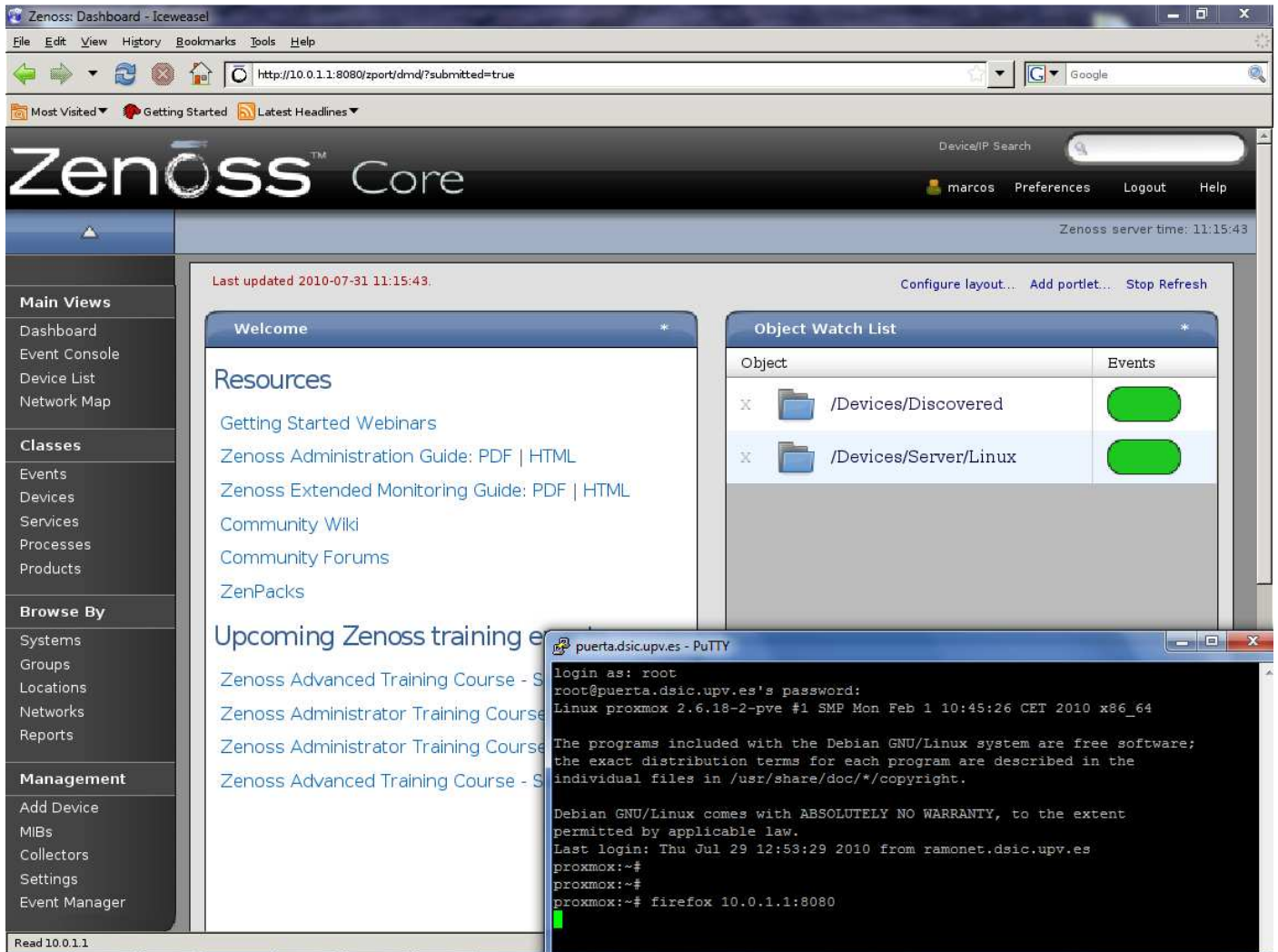
2.2 Software de monitorización de la red

2.2.1 Zenoss [8]

En un principio, y tras leer la descripción de Zenoss que dan en su página web oficial como monitor de red, pensamos que Zenoss era una buena solución para observar el tráfico que circula por la red de nuestro sistema y detectar comportamientos y comunicaciones “ilegales” entre los dispositivos dentro de nuestro entorno.

El propio ProxMox, desde su interfaz web, tiene un enlace para descargar una plantilla de Zenoss, concretamente es un sistema operativo básico de Ubuntu 8.04 con un Zenoss 2.5.1 preinstalado.

- 1- Una vez creada la máquina virtual en el nodo principal (puerta.dsic.upv.es) (veremos cómo se hace en el apartado de Diseño) arrancamos Zenoss.
- 2- Hecho esto tendremos un servidor web escuchando en el puerto 8080 de nuestra MV con Zenoss. Realizamos una conexión ssh al nodo principal (hemos utilizado Putty con la opción para soportar el entorno de ventanas X11 habilitada y el servidor de Xs gratuito VcXsrv).
- 3- Ejecutamos la orden “firefox direcciónIPDeLaMVDeZenoss:8080”. Si hemos redireccionado bien las Xs deberíamos entrar en el modo de configuración de Zenoss y acabado esto en su interfaz web de configuración.



Para añadir los nodos de nuestra red y poder monitorizar el tráfico que pasa por sus interfaces de red tendremos que hacer dos cosas por cada nodo:

- 1- Instalaremos un agente SNMP en cada nodo [9]. Para ello ejecutaremos las siguientes instrucciones en cada uno de ellos:
 - `apt-get update` Actualizamos los repositorios
 - `apt-get install snmpd` Instalamos el agente SNMP
 - `nano /etc/snmp/snmpd.conf` Modificamos el fichero para que nos quede como en la siguiente imagen:


```

puerta.dsic.upv.es - PuTTY
GNU nano 2.0.7 File: /etc/snmp/snmpd.conf

# NETWORK (EG: 10.10.10.0/24), and read/write access to only the
# localhost (127.0.0.1, not its real ipaddress).
#
# For more information, read the FAQ as well as the snmpd.conf(5)
# manual page.

####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming
# from):

#      sec.name  source          community
#com2sec paranoid default         public
com2sec readonly default         public
#com2sec readwrite default         private

com2sec networkstation5 10.0.0.0/8 public

#####
# Second, map the security names into group names:

#      sec.model  sec.name
group MyROSystem v1      paranoid
group MyROSystem v2c     paranoid
group MyROSystem usm     paranoid
group MyROGroup v1      readonly
group MyROGroup v2c     readonly
group MyROGroup usm     readonly
group MyRWGroup v1      readwrite
group MyRWGroup v2c     readwrite
group MyRWGroup usm     readwrite
group MyNR5Group v1     networkstation5
group MyNR5Group v2c    networkstation5
group MyNR5Group usm    networkstation5

####
# Third, create a view for us to let the groups have rights to:

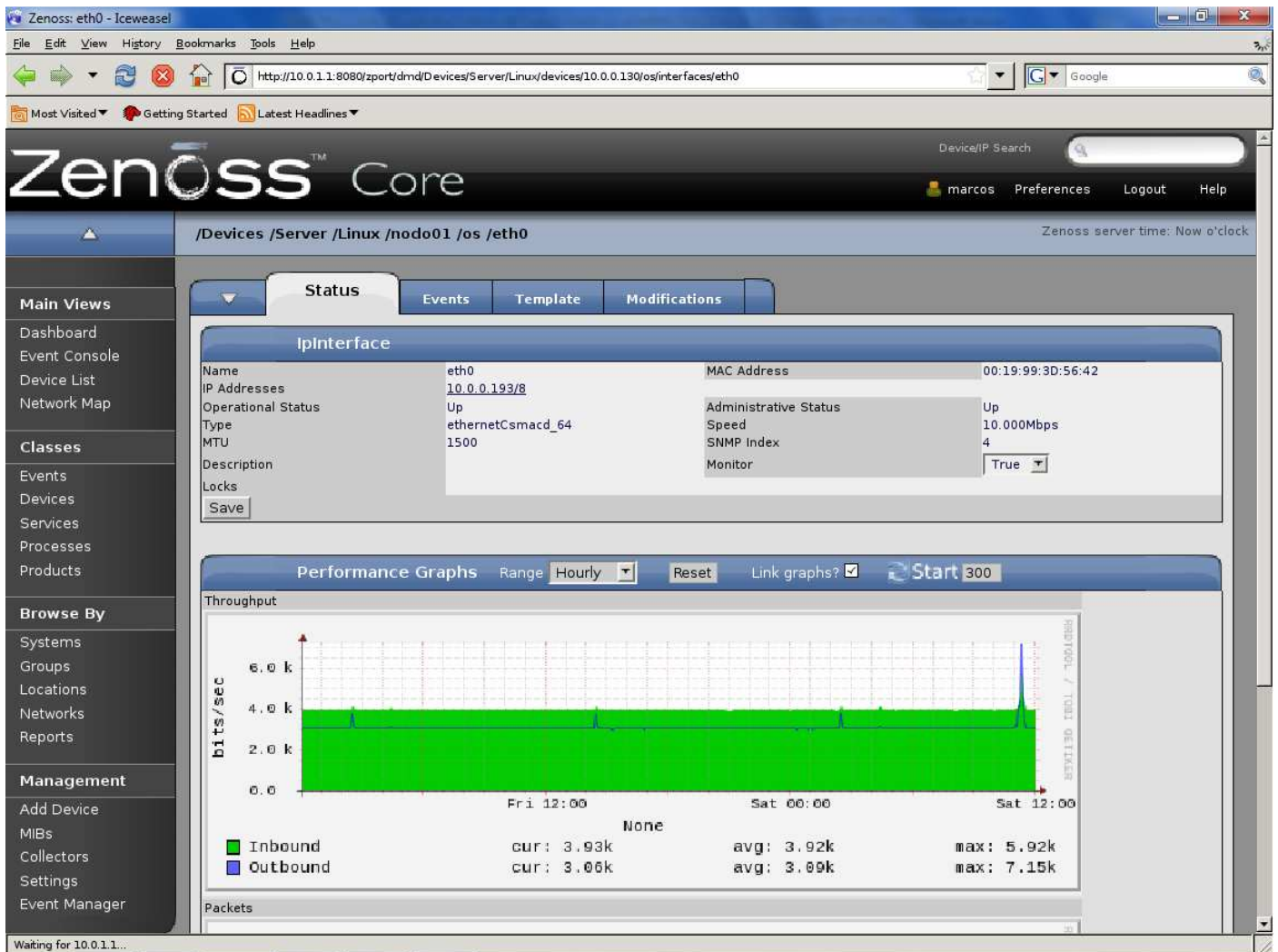
#      incl/excl subtree          mask
view all    included  .1          80
view system included  .iso.org.dod.internet.mgmt.mib-2.system

^G Get Help   ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit       ^J Justify   ^W Where Is  ^V Next Page  ^U UnCut Text ^T To Spell
    
```

- `nano /etc/default/snmpd` Borramos la dirección de loopback (127.0.0.1)

- 2- Añadir el dispositivo a Zenoss. Buscamos la etiqueta “Add Device” en el menú de la izquierda de la interfaz web de Zenoss y ponemos la dirección IP del nodo o dispositivo que queremos agregar en la casilla “nombre”, seguidamente seleccionamos el tipo de dispositivo y lo añadimos.[10]

Para ver la cantidad de tráfico que circula por una interfaz de red de nuestro equipo, nos dirigiremos a la opción “Device List”, seleccionamos un nodo de la lista, click en la pestaña “OS” y seleccionamos la interfaz que queremos monitorizar.



Zenoss también nos proporciona la opción de programar una serie de alarmas que mediante mensajes SNMP nos pueden avisar de situaciones que ocurren en nuestra red así como tomar decisiones al respecto de forma automática. Pero esto nos obliga a definir esas reglas, por tanto no soluciona el problema que buscamos de asegurarnos de que no nos dejamos nada por controlar con IPTables, ya que debemos configurar las mismas reglas para Zenoss. Además también queremos ver el origen y destino de los paquetes, que circulan por nuestra red, en un momento dado para poder

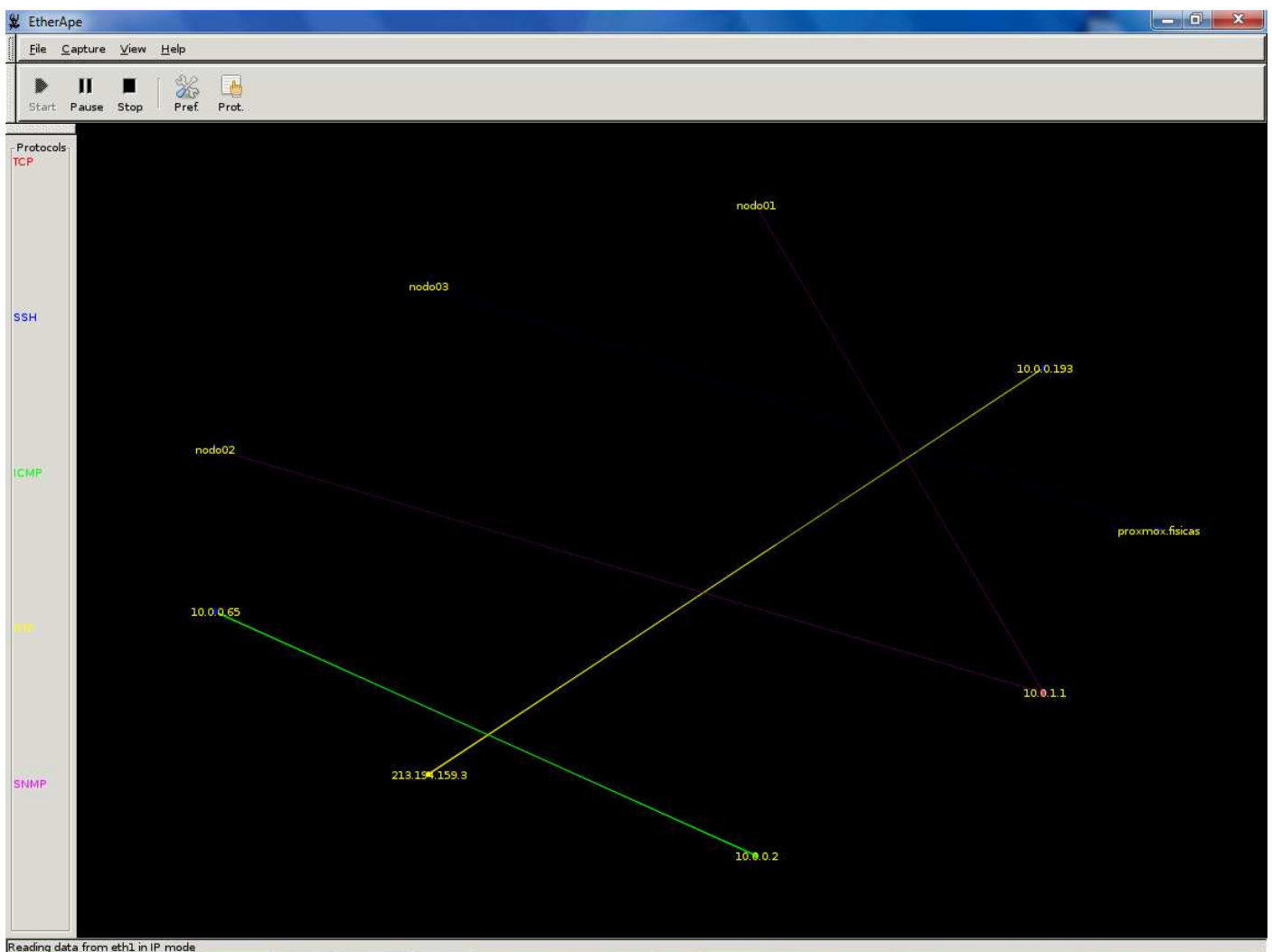
buscar conexiones incorrectas. Zenoss no nos proporciona esta característica de “Sniffer” ni ninguno de sus plugins o zenpacks.

Por estas razones nos vemos obligados a buscar otras soluciones para garantizar la seguridad dentro de nuestra red.

2.2.2 EtherApe[11]

EtherApe es un programa de monitorización de redes gráfico para Unix. Con el podemos ver las conexiones que existen en una red así como los protocolos en uso en esas conversaciones.

En la imagen podemos ver como se ha ejecutado la orden “ping” desde una máquina virtual contra el nodo01 al mostrarse de color verde los mensajes ICMP. Al mismo tiempo vemos, en color violeta, como se envían mensajes SNMP desde el nodo01 y el nodo02 a Zenoss que está instalado en una máquina virtual dentro del nodo01 y tiene como IP la 10.0.1.1. También podemos ver mensajes de sincronización de relojes entre un servidor NTP y uno de nuestros nodos.



Como veremos más adelante, Proxmox crea una interfaz lógica llamada venet0 en cada uno de los nodos del clúster donde nuestras máquinas virtuales envían y reciben la información que desean que atraviese la red. Por ello debemos configurar EtherApe para que no muestre la información que pasa por la interfaz Venet0 en cada uno de los nodos. Esto nos permitiría ver el tráfico entre las máquinas virtuales de los alumnos y detectar cualquier anomalía o fallo de seguridad.

El problema de este sistema es que deberíamos abrir una ventana por nodo, ya que este software no permite redireccionar todo el tráfico a una misma ventana, por tanto es poco escalable. La idea de condensar todo lo que pase en nuestra red para poder darnos cuenta de cualquier problema de un vistazo es lo que nos llevará a probar otras soluciones como Wireshark y T-shark.

2.2.3 Wireshark [12] y Tshark [13]

Ambos se encargan de capturar los paquetes que circulan por una red y mostrar tanto la cabecera como el contenido de los datos del paquete. La diferencia entre estos dos Sniffers es la posibilidad del primero de utilizar el modo gráfico.

Hemos realizado varios experimentos con estos programas. La primera opción es a su vez la más fácil de implementar.

- 1- Instalamos Wireshark en todos los nodos y los ejecutamos escuchando en la interfaz virtual Venet0.
- 2- Después solo hay que realizar conexiones ssh con la posibilidad de utilizar las Xs (El modo gráfico) activado. Esta solución nos proporciona prácticamente lo mismo que usar EtherApe y no resuelve ninguno de los problemas que presentaba.

Después de esto podemos llegar a la conclusión de que lo que necesitamos es un programa capaz de monitorizar remotamente interfaces en otras máquinas reales. Wireshark es capaz de hacer esto utilizando demonios (Pcapd [14]) en los nodos pero solo una interfaz al mismo tiempo, es decir, seguimos teniendo que abrir una ventana por nodo a monitorizar.

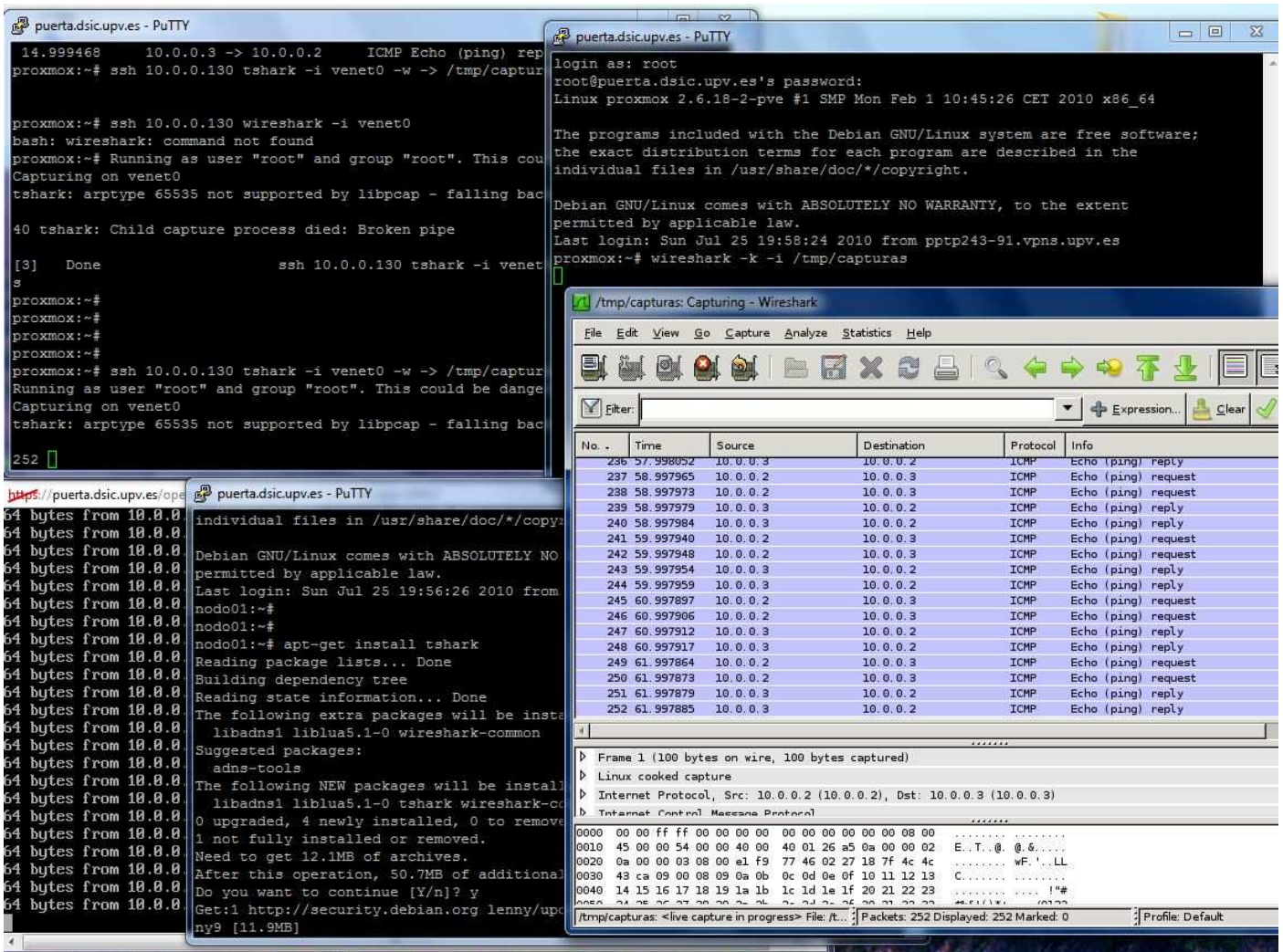
Esto nos lleva al siguiente planteamiento: tenemos que monitorizar remotamente varios nodos y recibir la información centralizada para poder leerla fusionada en un único programa y ventana. Los siguientes pasos a realizar son:

- 1- Instalaremos Tshark en todos los nodos y Wireshark en el nodo principal (En ProxMox).
- 2- Una vez arrancado Tshark en los nodos creamos un túnel FIFO para poder abrir con Wireshark localmente la información que se genera con Tshark remotamente.

Para ello deberemos introducir las siguientes órdenes el nodo principal[15]:

- 1- Creamos el túnel

- `mkfifo /tmp/capturas`
- 2- Iniciamos Tshark en el host remoto escuchando en la interfaz Venet0 y redireccionando la salida al túnel
 - `ssh hostremoto tshark -i venet0 -w -> /tmp/capturas &`
- 3- Leemos las capturas con Wireshark
 - `wireshark -k -i /tmp/capturas`



- 4- Por último generamos tráfico por medio de mensajes ICMP para comprobar su funcionamiento. Ahora podemos leer de la interfaz Venet0 de cualquier nodo pero todavía no podemos verlas todas a la vez.

¿Cómo hacemos que Wireshark lea lo que capturan varios tshark al mismo tiempo?

- 5- Pronto nos damos cuenta de que no podemos escribir en un mismo túnel desde varios nodos porque la salida del túnel se vuelve ininteligible. Por tanto el paso final será guardar la salida de cada uno de los Tsharks en un fichero de texto, después fusionar los archivos en uno sólo y abrir el fichero de capturas con Wireshark. El problema que surge es que a la hora de fusionar los archivos podría haber paquetes que no se hubieran escrito completamente obligando a Wireshark a detenerse en esos paquetes “rotos”. La solución pasa por poner un máximo de captura de paquetes y esperar a que se complete. Después ya podemos fusionarlos.

Nota: Los relojes de todos los servidores, como en cualquier sistema distribuido, deberán estar sincronizados con el fin de que, al fusionar las capturas, el orden de los paquetes no sea alterado y las conversaciones tengan sentido.

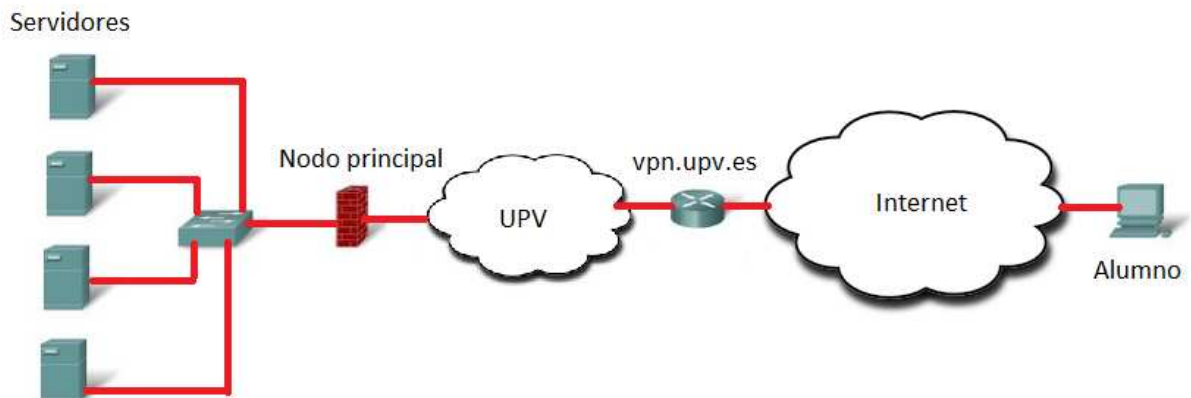
Capítulo 3 ANÁLISIS Y DESCRIPCIÓN DEL PRODUCTO

El fin de este proyecto es el de crear un laboratorio virtual, en el cual, los alumnos puedan realizar sus prácticas conectándose a sus máquinas desde cualquier parte del mundo y a cualquier hora del día (o de la noche). El hecho de utilizar la virtualización nos va a permitir proporcionar una serie de recursos a los alumnos y realizar un tipo de prácticas que no sería posible abarcar con máquinas físicas.

El sistema está constituido por una serie de servidores potentes que se encargan de alojar las máquinas virtuales. Estos nodos están conectados a un switch por medio de conexiones Gigabit Ethernet. Hemos conectado un último nodo a este switch, que será el encargado de recibir las conexiones de los alumnos y redireccionar el tráfico a las MVs correspondientes por medio de IPTables.

Este último nodo se encargará de realizar otras tareas como son:

- 1- Hacer de Gateway.
- 2- La monitorización del tráfico de red.
- 3- Almacén de imágenes y plantillas
- 4- Albergar la interfaz web que permitirá a los alumnos administrar sus MVs, una vez se hayan autenticado en el servidor VPN de la UPV.
- 5- Y la función de nodo principal de nuestro clúster ProxMox.



El software ProxMox será el encargado de crear las máquinas virtuales, utilizando las imágenes almacenadas en el disco duro local, por medio de OpenVZ, y enviarlas después a los nodos en los cuales queremos ejecutarlas. Este software también es capaz de darnos información sobre el estado en el que se encuentran las MVs (encendida, apagada, uso de CPU, uso de memoria RAM, disco duro,...) así como proporcionarnos una interfaz desde la que podemos arrancarlas, pararlas, o acceder por VNC a las máquinas, lo que nos da así un control total sobre ellas.

El control de entrada de usuarios se basa en:

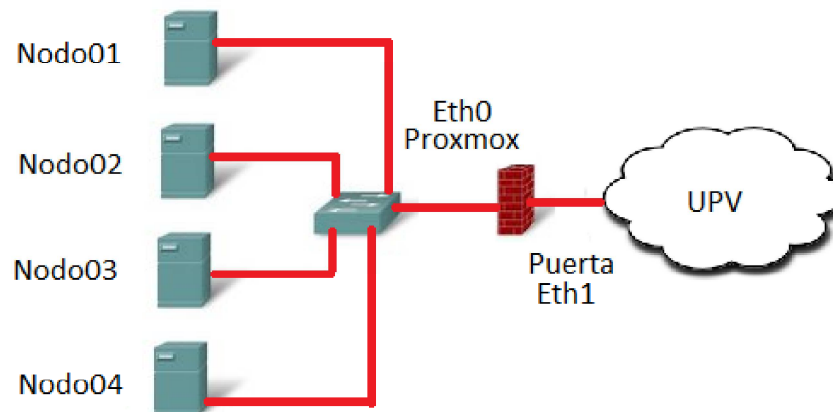
- 1- Una primera fase en la cual los alumnos tendrán que autenticarse contra la VPN de la UPV. Seguidamente intentarán la conexión contra sus MVs donde se comprobará en una base de datos si el alumno está autorizado.
- 2- El otro aspecto importante de la seguridad será proteger el acceso de los alumnos a máquinas virtuales que no les correspondan. Esto lo haremos instalando IPTables en cada uno de los nodos y monitorizando la red con software específico para asegurarnos de que las reglas de acceso son correctas.

La creación y restauración de copias de seguridad de las máquinas virtuales de los usuarios se llevarán a cabo utilizando instrucciones de OpenVZ como veremos en el apartado de “Diseño”.

Habrá que crear una interfaz web donde los alumnos accederán para manipular sus máquinas virtuales. La interfaz ejecutará instrucciones de OpenVZ para realizar las acciones sobre las MVs.

Capítulo 4 DISEÑO

Tras una primera reflexión nos ponemos a trabajar en la implementación del proyecto. Empezamos por abordar la parte en la que nos encargaremos de crear, físicamente, nuestra red, es decir, conectaremos las conexiones entre los dispositivos antes mencionados. Para ello utilizaremos cable de categoría 6 capaz de soportar el estándar Gigabit Ethernet al igual que las tarjetas de red que tenemos en los servidores. Una vez conectados todos los nodos al switch y el nodo principal conectado además a la red interna de la UPV, nuestra topología quedará de la siguiente manera en un primer momento.



Esta composición podría ser fácilmente ampliable añadiendo nodos y switches. Podemos considerar dentro de las necesidades de un ámbito lectivo que el sistema escala bien.

4.1 Asignando direcciones IP y configurando la red

Parece que el paso siguiente debería ser diseñar el reparto de direcciones IPs. Elegimos la subred 10.0.0.0/8 por ser la red privada que más IPs nos permite utilizar al proporcionarnos tres octetos

(más de 16 millones de combinaciones posibles) para darle el uso que nosotros queramos. La distribución de direcciones elegida se muestra en la siguiente figura:



Explicación detallada de cada uno de los segmentos:

- 1- La parte fija equivale a la red privada de clase A 10.0.0.0.
- 2- Los primeros cuatro bits de la parte variable se van a utilizar para diferenciar los diferentes años o cursos. El 0000 equivale al año 2010-2011, el 0001 al 2011-2012 y así sucesivamente.
- 3- Además hemos añadido un campo versión el cual estará a 0 si estamos utilizando una versión del proyecto final o un 1 si hemos cambiado algo. Así pues cada año podríamos tener dos versiones, la normal o una más nueva.
- 4- Los últimos 6 bits se emplean para que cada alumno pueda crear un total de 63 máquinas virtuales (ya que la primera dirección de cada alumno se reserva para utilizarse como direcciones de administración, es decir, se las asignaremos a los nodos, máquinas virtuales de monitorización u otros elementos de nuestra red distintos a las máquinas virtuales de los alumnos).
- 5- Por último, los 13 bits restantes identificarán al alumno. Esto nos permite tener un total de 8192 alumnos matriculados por cada año y versión.

Ejemplo para el alumno X, número de lista 63 (0000000111111) matriculado en el curso 2013-2014, versión normal y tercera máquina virtual que ha creado:

- Binario: 0000 1010.0011 0000.0000 1111.1100 0011
- Decimal: 10.48.15.195

Como la primera dirección de cada alumno se reserva para administración, las direcciones IPs de los elementos del sistema serán:

- Puerta: Interfaz de red del nodo principal por la parte exterior a nuestra red privada. Dirección IP interna de la UPV y por tanto corresponderá con una IP pública de clase B de la forma 158.42.x.x.
- ProxMox: Interfaz de red del nodo principal que se encuentra dentro de nuestra red privada y por tanto le asignaremos la primera de las IPs de Administración: 10.0.0.1
- Nodo01: Segunda IP de administración 10.0.0.65
- Nodo02: 10.0.0.129
- Nodo03: 10.0.0.193
- Nodo04: 10.0.1.1
- Zenoss: Monitor de tráfico de nuestra red: 10.0.1.65

Cómo funciona la red

A continuación procedemos con la explicación del camino que realiza un paquete desde que se genera en una máquina virtual hasta que llega a un servidor externo a nuestra red, como por ejemplo un servidor web de Internet.

Supongamos que queremos consultar una página web desde un navegador. Una vez generada la petición al servidor DNS, externo a nuestra red, el primer sitio dónde va un paquete, generado en una máquina virtual sobre OpenVZ, es a la interfaz virtual `venet0:0` quien, como nos indica la nomenclatura, es una subinterfaz de la interfaz virtual del nodo real que alberga las máquinas virtuales, es decir la `venet0` del nodo.[44]

```
[root@pruebaDeRed ~]# ifconfig
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

venet0     Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
            inet addr:127.0.0.1  P-t-P:127.0.0.1  Bcast:0.0.0.0  Mask:255.255.255.255
            UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
            RX packets:1396 errors:0 dropped:0 overruns:0 frame:0
            TX packets:951 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:107692 (105.1 KiB)  TX bytes:125428 (122.4 KiB)

venet0:0   Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
            inet addr:10.0.0.68  P-t-P:10.0.0.68  Bcast:10.0.0.68  Mask:255.255.255.255
            UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
```

Venet0 no es más que un router virtual, que crea OpenVZ, que lee la cabecera del paquete IP y lo redirige por la interfaz real correspondiente. [43]

Esta interfaz podría ser una interfaz real como eth0 con lo que el paquete ya saldría a la red externa, pero en nuestro caso se redirigirá a la interfaz vmbr0, que crea Proxmox por defecto y que no es más que un switch virtual a través del cual las máquinas virtuales de un mismo nodo pueden comunicarse entre sí como si estuviesen conectadas mediante cables reales, donde tenemos configurada nuestra IP y máscara del nodo físico que alberga la máquina virtual que quiere realizar una conexión externa, así como el Gateway o ruta por defecto donde enviaremos los paquetes dirigidos a redes distintas a la 10.0.0.0/8. Vmbr0 debe conectarse con una interfaz real haciendo así de puente. En nuestro caso, este puente está configurado para reenviar los paquetes a través de la interfaz física eth0, como se muestra en la siguiente imagen. [45]

```
GNU nano 2.0.7 File: interfaces
auto lo
iface lo inet loopback

auto vmbr0
iface vmbr0 inet static
    address 10.0.1.65
    netmask 255.0.0.0
    gateway 10.0.0.1
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
```

Por último el paquete llega a puerta a su interfaz vmbr0, que es la que tenemos determinada como Gateway para los nodos secundarios, y este puente reenvía finalmente el paquete por la interfaz con IP y Gateway externos, en nuestro caso eth0. Ahora el encargado de redirigir el paquete será el router de la UPV que hemos establecido como Gateway y que se encuentra en el mismo laboratorio que nuestros equipos.

```
[root@pruebaDeRed ~]# ping www.google.es
PING www.l.google.com (209.85.227.105) 56(84) bytes of data.
64 bytes from wy-in-f105.1e100.net (209.85.227.105): icmp_seq=1 ttl=49 time=42.3 ms
64 bytes from wy-in-f105.1e100.net (209.85.227.105): icmp_seq=2 ttl=49 time=41.9 ms
64 bytes from wy-in-f105.1e100.net (209.85.227.105): icmp_seq=3 ttl=49 time=42.2 ms
64 bytes from wy-in-f105.1e100.net (209.85.227.105): icmp_seq=4 ttl=49 time=42.1 ms

--- www.l.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 41.990/42.194/42.398/0.254 ms
[root@pruebaDeRed ~]#
```

4.2 Acceso a los elementos del sistema para su configuración

Dado que los sistemas que vamos a manipular se encuentran en una zona restringida y a la cual sólo se puede tener acceso unas horas del día, una vez instalado Proxmox en cada una de las máquinas y

conectada la red de forma física y lógica, todas las pruebas y configuraciones las vamos a realizar remotamente. Para ello el servidor principal (puerta) además de aceptar peticiones en el puerto 443 de ProxMox para conexiones HTTP, también aceptará conexiones SSH en el puerto 22. Al instalar ProxMox, por defecto, se habilita este servicio.

Para poder abrir programas en modo gráfico instalaremos en el PC, desde el cual vamos a trabajar desde casa, un servidor de Xs, en nuestro caso “vcxsrv”[23]. Ahora sólo debemos ejecutar el servidor de X cuando queramos hacer una conexión SSH con “puerta”, y en el software que usemos para la conexión, “Putty” en nuestro caso, habilitar la opción “permitir X11”.

Si queremos administrar otros nodos de nuestra red haremos una segunda conexión SSH desde puerta hacia cualquier otra dirección IP de nuestra red o nombre de host (si hemos establecido la relación host-dirección IP en el DNS) incluidas las máquinas virtuales. Esto lo haremos ejecutando la orden “ssh” en la línea de comandos del terminal.

```
root@virtual1alumno1:~  
login as: root  
root@puerta.dsic.upv.es's password:  
Linux proxmox 2.6.18-2-pve #1 SMP Mon Feb 1 10:45:26 CET 2010 x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Aug 15 19:26:13 2010 from pptp243-22.vpns.upv.es  
proxmox:~# ssh root@nodo01  
Linux nodo01 2.6.18-2-pve #1 SMP Mon Feb 1 10:45:26 CET 2010 x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Aug 15 19:29:56 2010 from 10.0.0.65  
nodo01:~# ssh 10.0.0.2  
The authenticity of host '10.0.0.2 (10.0.0.2)' can't be established.  
RSA key fingerprint is f1:0e:be:83:7c:0e:a8:e7:d0:9c:a4:d7:9e:20:ed:8a.  
Are you sure you want to continue connecting (yes/no)? y  
Please type 'yes' or 'no': yes  
Warning: Permanently added '10.0.0.2' (RSA) to the list of known hosts.  
  
root@10.0.0.2's password:  
Last login: Thu Jul 22 18:09:17 2010 from 10.0.0.65  
[root@virtual1alumno1 ~]#
```

4.3 Instalación de Proxmox

Una vez asignadas las direcciones IPs de la red vamos a comenzar a instalar el software necesario para llevar a cabo el proyecto. El nodo principal será también el nodo principal de nuestro clúster ProxMox.

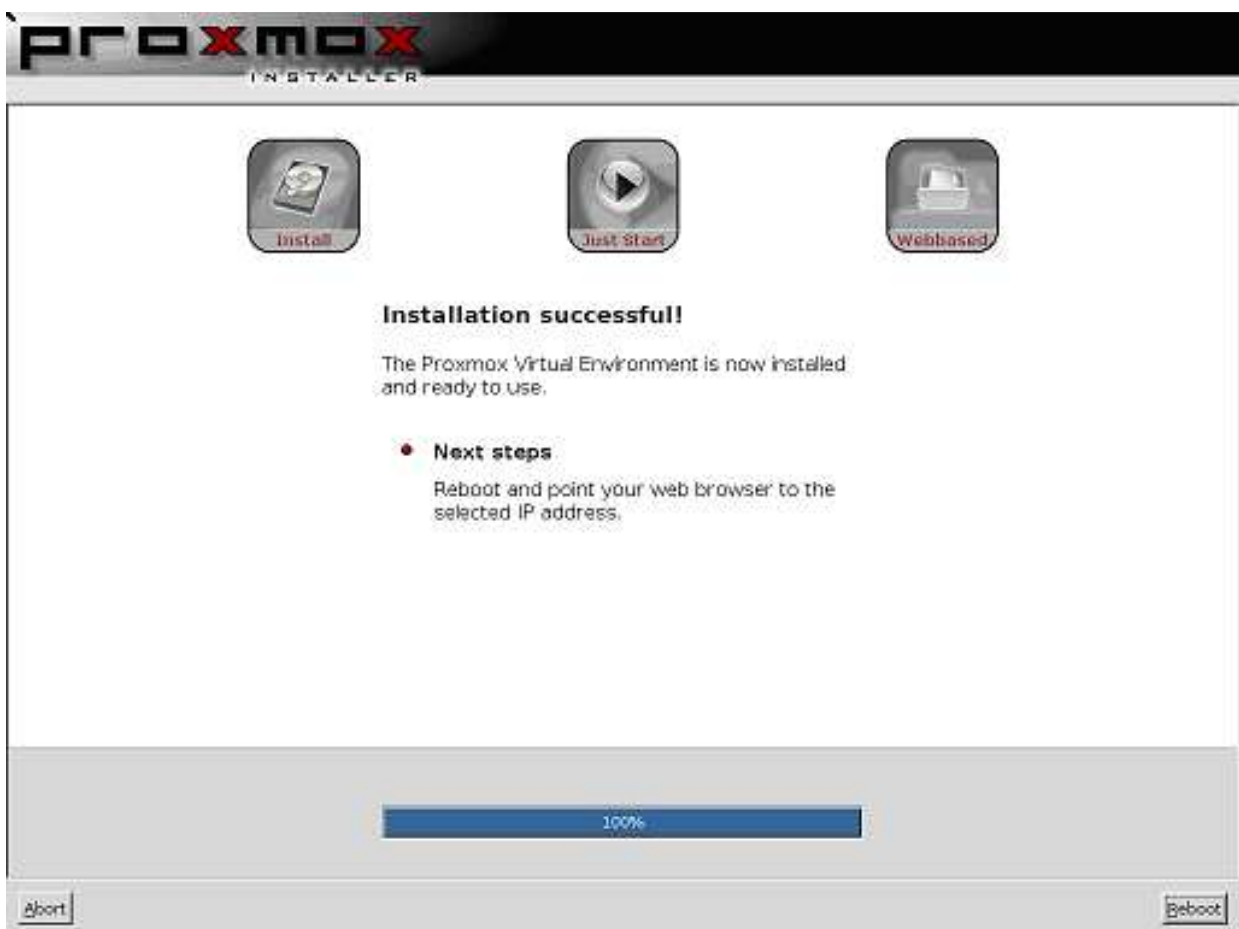
- 1- Empezaremos Instalando ProxMox en el nodo principal. Nos aseguramos de tener una partición con espacio suficiente para instalar sobre ella ProxMox. Descargamos la ISO de su página oficial y la grabamos en un CD. ProxMox sólo se presenta en versiones para procesadores de 64bits.
- 2- Insertamos el CD en el lector del nodo principal, arrancamos desde CD configurándolo previamente y de forma tradicional en la BIOS de la máquina, o pulsando la tecla de función 12 al arrancar (depende de la placa base que estemos utilizando).
- 3- Tras el arranque inicial nos aparecerá una pantalla con el logotipo de ProxMox donde tendremos que pulsar la tecla "Intro/Enter". [16][17]



A continuación se nos pedirá que aceptemos los términos y condiciones de uso, después se nos informará, entre otras cosas, de que se va a proceder con el formateo del disco duro/partición y de que no se nos va a ofrecer la opción de crear un partición nueva para ProxMox. Seguidamente se nos preguntará por la zona horaria así como por el lenguaje del

teclado. Crearemos un usuario (root) y su contraseña e introduciremos un e-mail de contacto para posibles incidencias.

- 4- Ahora introduciremos el nombre del host y su dirección IP, DNS y Puerta de enlace. Estos campos variarán en función del nodo que estemos configurando, así pues, para el nodo principal “puerta/proxmox” su Gateway será el Router perteneciente a la red de la UPV al que estemos conectados, como muchos otros servidores y PCs, mediante el switch correspondiente. Por otro lado, los nodos encargados de alojar las MVs destinadas a prácticas tendrán como Gateway a “puerta/proxmox” mientras que la información correspondiente a los DNS, dominio y máscara de red permanecerá idéntica.
- 5- Después de hacer click en el botón “Next” se formatea la partición y se instala ProxMox seguido de un reinicio del sistema.



- 6- Si todo ha salido bien ya podemos acceder a la interfaz de control de ProxMox desde un explorador web situado en otro ordenador (Ya sea dentro de la red de la UPV o desde fuera una vez autenticados en la VPN) realizando una conexión al puerto seguro 443 de la dirección IP que le hemos asignado.
- 7- Veremos que aparece una pantalla de bienvenida donde debemos autenticarnos (root y la contraseña que acabamos de establecer) y accederemos a todas la opciones que ProxMox nos ofrece. También podemos ver que en nuestro clúster sólo contamos con un nodo, este

mismo, así pues y antes de empezar a manipular las opciones de Proxmox vamos a añadir a nuestro clúster los nodos que deberán ejecutar las MVs.

4.4 Añadiendo nodos secundarios al nodo principal [18]

Esta tarea es muy sencilla y se resume en los siguientes pasos:

- 1- Lo primero que debemos hacer es instalar Proxmox en los nodos secundarios sin olvidar utilizar el mismo nombre de dominio que en el principal.
- 2- Una vez realizado esto escribiremos en la línea de comandos del nodo principal la instrucción:
`proxmox:~# pveca -c`

Hemos creado un servidor principal. Para comprobarlo escribimos a continuación:

```
proxmox:~# pveca -l
```

Que provocará, en nuestro caso, el siguiente resultado:

```
proxmox:~# pveca -l
CID----IPADDRESS----ROLE-STATE-----UPTIME---LOAD---MEM---DISK
 1 : 10.0.0.1          M   A     1 day 10:34  0.26   37%   4%
```

Póngase especial interés a la “M” que aparece en la columna de rol, esto significa que este nodo es el “master”. La letra A de estado indica que el nodo está activo, Podemos ver el identificador (1 en nuestro caso por ser el primero), el tiempo que lleva el nodo encendido así como la carga de CPU, memoria RAM y disco duro.

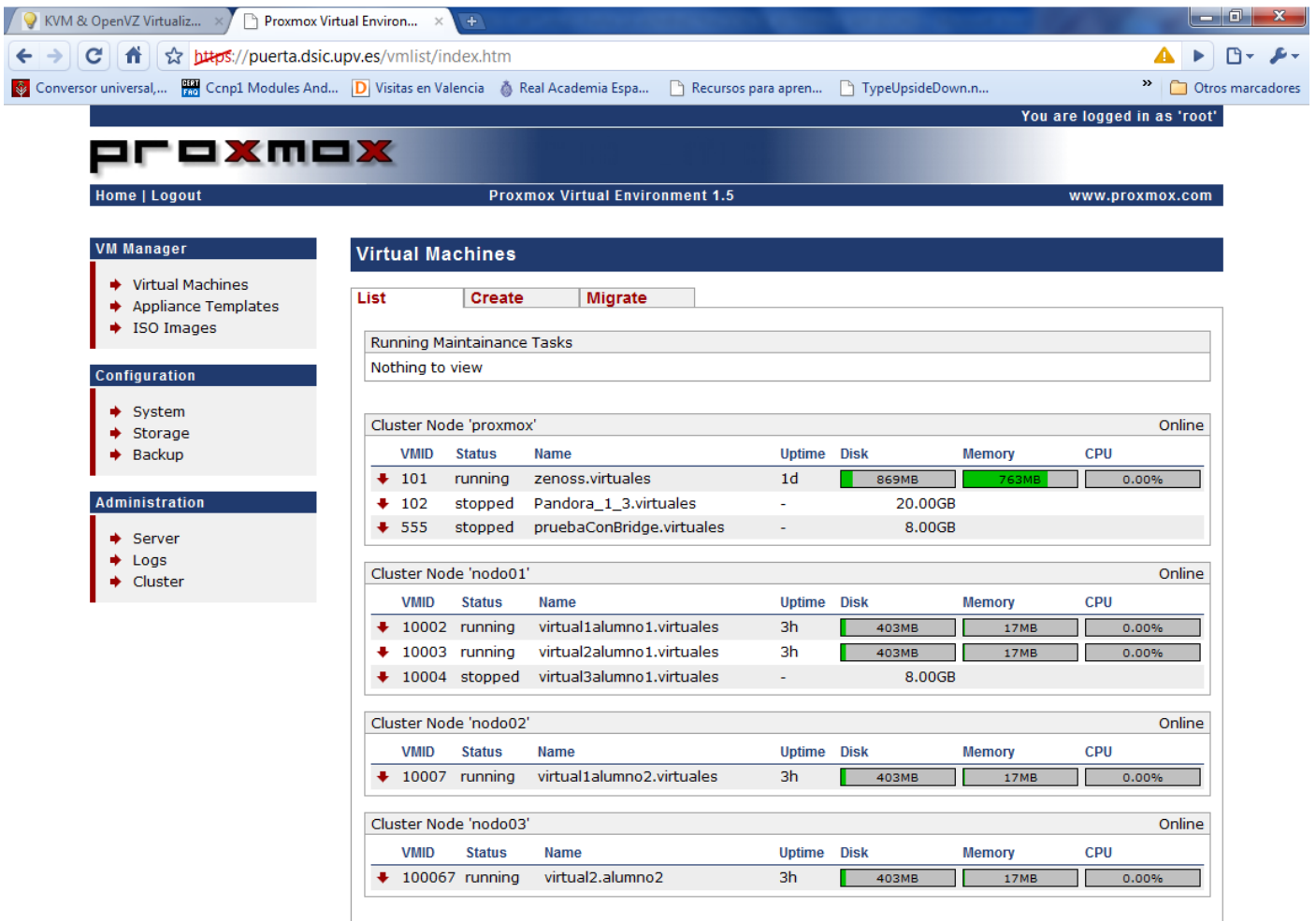
- 3- A continuación nos dirigimos a cada uno de los nodos secundarios y ejecutamos la orden:

```
Nodoxx:~# pveca -a -h 10.0.0.1
```

Donde la dirección IP corresponde a la dirección del nodo principal. Y tras un intercambio de claves podremos volver al nodo “proxmox” para listar de nuevo los nodos pertenecientes al clúster:

```
proxmox:~# pveca -l
CID----IPADDRESS----ROLE-STATE-----UPTIME---LOAD---MEM---DISK
 1 : 10.0.0.1          M   A     1 day 10:40  0.16   37%   4%
 2 : 10.0.0.130       N   A     1 day 10:32  0.00   3%    1%
 3 : 10.0.1.65        N   A     1 day 10:31  0.00   3%    1%
 4 : 10.0.1.129      N   A     1 day 10:30  0.00   3%    1%
```

- 4- Si volvemos a la interfaz gráfica de Proxmox en el nodo principal podremos ver, ahora sí, como aparecen los demás nodos incluidos en el clúster y cómo podemos manipularlos así como las máquinas virtuales que contendrán.



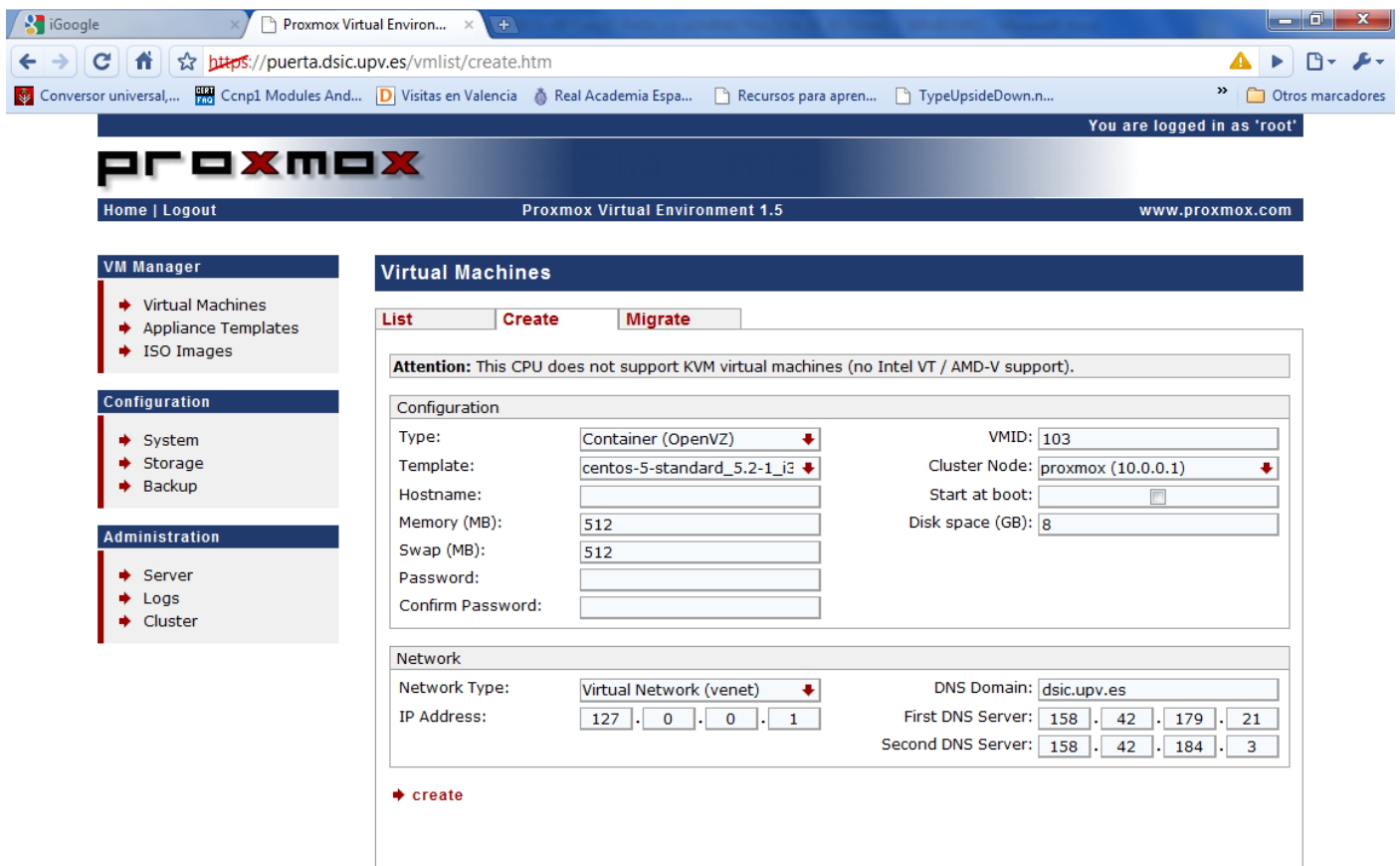
4.5 Crear una máquina virtual con ProxMox

Aunque la creación de máquinas virtuales está pensado que se lleve a cabo por parte de los alumnos a través de la interfaz gráfica que creamos y mediante instrucciones de Open Virtuozzo, es posible que nos sea necesario crear alguna desde la interfaz de ProxMox (como la máquina que contendrá la imagen de Zenoss para el monitoreo de la red) por ello vamos a describir este proceso para máquinas virtuales sobre OpenVZ:

- 1- En primer lugar tendremos que almacenar en local la imagen de la MV que queramos añadir a nuestro sistema. Estas imágenes pueden ser fácilmente descargadas desde muchas páginas web como "HowToForge" o pueden ser creadas a partir de la ISO de un sistema operativo. Además de plantillas también es posible instalar una ISO completa en una máquina virtual creada.
- 2- En el siguiente ejemplo descargaremos una plantilla de CentOS-5 desde la propia interfaz de ProxMox ya que éste nos proporciona una serie de plantillas que podemos descargar desde la pestaña "Downloads" sin necesidad de buscarlas por la web. En caso de descargarlas al disco

duro desde algún otro servidor deberíamos añadirlas a Proxmox desde la pestaña “Local” seguido de pulsar el botón “seleccionar archivo”.

Para ello nos dirigimos a la opción del menú de la izquierda con el nombre de “Appliance Templates” y elegimos la opción “CentOS 5 Standard”. Una vez hecho esto haremos click sobre la opción “Virtual Machines” y seguidamente en la pestaña “create”. Ahora aparecerá una ventana como la que se muestra en la imagen:



The screenshot shows the Proxmox Virtual Environment 1.5 web interface. The browser address bar shows the URL <https://puerta.dsic.upv.es/vmlist/create.htm>. The page title is "Proxmox Virtual Environment 1.5". The user is logged in as 'root'. The main navigation menu on the left includes "VM Manager" (Virtual Machines, Appliance Templates, ISO Images), "Configuration" (System, Storage, Backup), and "Administration" (Server, Logs, Cluster). The "Virtual Machines" section is active, showing a "Create" form. The form has a warning: "Attention: This CPU does not support KVM virtual machines (no Intel VT / AMD-V support)." The form is divided into "Configuration" and "Network" sections. The "Configuration" section includes fields for Type (Container (OpenVZ)), Template (centos-5-standard_5.2-1_i386), VMID (103), Cluster Node (proxmox (10.0.0.1)), Hostname, Memory (MB) (512), Swap (MB) (512), Password, and Confirm Password. The "Network" section includes fields for Network Type (Virtual Network (venet)), IP Address (127.0.0.1), DNS Domain (dsic.upv.es), First DNS Server (158.42.179.21), and Second DNS Server (158.42.184.3). A "create" button is located at the bottom of the form.

El primer campo, “Type”, hace referencia al tipo de MV que vamos a crear, en nuestro caso Open Virtuozzo será el software de virtualización empleado.

- 3- Elegimos Centos 5 como el sistema operativo que queremos virtualizar, le damos un nombre que la identifique que será el nombre del alumno seguido del número de su máquina, un punto y el dominio (Alumno1Virtual1.virtuales para el ejemplo) Seleccionamos la cantidad de memoria RAM que destinaremos del nodo real a la MV y la cantidad de memoria de disco duro que se destinará a Swapping en el disco duro si el consumo de memoria fuese superior al disponible. Un password para garantizar el acceso privado del usuario y un número de MV que lo haremos coincidir con la IP que se le asigna a la máquina para su pronta identificación.

A continuación podemos elegir si la MV se arrancará automáticamente cuando se inicie el sistema o si, por el contrario, no lo hará. La variable “Cluster Node” nos permite seleccionar el nodo en el cual queremos que se cree la máquina virtual, aunque también sería posible migrarla fácilmente después de crearla haciendo uso de la pestaña “Migrate”, desde donde podemos elegir la MV que queremos migrar, independientemente del nodo en el que se encuentre, y el nodo al cual irá la máquina. Reservamos espacio en disco para el SO y los datos y pasamos a la configuración de los ajustes de red.

- De los dos tipos de red el que más nos va a interesar es el primero, es decir, “virtual Network” o “venet” ya que nos permitirá tener un punto donde colocar un sniffer o monitor del tráfico para ver los paquetes que envían y reciben las máquinas virtuales además de permitirnos asignarles una IP a éstas. También tiene otras ventajas que podemos ver en la siguiente tabla extraída de [19]:

Differences between veth and venet

Feature	veth	venet
MAC address	Yes	No
Broadcasts inside CT	Yes	No
Traffic sniffing	Yes	No
Network security	Low	High
Can be used in bridges	Yes	No
Performance	Fast	Fastest

Nota: Es necesario explicar que en la tabla anterior, donde se dice que no se puede esnifar el tráfico se refiere a que se creará otra interfaz distinta de eth0 para las máquinas virtuales, llamada venet0, que es donde se podrá ver los paquetes.

- Sólo falta configurar el dominio de donde se encuentran los servidores DNS así como la dirección IP del primario y del secundario (si se desea). Finalmente haremos click en “create” y en un corto período de tiempo podremos ver nuestra MV en la pestaña “List” de “virtual machines”.
- Al hacer click sobre el nombre de la MV accederemos a un submenú desde el cual podemos arrancar o parar la máquina además de ver el uso de los recursos que está haciendo, comprobar sus atributos e incluso cambiar algunos de ellos como su dirección IP o el nombre del host. También podremos ver la opción “OpenVNC console” desde la cual podremos acceder a la MV mediante una sesión VNC (siempre que tengamos instalado JAVA en el navegador web). Además podríamos acceder a la MV mediante una sesión ssh conectando al puerto 22 e identificándonos como “root” y con la clave escrita anteriormente.

Nota: Si quisiéramos crear máquinas virtuales para KVM, el procesador de todos los nodos, especialmente el del nodo principal, debe soportar virtualización por hardware.

4.6 Otras funciones de ProxMox

Otras opciones interesantes que ProxMox nos ofrece son las siguientes:

- 1- Haciendo click en la etiqueta "Server" dentro de submenú "Administration" vemos que se nos muestran todos los servidores que hay, habilitados o no, dentro de nuestro ProxMox. Servidores con el web, el SSH, SMTP, NTP... desde aquí podemos arrancarlos o pararlos según nos convenga con un simple click.
- 2- Vemos que este submenú cuenta también con otra pestaña con nombre "Certificates" dónde podemos descargar e instalar certificados en nuestro servidor para establecer conexiones seguras, por ejemplo, al conectar al servidor 443 y no ser víctimas de un engaño con fines como el de conseguir nuestra contraseña y ataques de *"Man in the middle"*.
- 3- La siguiente pestaña que encontramos es "Logs" que, como su nombre indica, guarda una entrada para cada evento que ocurre en cualquiera de los nodos del clúster.
- 4- "Cluster" nos da información bastante completa sobre el estado de cada uno de los servidores: direcciones IP, rol, estado (activado o desactivado), tiempo que lleva activo el nodo, porcentaje de procesos que esperan para entrada/salida [27], uso de CPU, memoria y disco.
- 5- Dentro del submenú "Configuration" encontramos más opciones interesantes. Empezando por la etiqueta "System" donde podemos ver la configuración de las interfaces de red, la configuración de los DNS (así como modificarla), la configuración del Network Time Protocol, cambiar la clave o el e-mail de contacto, o el idioma de la aplicación.
- 6- Si entramos ahora en "Storage" podemos ver los almacenes disponibles, en nuestro caso sólo local, y haciendo click sobre el podemos ver las imágenes y plantillas que tenemos almacenadas. También podemos crear varios tipos de almacenes si hacemos click en la flecha roja que se encuentra al lado del nombre de la sección.
- 7- Por último tenemos el botón BackUps que, como su nombre indica, es donde podemos configurar copias de seguridad del sistema haciendo click en la flecha roja y siempre que hayamos creado un almacén para copias de seguridad en "Storage".

4.7 Crear y administrar una máquina virtual con instrucciones de OpenVZ

Ya que uno de los propósitos es el de crear una interfaz para los usuarios desde la cual puedan administrar sus MVs, debemos de encontrar una forma mediante la cual nuestro programa se comunique con las MVs para obtener y enviar información a las mismas. Para ello hemos decidido utilizar las órdenes que ofrece OpenVZ de forma que cada selección que haga el alumno se transformará en una simple orden dirigida al software de virtualización.

Las instrucciones serán las siguientes:

- 1- Consultar el estado de las máquinas virtuales del alumno “alumno1” previamente identificado en la VPN de la UPV:

```
nodo01:~# vzlist -a | grep alumno1
 10002      18 running  10.0.0.2      virtual1alumno1.virtuales
 10003      18 running  10.0.0.3      virtual2alumno1.virtuales
 10004      - stopped  10.0.0.4      virtual3alumno1.virtuales
nodo01:~#
```

En el diseño definitivo “alumno1” será sustituido por una variable que contendrá el nombre del alumno. Con el resultado obtenido ya sólo es necesario formatear la salida que recibimos de OpenVZ en nuestra interfaz gráfica.

- 2- Para crear una máquina virtual el alumno deberá elegir una de las posibles máquinas que el profesor desea que puedan ser creadas. Tras ello se enviará una orden con el formato:

“vzctl create Identificador –ostemplate nombre del archivo de la imagen a virtualizar –ipadd dirección IP del rango del alumno –hostname nombre de la máquina”

Y se informará al alumno del resultado de la operación.

- 3- Para cambiar el estado de una máquina en concreto tras seleccionar el alumno, en nuestra interfaz, la máquina que desea manipular y hacer click sobre el botón que contenga la palabra “arrancar, parar, reiniciar”, la instrucción que deberemos introducir será:

“vzctl start Identificador de la máquina”

Y lo mismo para la opción stop o restart. También podemos consultar el estado actual de la máquina con

“vzctl status Identificador”:

```
nodo01:~# vzctl status 10003
CTID 10003 exist mounted running
nodo01:~#
```

- 4- Para eliminar una máquina seguiremos el procedimiento inmediatamente anterior cambiando la palabra “status” por “destroy”
- 5- OpenVZ permite otras opciones, como por ejemplo cambiar los atributos de la MV mediante la orden “set”, pero no van a ser necesarios para nuestro trabajo.

4.8 Copias de seguridad de las máquinas virtuales

También será necesario que el alumno tenga la opción de guardar el estado de su máquina virtual mediante copias de seguridad. Esto le permitirá probar nuevas modificaciones sin arriesgar el trabajo ya hecho. Para llevar a cabo esta tarea vamos a usar el comando “VzDump” que consiste en una utilidad para la administración de copias de seguridad de máquinas virtuales de OpenVz.

Para realizar la copia de seguridad de una MV deberemos:

- 1- Ofrecer en nuestra interfaz gráfica la opción de seleccionar el número de MV de la cual queremos hacer el backup.
- 2- Una vez hecho esto enviaremos la orden siguiente:

```
“vzdump Identificador de la máquina virtual”
```

Con ello crearemos la copia de seguridad que se almacenará en “/var/lib/vz/dump/vzdump-openvz-Identificador de la MV seguido de la fecha y la hora”. También se creará un registro (“log”) en ese mismo directorio que podremos consultar por si queremos ofrecer más información al usuario.

- 3- Finalmente deberíamos proporcionar como salida el nombre de la CS al alumno para que más tarde sea capaz de restaurarla.

Es interesante mencionar que con el fin de que el disco duro del servidor no se llene debido a la creación desmesurada de copias, es posible seleccionar el número máximo de copias que vamos a permitir almacenar. Esto lo haremos modificando en el archivo “/etc/vzdump.conf” la variable “maxfiles: n” donde n es el número de backups máximo o con la siguiente instrucción:

```
“vzdump –maxfiles n”
```

En esta segunda opción podemos añadir el identificador de la máquina, con lo cual cada MV tendría un máximo (igual o distinto) y un alumno no podría acaparar todos los espacios disponibles.

VzDump tiene otras opciones interesantes como la capacidad de comprimir las copias con `--compress` entre otras.


```

proxmox:~# vzdump 101
INFO: starting new backup job: vzdump 101
INFO: Starting Backup of VM 101 (openvz)
INFO: CTID 101 exist mounted running
INFO: status = CTID 101 exist mounted running
INFO: mode failure - unable to dump into snapshot (use option --dumpdir)
INFO: trying 'suspend' mode instead
INFO: backup mode: suspend
INFO: bandwidth limit: 10240 KB/s
INFO: starting first sync /var/lib/vz/private/101/ to /var/lib/vz/dump/vzdump-op
envz-101-2010_08_15-19_37_21.tmp
INFO: Number of files: 36598
INFO: Number of files transferred: 31433
INFO: Total file size: 809923717 bytes
INFO: Total transferred file size: 807356528 bytes
INFO: Literal data: 807356528 bytes
INFO: Matched data: 0 bytes
INFO: File list size: 769582
INFO: File list generation time: 0.001 seconds
INFO: File list transfer time: 0.000 seconds
INFO: Total bytes sent: 809590506
INFO: Total bytes received: 631507
INFO: sent 809590506 bytes received 631507 bytes 8061910.58 bytes/sec
INFO: total size is 809923717 speedup is 1.00
INFO: first sync finished (101 seconds)
INFO: suspend vm
INFO: Setting up checkpoint...
INFO: suspend...
INFO: get context...
INFO: Checkpointing completed succesfully
INFO: starting final sync /var/lib/vz/private/101/ to /var/lib/vz/dump/vzdump-op
envz-101-2010_08_15-19_37_21.tmp
INFO: Number of files: 36598
INFO: Number of files transferred: 15
INFO: Total file size: 809923871 bytes
INFO: Total transferred file size: 38507947 bytes
INFO: Literal data: 72015 bytes
INFO: Matched data: 38435932 bytes
INFO: File list size: 769582
INFO: File list generation time: 0.001 seconds
INFO: File list transfer time: 0.000 seconds
INFO: Total bytes sent: 895510
INFO: Total bytes received: 78512
INFO: sent 895510 bytes received 78512 bytes 216449.33 bytes/sec
INFO: total size is 809923871 speedup is 831.53
INFO: final sync finished (4 seconds)
INFO: resume vm
INFO: Resuming...
INFO: vm is online again after 5 seconds
INFO: creating archive '/var/lib/vz/dump/vzdump-openvz-101-2010_08_15-19_37_21.t
ar'
INFO: Total bytes written: 839424000 (801MiB, 11MiB/s)
INFO: archive file size: 800MB
INFO: delete old backup '/var/lib/vz/dump/vzdump-openvz-101-2010_08_15-19_31_39.
tar'
INFO: Finished Backup of VM 101 (00:03:08)
INFO: Backup job finished successfully

```

Para restaurar una máquina virtual utilizaremos la orden vzrestore cuya sintaxis es la siguiente:

“vzrestore nombreDelArchivoQueContieneLaCopia identificadorDeLaMáquina”

Nota: Para restaurar una máquina virtual debemos eliminar previamente la máquina con el ID que queremos restaurar o de lo contrario nos obligará a seleccionar un ID nuevo para la MV que se cree

al restaurar. Esto es fácil hacerlo de forma transparente al usuario si antes de enviar la orden “vzrestore” ejecutamos una instrucción “vzctl destroy”

Si queremos más opciones podemos usar también la orden “qmrestore”, pero esta alternativa no resulta más interesante que la anterior para este proyecto.

4.9 Ejecución de máquinas virtuales creadas en Linux en sistemas Windows

Se ha considerado la opción de que los alumnos puedan llevarse a casa archivos de imagen de las máquinas virtuales para poder trabajar con ellas de forma local, lo cual parece más rápido y cómodo que hacerlo remotamente, sobre todo si la red está congestionada.

Para llevar a cabo esta tarea es necesario realizar sobre el archivo de imagen una serie de pasos con el fin de que ésta pueda ser ejecuta y manipulada en un sistema Windows. Tras varias pruebas con diferente software de diferentes compañías y varias horas infructuosas nos decidimos por la opción de utilizar la utilidad para convertir imágenes de Qemu: “qemu-img”.

Con este comando podemos cambiar el formato de imágenes creadas con KVM, VMWare,... (lo cual resume varios problemas en una única solución) y abrir estas nuevas imágenes en un Windows en el que tengamos instalado Qemu (Para Windows) [20][21].

Para ello haremos:

- 1- Instalar Qemu (el cual viene con qemu-img) con la orden:

```
“apt-get install qemu”
```

Lo haremos en un sistema Linux donde esté la MV que queremos migrar a Windows.

- 2- Una vez hecho esto sólo tenemos que seleccionar la imagen que queremos cambiar de formato y la nueva extensión que ésta tendrá.

Ejemplo para convertir una imagen de VMWare a una imagen que pueda abrirse con qemu:

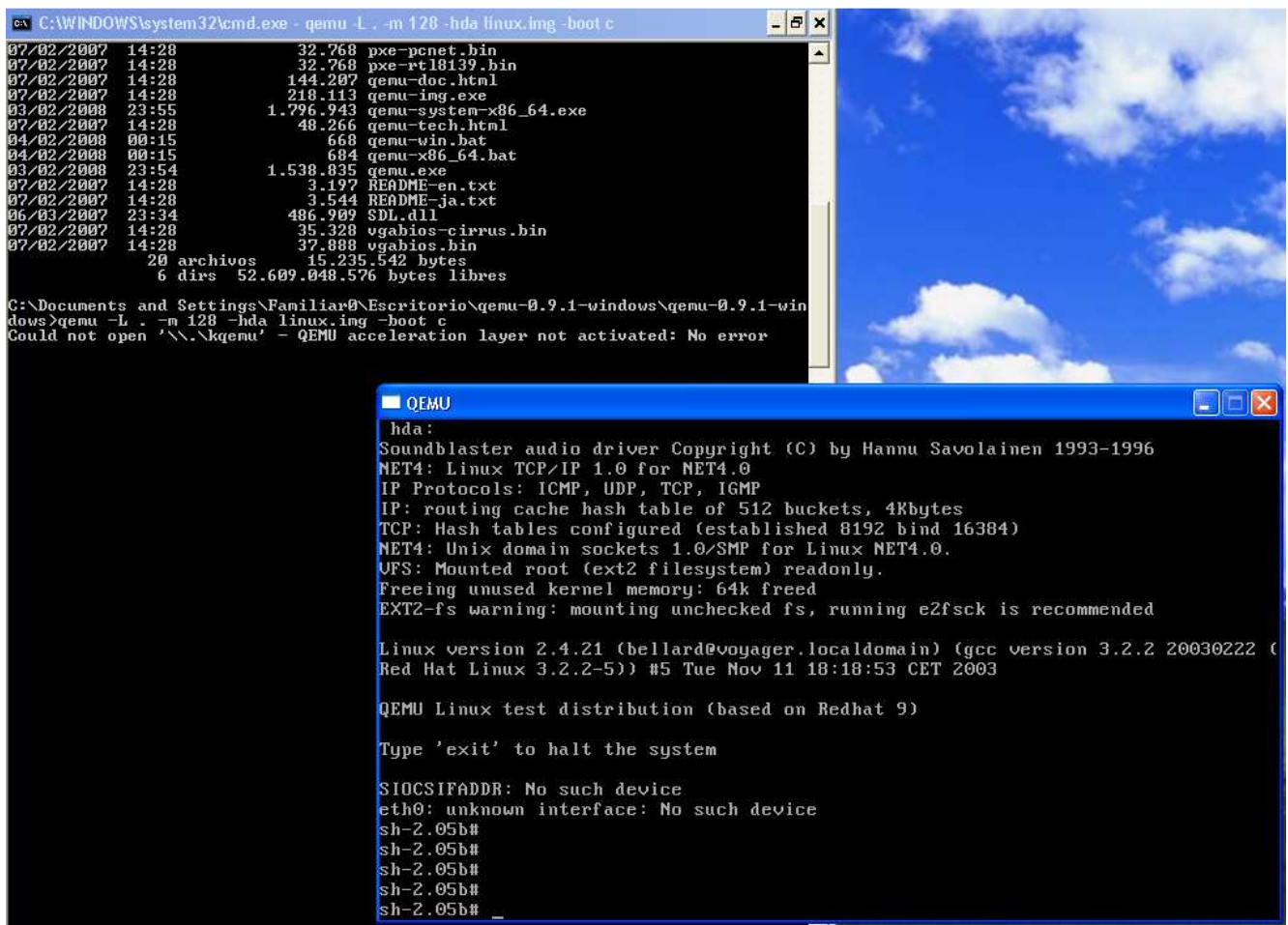
```
“qemu-img convert imagenAntigua.vmdk imagenNueva.bin”
```

También es posible realizar el proceso contrario invirtiendo el orden. Aunque si lo que queremos es crear una MV para VMWare deberemos generar también el archivo vmx con alguna de las herramientas que podemos encontrar en la red. [22]

Nota: si la imagen creada con VMWare está almacenada en varios archivos no será posible convertirla con qemu-img.

- 3- Lo siguiente que debemos hacer, tras descargar e instalar Qemu en Windows, es ejecutar la imagen. Para ello introducimos, en la línea de comandos, la siguiente orden:

`"qemu.exe -m RAMDisponible -hda NombreDelArchivo -boot -c"`



```
C:\WINDOWS\system32\cmd.exe - qemu -L . -m 128 -hda linux.img -boot c
07/02/2007 14:28 32.768 pxe-pcnet.bin
07/02/2007 14:28 32.768 pxe-rt18139.bin
07/02/2007 14:28 144.207 qemu-doc.html
07/02/2007 14:28 218.113 qemu-img.exe
03/02/2008 23:55 1.796.943 qemu-system-x86_64.exe
07/02/2007 14:28 48.266 qemu-tech.html
04/02/2008 00:15 668 qemu-win.bat
04/02/2008 00:15 684 qemu-x86_64.bat
03/02/2008 23:54 1.538.835 qemu.exe
07/02/2007 14:28 3.197 README-en.txt
07/02/2007 14:28 3.544 README-ja.txt
06/03/2007 23:34 486.909 SDL.dll
07/02/2007 14:28 35.328 vgabios-cirrus.bin
07/02/2007 14:28 37.888 vgabios.bin
20 archivos 15.235.542 bytes
6 dirs 52.609.048.576 bytes libres

C:\Documents and Settings\Familiar0\Escritorio\qemu-0.9.1-windows\qemu-0.9.1-win
dows>qemu -L . -m 128 -hda linux.img -boot c
Could not open '\\.\kqemu' - QEMU acceleration layer not activated: No error

QEMU
hda:
Soundblaster audio driver Copyright (C) by Hannu Savolainen 1993-1996
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 8192 bind 16384)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 64k freed
EXT2-fs warning: mounting unchecked fs, running e2fsck is recommended

Linux version 2.4.21 (bellard@voyager.localdomain) (gcc version 3.2.2 20030222 (
Red Hat Linux 3.2.2-5)) #5 Tue Nov 11 18:18:53 CET 2003

QEMU Linux test distribution (based on Redhat 9)

Type 'exit' to halt the system

SIOCSIFADDR: No such device
eth0: unknown interface: No such device
sh-2.05b#
sh-2.05b#
sh-2.05b#
sh-2.05b#
sh-2.05b#
```

Nota: también es posible iniciar MVs que ejecuten modo gráfico.

4.10 Restauración rápida del sistema

Para reducir el tiempo necesario en estabilizar todo el sistema cuando uno de los nodos cae o hemos de añadir uno nuevo, hemos creados dos scripts, uno para el nodo principal y otro para cualquiera de los secundarios, capaces de reinstalar la red así como formar de nuevo las relaciones entre los nodos dentro del clúster.

Para el nodo principal

Las tareas de recuperación completa del nodo principal en caso de avería insalvable o pérdida total del sistema son las siguientes:

- **Reinstalar Proxmox:** La reinstalación de Proxmox consistirá en una instalación nueva del mismo. Si seguimos los pasos explicados en el punto 4.3 conseguiremos eliminar, además, de

forma automática la instalación anterior debido a que ProxMox formatea siempre la partición antes de instalarse en ella.

- **Restaurar la red:** Lo primero que hay que hacer es configurar la red. Para ello hemos guardado el archivo “/etc/network/interfaces” en un sistema de almacenamiento seguro a modo de copia de seguridad. Así pues, si ejecutamos el primero de los scripts, de nombre “NodoPrincipal”, conseguiremos reemplazar el archivo nuevo por el que contiene la configuración de red correcta y acto seguido reiniciar las interfaces de red con la orden:

```
/etc/init.d/networking restart
```

- **Eliminar configuraciones previas:** Para que los nodos vuelvan a verse entre ellos y establezcan las relaciones pertinentes es necesario eliminar el archivo donde se guardan las configuraciones anteriores (en caso de no haber tenido que reinstalar el sistema completamente) llamado “/etc/pve/cluster.cfg” así como las posibles claves que tengamos guardadas de los otros nodos y que puede ser que no coincidan. Para ello borramos el contenido del fichero “/root/.ssh/known_hosts”. Esto también lo hará de forma automática nuestro script.
- **Creando de nuevo el clúster:** Para esta tarea basta con que el script ejecute finalmente la orden que crea un clúster nuevo estableciendo este nodo como master del mismo:

```
pveca -c
```

Terminados los pasos anteriores es imprescindible ejecutar el segundo script, llamado *NodoGeneral*, en cada uno de los nodos secundarios. Para enviar este archivo a cada uno de ellos utilizaremos la orden “scp” o “secure copy” que nos proporciona SSH. Este programa lo utilizaremos para ejecutar el fichero con la orden

```
sh NodoGeneral
```

Para el nodo general o secundario

Para los nodos secundarios el procedimiento a realizar es el mismo exceptuando la necesidad de configurar la red ya que, al instalar ProxMox ya configuramos la IP, máscara y servidores DNS que éste tendrá y el resto de piezas de la configuración de red se instalan de forma automática. Para finalizar sustituiremos la orden de creación del clúster por la que permite añadir un nuevo nodo:

```
pveca -a -h 10.0.0.1
```

Nota: El contenido de los scripts se encuentra en forma de anexo al final de este documento.

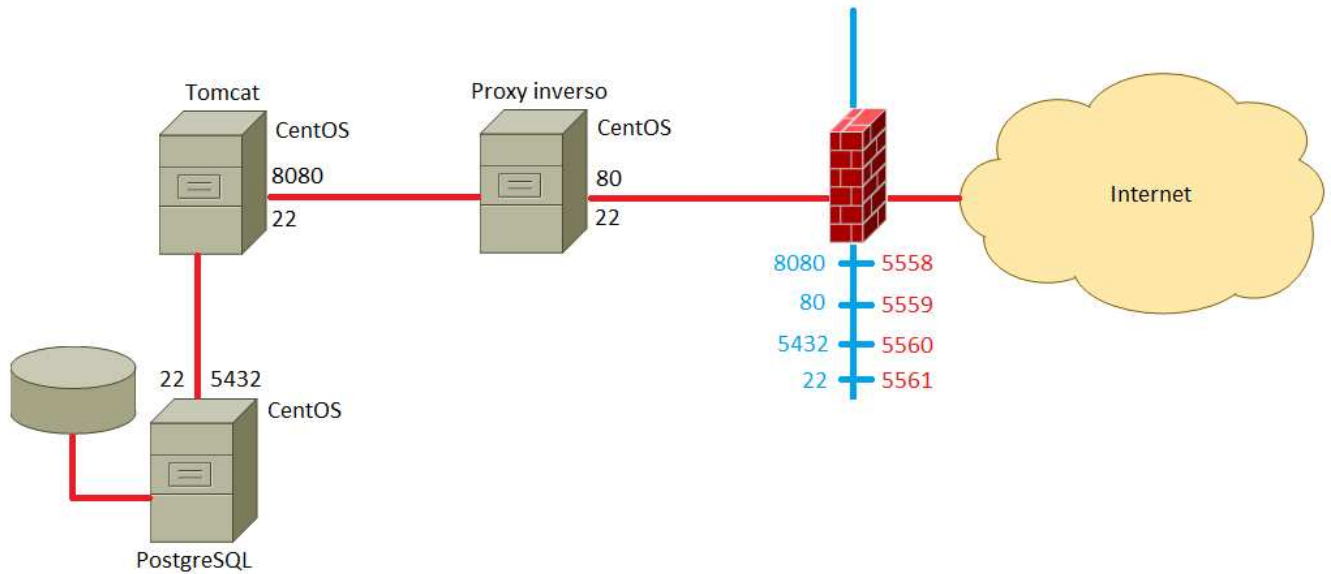
Capítulo 5 RESULTADOS Y CONCLUSIONES

5.1 Simulación de uso

La pregunta que debemos hacernos llegados a este punto es: ¿Cómo puede una asignatura sacar partido de esta tecnología? Para responderla vamos a proponer un escenario, bastante completo, que abarque muchas de las posibilidades que tiene el proyecto además de ilustrar las secuencias necesarias para realizar las operaciones típicas generales.

El escenario de ejemplo que se le va a plantear al alumno para que resuelva una práctica es el siguiente:

Se le proporcionan a cada estudiante tres máquinas virtuales: una con una instalación de Tomcat; servidor que escuchará peticiones en el puerto 8080, una segunda con PostgreSQL preinstalado; este servidor de bases de datos utilizará el puerto 5432 y por último, una tercera máquina virtual con la instalación de un proxy inverso capaz de aceptar peticiones en el puerto 80. También vamos a permitir una conexión al puerto 22 (ssh) de la máquina con Tomcat para que el alumno pueda gestionarla y acceder a las otras MV, que le correspondan, dentro de la red privada mediante conexiones, también, ssh



Creando una plantilla

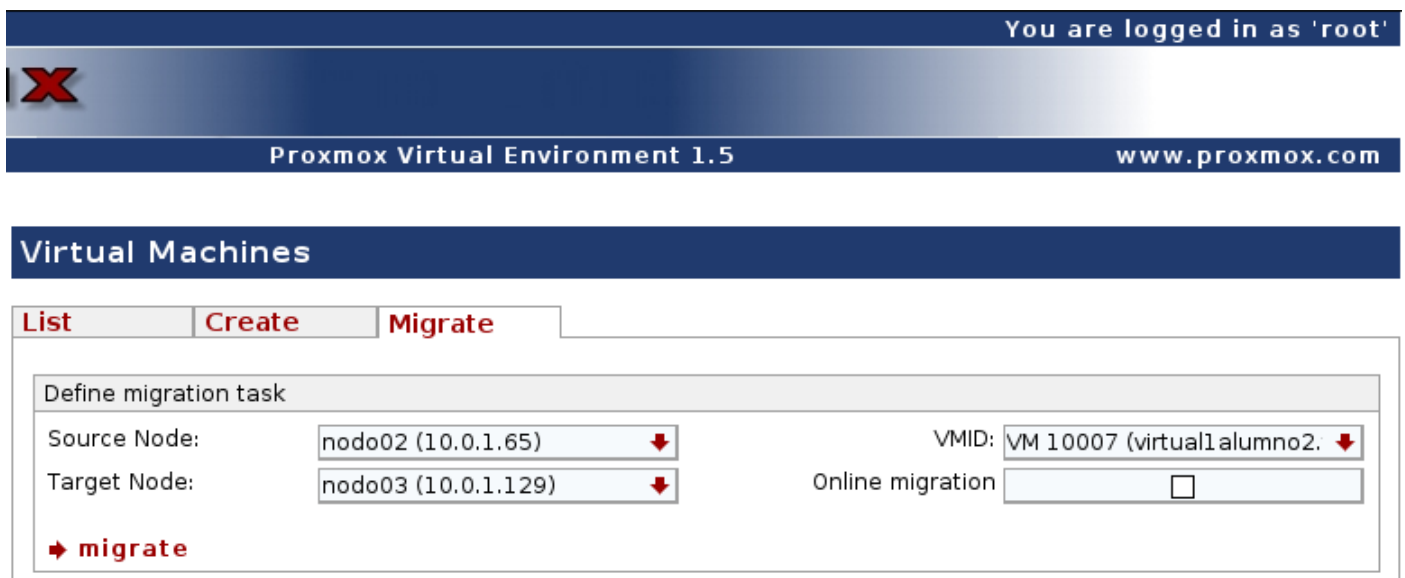
Todo empieza por la creación de una plantilla por parte del profesor. La forma más rápida de crearla es basándonos en una de las plantillas que ofrece ProxMox o que se pueden encontrar en la web:

- 1- Empezamos creando una máquina virtual con la plantilla base elegida tal y como se explicó en el cuarto capítulo. En nuestro caso elegiremos una de las que vienen con ProxMox con el sistema operativo CentOS 5 instalado.
- 2- Entramos en la máquina virtual por VNC desde la interfaz de ProxMox o mediante una conexión ssh e instalamos o eliminamos los programas que queramos. Por ejemplo podríamos instalar Tomcat.
- 3- Salimos de la MV sin detenerla, borramos su dirección IP desde la pestaña de propiedades de la MV que nos proporciona ProxMox y le damos al botón de guardado "Save".
- 4- Ahora sí, detenemos la máquina haciendo click en el botón "Stop".
- 5- Comprimimos los archivos pertenecientes a la MV y que se encuentran en el directorio `/var/lib/vz/private/Identificador de la MV`
- 6- El archivo resultante lo renombramos según el formato para las templates que admite ProxMox `(nombreDelSistemaOperativo-edición-nombreDeLaMV_Versión_JuegoDeInstrucciones.tar.gz)` y lo movemos finalmente al directorio donde se almacenan las templates `/var/lib/vz/template/cache` [46][47]

Repetiremos el proceso para las otras dos plantillas que vamos a utilizar en este ejemplo.

Preparando el sistema

El alumno envía un e-mail al profesor informándole de que desea realizar la práctica. El profesor elige cuatro puertos del rango 50.000 al 65.000 para asignárselos al alumno. Estos se corresponderán con los puertos 80 del proxy inverso, 8080 de Tomcat, 5432 de PostgreSQL y 22 del ssh de la máquina con Tomcat. También calculamos las IPs que puede utilizar el alumno basándonos en el rango que reservamos en el punto 4.1. Después estos datos se almacenan en un fichero, se dan de alta en el firewall de puerta para permitir el acceso de ese alumno a esos puertos (mediante NAT cuando conecten contra la IP de puerta) y también se envían al alumno para que pueda usarlos. Sólo falta crear las máquinas virtuales pertenecientes al alumno con las IPs elegidas. Al no tratarse de un sistema automático y de no tener, de momento, una interfaz dónde el usuario pueda crear y administrar sus MVs, será el profesor quien las cree basándose en las plantillas realizadas con anterioridad y balanceando la carga entre cada uno de los nodos secundarios. Esta forma de plantear el sistema nos permite poder migrar las máquinas, como ya explicamos en el punto 4.5, de forma completamente transparente al alumno.



The screenshot shows the Proxmox Virtual Environment 1.5 web interface. At the top, it indicates the user is logged in as 'root'. The main navigation bar includes 'List', 'Create', and 'Migrate' tabs, with 'Migrate' being the active tab. Below the navigation, there is a 'Define migration task' form. The form contains the following fields:

- Source Node: nodo02 (10.0.1.65)
- Target Node: nodo03 (10.0.1.129)
- VMID: VM 10007 (virtual1alumno2)
- Online migration:

A red arrow icon followed by the text 'migrate' is visible at the bottom left of the form area.



La parte del alumno

- 1- "Alumno1" realiza una conexión VPN a la UPV.[25]



- 2- Una vez conectado y autenticado en la intranet se le permite el acceso al servidor "puerta.dsic.upv.es" mediante un navegador web o una conexión ssh. En función del puerto, de los que se le han asignado, al que conecte podrá ver su servidor web o acceder a una máquina para administrarla, y de ahí también a otras. Ahora ya solo tiene que resolver el boletín de la práctica correspondiente.

```

root@virtuallalumno1:/
[root@virtuallalumno1 ~]# cd ..
[root@virtuallalumno1 /]# ls -l
total 60
lrwxrwxrwx 1 root root 39 Aug 11 20:50 aquota.group -> /proc/vz/vzaquota/00000014/aquota.group
lrwxrwxrwx 1 root root 38 Aug 11 20:50 aquota.user -> /proc/vz/vzaquota/00000014/aquota.user
drwxr-xr-x 2 root root 4096 Jan 17 2009 bin
drwxr-xr-x 2 root root 4096 Mar 29 2007 boot
drwxr-xr-x 7 root root 1860 Aug 11 20:50 dev
drwxr-xr-x 46 root root 4096 Aug 11 20:50 etc
drwxr-xr-x 2 root root 4096 Mar 29 2007 home
drwxr-xr-x 11 root root 4096 Jan 17 2009 lib
drwxr-xr-x 2 root root 4096 Mar 29 2007 media
drwxr-xr-x 2 root root 4096 Mar 29 2007 mnt
drwxr-xr-x 2 root root 4096 Mar 29 2007 opt
dr-xr-xr-x 31 root root 0 Aug 11 20:50 proc
drwxr-x--- 3 root root 4096 Jul 22 19:08 root
drwxr-xr-x 2 root root 4096 Jun 21 19:18 sbin
drwxr-xr-x 2 root root 4096 Mar 29 2007 selinux
drwxr-xr-x 2 root root 4096 Mar 29 2007 srv
drwxr-xr-x 3 root root 0 Aug 11 20:50 sys
drwxrwxrwt 3 root root 4096 Aug 11 20:50 tmp
drwxr-xr-x 13 root root 4096 Jan 8 2009 usr
drwxr-xr-x 19 root root 4096 Jan 8 2009 var
[root@virtuallalumno1 /]#
    
```

Monitorización y control

Por otro lado, el profesor puede conectarse a los programas de monitorización de la red mediante conexión SSH al nodo principal (“puerta”) y ver en tiempo real que está ocurriendo en la red interna. Para ello podremos usar programas tales como Zenoss, etherape o wireshark, antes descritos, que se ejecutarán en el nodo principal y que pueden configurarse, como vimos en el capítulo 2, para monitorizar las interfaces virtuales por las que todo el tráfico de las MVs. Así podremos darnos cuenta de cualquier conexión no permitida que haya conseguido saltarse las reglas de IPTables configuradas en cada nodo para evitar que los alumnos puedan acceder a máquinas virtuales que no les pertenecen.

5.2 Valoración de resultados

Vemos claramente que la solución realizada resuelve el problema planteado. Se ha conseguido crear un sistema el cual va a permitir a los alumnos realizar prácticas con un dominio total sobre el sistema operativo y además con una disponibilidad de los recursos de 24 horas al día durante todo el curso. Por otra parte el profesor controla todos los elementos del mismo y es capaz de observar situaciones extrañas que puedan ocurrir además de tener una visión global del estado de todas las MVs creadas así como del uso de los recursos físicos: uso del procesador, memoria RAM, espacio en disco duro, saturación de la red... Este control le permite, además, migrar y distribuir las MVs entre los diferentes nodos en caso de que se aprecie un desequilibrio en la utilización de recursos de cada servidor.

La seguridad es otro tema que hemos conseguido completar. Por un lado tenemos aislamiento entre el trabajo de los usuarios gracias a los firewalls introducidos en cada nodo mediante IPTables y gracias a la supervisión de la red con el software de monitoreo. Además, el hecho de que nadie pueda entrar a la red interna de la UPV sin haberse autenticado previamente nos asegura que nadie externo a la UPV podrá utilizar la aplicación. También es cierto que sólo los alumnos que estén presentes en la lista de alumnos autorizados, es decir, los alumnos matriculados en la asignatura y dados de alta por el profesor, podrán acceder al sistema. Por otro lado, los estudiantes sólo podrán realizar en sus máquinas virtuales lo que el profesor haya permitido en la plantilla que el alumno está obligado a utilizar tanto si el profesor crea la MV tanto como si se selecciona en la interfaz de usuario a la hora de crear una MV. Esto quiere decir, por ejemplo, que si la plantilla no contiene el comando apt-get, el usuario no podrá descargar e instalar programas de Internet.

Podemos considerar que el sistema es rápido, si bien es cierto que no ha sido probado en un entorno real y es obvio que si se agotasen los recursos por un uso masivo por parte de los usuarios en períodos clave, como una semana antes de entregar los trabajos, el sistema se volvería lento e incluso podría dejar de funcionar. Por este motivo hemos presentado también la opción de la descarga de las plantillas con el fin de poder trabajar en el ordenador local de cada usuario aliviando así la carga de nuestros servidores. Por otro lado intentaremos trabajar sobre las MV desde un terminal de comandos y no en modo gráfico, ya que éste exige mayor capacidad de procesamiento y memoria y genera mucho más tráfico de red, la cual es el cuello de botella del proyecto.

Otra parte a la que se le ha dado importancia es la necesidad de que fuese fácil de manejar para que el usuario final pudiese concentrarse en lo que realmente importa, que es la resolución de una práctica. Pensamos que el alumno no va a dedicar mucho tiempo a la comprensión de la solución puesto que hasta la interfaz del usuario es muy intuitiva. Quizá pueda sorprender algo más la forma de realizar la conexión SSH a las MVs creadas así como a cada uno de los puertos, aunque esto vendrá bien explicado en el boletín de la práctica.

Creemos que la solución planteada podría ser aplicable a un entorno de trabajo real, es decir, es factible que los alumnos utilicen nuestro proyecto, con todas las funciones terminadas, en un futuro, para realizar el tipo de prácticas que se pretende en la Universidad Politécnica de Valencia.

5.3 Ampliaciones futuras

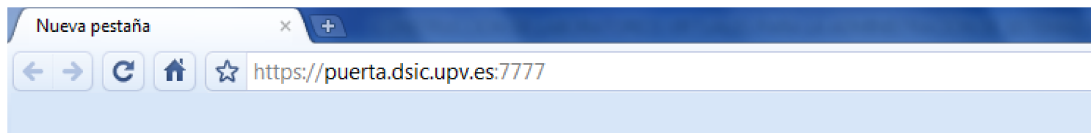
Futuros proyectos

Se plantean tres proyectos futuros de utilidad para esta solución:

- 1- Automigración de máquinas virtuales: Podemos crear un no muy complejo programa en, por ejemplo, Perl que cuando detecte un desequilibrio en el nivel de trabajo de los nodos, sea capaz de migrar automáticamente una máquina virtual de un servidor a otro haciendo uso de la subrutina de migración de ProxMox. Esto haría de nuestro proyecto un sistema más eficiente y más rápido.
- 2- Chat: Si queremos proponer prácticas que puedan ser resueltas en grupo y teniendo en cuenta que cada alumno puede estar trabajando en su respectiva casa, es necesario una comunicación entre los miembros del equipo para que no se den situaciones del tipo: un alumno restaura una copia de seguridad mientras otro miembro está realizando una prueba. Este chat puede ser una aplicación simple que esté integrada dentro de las plantillas y que con una instrucción “*chat mensaje*” se muestre por pantalla en los terminales de los usuarios conectados a la MV el texto que se pretendía enviar, así podríamos avisar al resto del grupo de nuestras intenciones. Hay algunas soluciones ya creadas para Unix con las que diversas conexiones ssh a un mismo servidor pueden comunicarse entre sí por mensajes de texto como podría ser ytalk y talkd.[26]
- 3- Interfaz web: La interfaz web es un elemento necesario en el sistema para permitir que los alumnos interactúen con el mismo de forma fácil y segura. Esta deberá ser programada e instalada en el servidor principal en proyectos futuros.

Ejemplo de uso con interfaz web

- 1- El alumno “Alumno1” realiza una conexión VPN a la UPV.[25]
- 2- Una vez conectado y autenticado en la intranet se le permite el acceso al puerto 7777 del servidor “puerta.dsic.upv.es” mediante un navegador web accediendo automáticamente a una interfaz personalizada para cada alumno con sus máquinas virtuales.
- 3- La aplicación en “puerta” carga los datos pertenecientes al usuario y rellena la interfaz.



Alumno Alumno1

MV	IP	Puerto	Estado					
CentosApache1	10.0.0.2	65534	iniciada	Restart	Shutdown	Stop	Remove	BackUp
CentosApache2	10.0.0.3	65537	detenida	Start	Shutdown	Stop	Remove	BackUp
Prueba2	10.0.0.5	65543	iniciada	Restart	Shutdown	Stop	Remove	BackUp

Crear MV nueva

Template: Centos5ConApache | Nombre: |

Centos5ConApache
 Centos5ConPlone
 Centos5ConZope
 WindowsXP

Prueba2 ha sido creada correctamente. Identificador 10005.

Copias de seguridad

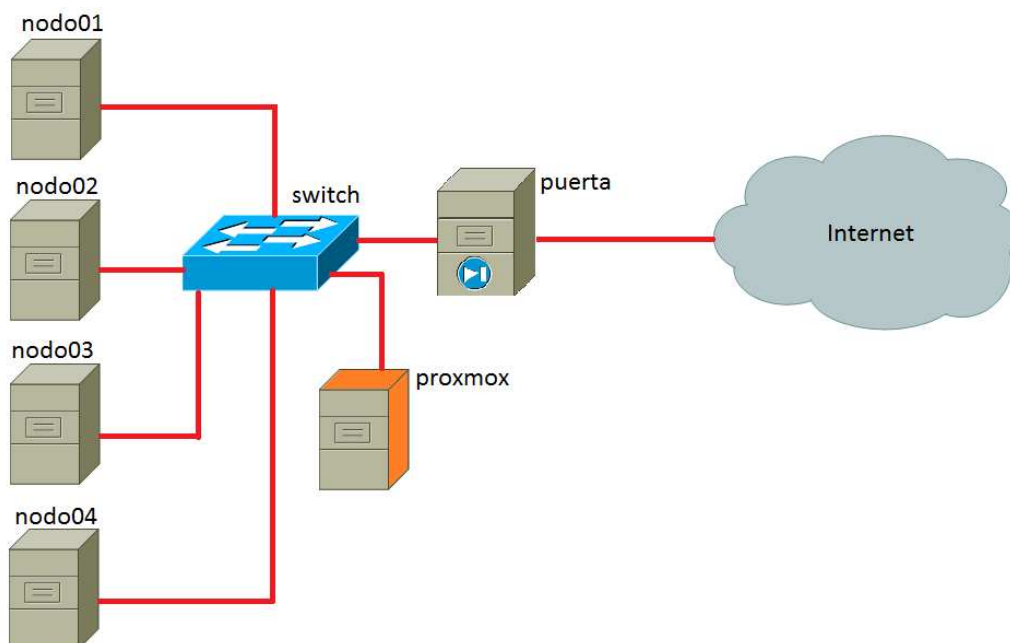
Apache1_15	Restore	Download
Apache1_16	Restore	Download
Apache2_2	Restore	Download
Apache2_3	Restore	Download

- En esta interfaz el alumno consulta sus máquinas virtuales, crea nuevas (a partir de una de las plantillas que ha preparado previamente el profesor con el material necesario para la asignatura) y crea o restaura copias de seguridad.
- Esta MV que el alumno quiere crear se llevará a cabo mediante una orden a OpenVz como vimos en el apartado de diseño, además cada nueva máquina virtual se crea en el nodo siguiente al que se creó la última obteniendo así balanceo de carga entre los servidores del Clúster ProxMox.
- El alumno accede a la MV mediante una conexión SSH a “puerta” con el puerto que la aplicación le otorga automáticamente a la misma o uno de los que el profesor le ha asignado al alumno. Los dos números de puerto que van inmediatamente a continuación del otorgado serán para conexiones web, es decir, cumplirán la función del puerto bien conocido 80 y el 8080.
- “Puerta”, después de comprobar (en un fichero donde cada alumno tiene asignados los puertos) que, efectivamente, el puerto le corresponde al alumno que intenta conectar a el mismo y mediante IPTables, redirecciona la conexión a la máquina virtual correcta.

- 8- El alumno puede ahora trabajar sobre la MV.
- 9- Si se desea restaurar una copia de seguridad, previamente creada con el botón “BackUp” de la interfaz web en “puerta”, el alumno sólo tiene que volver a la interfaz web y hacer click sobre el botón “restore” del apartado copias de seguridad. Esto producirá la eliminación de la MV a la que corresponde la copia y la creación de una nueva a partir de la copia volviendo así a un estado anterior de la MV.
- 10- Mediante el botón “Download” el alumno puede descargar mediante FTP cualquier imagen guardada para poder trabajar en modo local desde casa. Si el sistema operativo que usa es Windows, podrá transformar la imagen a un formato compatible con Qemu Windows de la forma que explicamos en el capítulo 4. No permitiremos subir imágenes al sistema ya que esto lo consideramos como un fallo de seguridad. Así pues una vez realizadas las pruebas en local, el trabajo final deberá hacerse sobre nuestro sistema.
- 11- Por otro lado, el profesor puede conectarse a los programas de monitorización de la red mediante conexión SSH al nodo principal (“puerta”) y ver en tiempo real que está ocurriendo en la red interna.

Correcciones

Al descubrir que la única forma posible de poder crear máquinas virtuales KVM, por si nos hiciesen falta en un futuro, es teniendo soporte para virtualización por hardware en los procesadores de todos los nodos que forman el sistema, especialmente el principal (aunque no vaya a virtualizar máquinas KVM en el mismo), hemos decidido añadir un nuevo nodo al sistema, llamado “proxmox” el cual será ahora el nuevo máster del clúster y liberando “puerta” de este trabajo. Ya que puerta no tiene esta característica en su procesador, se limitará a traducir los puertos para permitir las conexiones a cada una de las MVs de los alumnos, así como hacer la función de firewall y ser la única puerta de entrada desde el exterior hacia nuestra red privada.



Capítulo 6 BIBLIOGRAFÍA

[1] <http://ovirt.org>

Página oficial de Ovirt. (12-09-2010)

[2] www.openqrm.com

Página oficial de OpenQRM. (12-09-2010)

[3] www.tranquilidadtecnológica.com

La página ya no se encuentra disponible. (12-09-2010)

[4] <http://sourceforge.net/projects/openqrm>

Descarga gratuita de OpenQRM basado en Ubuntu 8. (12-09-2010)

[5] <http://www.openqrm-ng.net/downloads/base/openqrm-4.4-initrd-template-selection.tgz>

Enlace directo a la descarga de un paquete con varias plantillas diferentes. (12-09-2010)

[6] http://pve.proxmox.com/wiki/Main_Page

Página oficial de ProxMox Virtual Environment. (12-09-2010)

[7] www.serverchief.com/content/how-to-install-openqrm

Tutorial para instalar OpenQRM. (12-09-2010)

[8] <http://www.zenoss.com/>

Web oficial del programa Zenoss. (12-09-2010)

[9] <http://www.scribd.com/doc/8410056/Configuracion-Del-Agente-Snmp-en-DebianWindows-XpWindows-Vista-Windows-Server-2003>

Tutorial de configuración de un agente SNMP. (12-09-2010)

[10] <http://www.slideshare.net/ces1227/zenoss-manual-presentation>

Manual del programa Zenoss. (12-09-2010)

[11] <http://etherape.sourceforge.net/>

Información y descarga del programa de monitorización Etherape. (12-09-2010)

[12] www.wireshark.org

Página oficial del programa Wireshark. (12-09-2010)

[13] <http://www.wireshark.org/docs/man-pages/tshark.html>

Manual del programa Tshark. (12-09-2010)

[14] <http://www.liberouter.org/nific/usecases/rpcap/rpcap.php>

Información y descarga de rpcap. (12-09-2010)

[15] <http://blog.loftninjas.org/2008/08/>

Explicación para capturar remotamente paquetes con Wireshark y Tshark. (12-09-2010)

[16] <http://blog.ownhost.net/page/2/>

Tutorial de cómo instalas ProxMox desde el que se han extraído fotografías. (12-09-2010)

[17] <http://c-nergy.be/blog/?p=356>

Tutorial de cómo instalar ProxMox. (12-09-2010)

[18] <http://www.howtoforge.com/kvm-and-openssh-virtualization-and-cloud-computing-with-proxmox-ve-p2>

Tutorial sobre cómo crear un clúster y añadir más nodos. (12-09-2010)

[19] http://wiki.openssh.org/Differences_between_venet_and_veth

Explicación de las diferencias entre las opciones venet y veth. (12-09-2010)

[20] <http://ytuquelees.net/convertir-una-imagen-kvmqemu-a-vmware/>

Tutorial sobre cómo convertir imágenes KVM a VMWare. (12-09-2010)

[21] <http://www.h7.dion.ne.jp/~qemu-win/>

Información y descarga de Qemu para Windows. (12-09-2010)

[22] <http://easyvmx.com/>

Descarga del programa generador de archivos vmx easyvmx. (12-09-2010)

[23] <http://sourceforge.net/projects/vcxsrv/>

Descarga gratuita de VcXsrv Windows X Server. (12-09-2010)

[24] <http://www.java.com/es/download/>

Descarga gratuita de JAVA. (12-09-2010)

[25] <http://www.upv.es/contenidos/miw/infoweb/infoacceso/dat/732845normalc.html>

Explicación de la página web de la UPV dedicada a configurar la conexión VPN necesaria para entrar en la red privada de la universidad. (12-09-2010)

[26] <http://forums.whirlpool.net.au/forum-replies-archive.cfm/987115.html>

Tema del foro de Whirlpool dedicado a proponer aplicaciones de chat sobre ssh. (12-09-2010)

[27] <http://forum.proxmox.com/threads/2823-What-is-IODelay>

Tema del foro de ProxMox donde se explica el campo IODelay. (12-09-2010)

[28] www.arkoon.net/-Glosario-.html

Definición del término Appliance o Plantilla en el ámbito de la virtualización. (12-09-2010)

[29] es.wikipedia.org/wiki/Bit

Definición del término bit en el ámbito informático.(12-09-2010)

[30] es.security.ngoinabox.org/glossary

Definición del término certificado de seguridad en el ámbito informático.(12-09-2010)

[31] [es.wikipedia.org/wiki/Cluster_\(informática\)](http://es.wikipedia.org/wiki/Cluster_(informática))

Definición del término clúster en el ámbito informático.(12-09-2010)

[32] [es.wikipedia.org/wiki/Dominio_\(redes_informáticas\)](http://es.wikipedia.org/wiki/Dominio_(redes_informáticas))

Definición del término dominio en el ámbito informático.(12-09-2010)

[33] es.wikipedia.org/wiki/Escalabilidad

Definición del término escalabilidad. (12-09-2010)

[34] http://es.wikipedia.org/wiki/Gigabit_Ethernet

Definición de los tipos de tecnología Ethernet. (12-09-2010)

[35] http://es.wikipedia.org/wiki/Imagen_ISO

Definición del término Imagen ISO. (12-09-2010)

[36] es.wikipedia.org/wiki/Máquina_virtual

Definición del término Máquina Virtual. (12-09-2010)

[37] wwwdi.ujaen.es/~lina/TemasSO/glosario/GLOSARIO.htm

Definición del término Sistema distribuido. (12-09-2010)

[38] es.wikipedia.org/wiki/Software

Descripción del término software.(12-09-2010)

[39] www.faq-mac.com/noticias/31544/diccionario-basico-tecnologia

Descripción de las siglas VNC. (12-09-2010)

[40] www.hard-h2o.com/diccionario-informatico_s-z.html

Descripción del término virtualización.(12-09-2010)

[41] <http://es.wikipedia.org/wiki/Virtualizaci%C3%B3n>

Qué es la virtualización. (11-09-2010)

[42] <http://es.wikipedia.org/wiki/Paravirtualizaci%C3%B3n>

Qué es la paravirtualización. (12-09-2010)

[43] <http://wiki.openvz.org/Venet>

Explicación sobre la interfaz de red virtual venet. (13-09-2010)

[44] http://wiki.openvz.org/Traffic_shaping_with_tc

Breve explicación sobre la comunicación de la máquina virtual con un servidor remoto. (13-09-2010)

[45] http://pve.proxmox.com/wiki/Network_Model

Explicación sobre la interfaz virtual vmbr y como configurarlo dependiendo de la utilidad que se le vaya a dar. (13-09-2010)

[46] http://wiki.openvz.org/Centos_template

Información sobre cómo crear plantillas OpenVZ. (14-09-2010)

[47] <http://forum.proxmox.com/threads/452-Creating-custom-templates?highlight=create+template>

Cómo crear plantillas para ProxMox y OpenVZ. (14-09-2010)

ANEXOS

A.1 Scripts

A.1.1 NodoPrincipal

```
#!/bin/sh
echo ""
echo "Restaurando nodo principal"
echo ""
echo "Configurando la red"
echo ""
cp -v interfacesNodoP /etc/network/interfaces
/etc/init.d/networking restart
echo ""
echo "Eliminando posibles claves antiguas"
echo ""
rm -v /etc/pve/cluster.cfg
> /etc/pve/cluster.cfg
rm -v /root/.ssh/known_hosts
> /root/.ssh/known_hosts
echo ""
echo "Creando master del clúster"
echo ""
pveca -c
pveca -l
echo ""
echo "Terminado. No olvide ejecutar la restauración en los nodos generales"
```

A.1.2 NodoGeneral

```
#!/bin/sh
echo ""
echo "Restaurando nodo general"
echo ""
echo "Eliminando posibles claves antiguas"
echo ""
rm -v /etc/pve/cluster.cfg
> /etc/pve/cluster.cfg
rm -v /root/.ssh/known_hosts
> /root/.ssh/known_hosts
echo ""
echo "Añadiendo nodo al clúster"
echo ""
pveca -a -h 10.0.0.1
echo ""
echo "Terminado"
echo ""
```