

Proyecto de Virtualización de Aulas informáticas mediante el uso de Software Libre.

Aplicación práctica en el Dpto. de Organización de Empresas de la Universitat Politècnica de València.

Francisco J. Villar, Raúl Blasco Herrera, Fernando Ferrer

Valencia Noviembre de 2015

Resumen

Este documento recoge el desarrollo del proyecto “Virtualización de aulas informáticas mediante el uso de Software Libre” implementado en el Departamento de Organización de Empresas (DOE) de la Universitat Politècnica de València, UPV. Se ha recogido y ampliado la experiencia previa realizada en el Departamento de Sistemas de Información y Computación (DSIC), UPV. Dado que todo el sistema se ha basado en la utilización de herramientas de software de libre distribución, su implementación requiere de un alto coste en horas de personal técnico, este documento ha sido creado con la intención de ayudar a reducir el número de horas necesario para su puesta en funcionamiento en escenarios similares.

Abstract

This document describes the development of the project: "Computer Labs virtualization using Free Software" implemented in the Department of Management (DOE) of the Polytechnic University of Valencia, UPV. It has collected and expanded prior experience and knowledge obtained at the Information Systems and Computing Department (DSIC), UPV. Since the entire system is based on the usage free software tools, its implementation has required a high degree of technical personnel involvement. This document has been created to help reducing the number of hours needed to its implementation in similar scenarios.

Mapa del documento

El documento comienza con una sección dedicada a presentar los objetivos, el escenario y que servicio se pretende ofrecer, así como las fases seguidas para conseguirlo. A partir de ahí encontraremos dos capítulos en los que desarrollaremos los dos modos de abordar nuestra problemática en formato de “guía”. Para finalizar se incluyen anexos con el objeto de reforzar y complementar los aspectos de la instalación que así lo han requerido.

Contenido

1	Capítulo 1: IMPLEMENTACIÓN DE LA FASE 1.....	9
1.1	Antecedentes.....	9
1.2	Instalación.....	11
1.2.1	Consideraciones y requisitos previos	11
1.2.2	Preparación de los sistemas con VT-x	12
1.2.3	Preparación de los sistemas sin VT-x.....	14
1.3	Gestión y Actualización de las máquinas virtuales locales.....	15
1.3.1	Preparación de la nueva máquina virtual.....	15
1.3.2	Servidor de actualización de máquinas virtuales locales: "actualizador".	17
1.3.3	Despliegue de una nueva máquina virtual en el laboratorio.....	20
2	Capítulo 2: IMPLEMENTACIÓN DE LA FASE 2.....	22
2.1	Descripción de la arquitectura elegida.....	22
2.1.1	Estructura lógica.....	23
2.1.2	Infraestructura hardware.....	25
2.2	Instalación de la Cabina de Almacenamiento.....	27
2.2.1	Configuración almacenamiento en Asus RS12, caso particular DOE.....	28
2.2.2	Instalación de Centos 7 server "minimal" en la cabina de almacenamiento.....	28
2.2.3	Configurar recursos NFS mediante LVM.....	29
2.2.4	Supervisión y control de los Discos del Raid	32
2.3	Instalación de los servidores con XenServer	33
2.3.1	Instalación de un servidor con XenServer 6.5:.....	33
2.3.2	Instalación y configuración de XenCenter.....	34
2.4	Creación y migración de máquinas virtuales.....	39
2.4.1	Creación de una máquina virtual desde XenCenter.....	39
2.4.2	Generación de máquinas virtuales y "clones" para aulas.....	40
2.4.3	Modificar una máquina virtual para "congelar" su disco duro.....	43
2.4.4	Creación de máquinas virtuales y "clones" para otros propósitos.....	47
2.4.5	Migración desde VirtualBox con XP.....	48
2.5	Bróker estático: V1.0.....	50
2.5.1	Vinculación cliente físico - máquina virtual.....	50

2.5.2	Funciones de servidor DHCP y DNS.....	52
2.5.3	Funciones de enrutamiento.....	56
2.6	Bróker v2.0: Versión Dinámica.....	58
2.6.1	Ajustes necesarios en el Servidor "broker.dominio.compañia.es".....	58
2.6.2	Ajustes necesarios en el cliente físico.....	61
2.7	Alta Disponibilidad (HA).....	62
2.7.1	Creación de un nuevo recurso de almacenamiento (SR) para HA.....	63
2.7.2	Montar el nuevo SR en XenCenter.....	64
2.7.3	Habilitar HA.....	65
3	ANEXOS.....	68
3.1	ANEXO I: "Fichero script "Isi"".....	68
3.2	ANEXO II: Manejo de unidades USB. Auto-montaje USB:.....	77
3.3	ANEXO III: Desarrollo de una Interfaz gráfica "Web-app" en "actualizador.dominio.compañia.es".....	80
3.4	ANEXO IV: Modificar el servidor "maestro" del pool de XenServer.....	86



Proyecto de Virtualización de Aulas informáticas mediante el uso de Software Libre. Aplicación práctica en el Dpto. de Organización de Empresas by Francisco J. Villar, Raúl Blasco Herrera, Fernando Ferrer is licensed under a [Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional License](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Lista de figuras

Figura 1 Esquema virtualización basada en host local	7
Figura 2 Esquema virtualización basada en host remoto	8
Figura 3 En la Fase 3: trae tu propio dispositivo.	9
Figura 4 Esquema virtualización con hipervisor de tipo 2	10
Figura 5 Mostrar detalles de una instantánea en VirtualBox	16
Figura 6 Detalle del registro de cambios en las instantáneas de VirtualBox	16
Figura 7 Gráfica de la estructura lógica Fase 3	23
Figura 8 Gráfico de distribución elementos adquiridos en la Fase 2.	26
Figura 9 Agregando un servidor en XenCenter.	35
Figura 10 Crear un nuevo "Pool" en XenCenter.....	35
Figura 11 Cuadro de dialogo de XenCenter en la creación del "Pool"	36
Figura 12 Tras crear el "Pool" creamos el almacenamiento de datos.	37
Figura 13 Asistente de creación de nuevo almacenamiento de datos. Pantalla de inicio.	37
Figura 14 Asistente de creación SR. Introducción datos configuración.	38
Figura 15 Creación VM desde XenCenter.	39
Figura 16 Actualizar lista discos virtuales contenidos en un SR (XenCenter)	49
Figura 17 Gráfico ruta entre máquina física – máquina virtual.	57
Figura 18 Asistente de configuración de HA en XenCenter, página de inicio.	65
Figura 19 Selección de repositorio de datos (SR) para la función de Alta Disponibilidad (HA).	66
Figura 20 HA. Configuración de máquinas virtuales con Alta Disponibilidad habilitada.....	67
Figura 21 Pantalla de inicio de sesión.....	80
Figura 22 Pantalla gestión de laboratorios	81
Figura 23 Sección iptables.	81
Figura 24. Sección Selección de equipos.	82
Figura 25. Sección Ficheros.....	82
Figura 26. Pantalla SCP	83
Figura 27 Detalle de la Sección SCP, copia de ficheros a equipos.....	84
Figura 28 Pantalla Bróker.....	84
Figura 29 Pantalla de gestión de usuarios.	85
Figura 30. Pantalla listar accesos de usuarios.....	85

Lista de tablas

Tabla 1 Ventajas y desventajas de la Fase 1	10
Tabla 2 Modificación realizada en /etc/exports de “actualizador”	17
Tabla 3 Script para habilitar la respuesta al ping en el servidor de actualizaciones “actualizador”.	19
Tabla 4 Extracto con las modificaciones del fichero /etc/rc.d/rc.local de los equipos de laboratorio para iniciar copia nueva VM.	20
Tabla 5 Fichero “bootscript” de ejemplo con proceso de copia y/o actualización de VM.	21
Tabla 6 Elementos hardware adquiridos Fase 2.	25
Tabla 7 Redes creadas en la Fase 2	26
Tabla 8 Comparación ocupación espacio en disco para NFS vs iSCSI.....	27
Tabla 9 Ejemplo de /home/xen-scripts/CreaLAB-v3.sh para crear clones para laboratorios.	42
Tabla 10 Ficheros “csv” de ayuda en la creación de nuevos clones de VM’s de laboratorios.....	43
Tabla 11 Extracto de “rc.local” equipos laboratorios con detalle del establecimiento del vínculo con su VM.	52
Tabla 12 Archivo dhcpd.conf en “bróker”. Configuración servicio DHCP.....	53
Tabla 13 Extracto del script “DHCP+DNS.sh”	54
Tabla 14 Archivo dnsmasq.conf en “bróker”. Configuración servicio DNS.....	55
Tabla 15 Extracto del script DHCP+DNS.sh	56
Tabla 16 Script “túnel.sh”	59
Tabla 17 Script “nmap.sh”.	61
Tabla 18 Extracto del fichero rc.local: inclusión de entrada.key y nmap.key.....	62
Tabla 19 Contenido del script manejo unidades USB: vbox-automount-usb.sh	80

Abreviaturas

Abreviatura	Término
<i>VM</i>	Máquina Virtual (Virtual Machine)
<i>RDP</i>	Conexión de escritorio remoto (Remote Desktop Protocol)
<i>BYOD</i>	Bring your own device
<i>Pool</i>	Conjunto de recursos agrupados y disponibles para la infraestructura de virtualización (Servidores, almacén, etc.).
<i>VLAN</i>	Red virtual.

INTRODUCCIÓN.

El proyecto persigue dos objetivos principales: por un lado encontrar un método por el que podamos independizar hardware de software, de forma que un mismo conjunto de software o aplicaciones “encapsuladas” en un solo fichero, pueda usarse en todos los laboratorios del departamento, sea cual sea el hardware disponible y de otro, recoger la documentación necesaria para que esta experiencia sea sencilla y rápida de replicar por personal técnico de otros departamentos en sus propios laboratorios o aulas informáticas.

Identificamos tres componentes básicos en el reto que supone alcanzar los objetivos del apartado anterior:

1. Las aplicaciones necesarias para que los alumnos realicen sus prácticas en las aulas, las cuales necesitan un sistema operativo para funcionar y no todas estas aplicaciones se pueden ejecutar en todos los sistemas operativos disponibles.
2. Los propios sistemas operativos. Interfaz entre el hardware y las aplicaciones, su problema es que con el tiempo y uso se degrada su rendimiento, además dejan de recibir soporte y pasan a ser obsoletos.
3. Hardware, probablemente el más delicado y costoso, se estropean, se quedan obsoletos con el tiempo y además existe mucha variedad de modelos y características que para determinados sistemas operativos supone un auténtico galimatías de controladores y software auxiliar.

Estos componentes deben adaptarse entre sí para que el usuario pueda realizar su actividad, para ello hay que realizar distintas adaptaciones:

- Adaptar el sistema operativo al hardware en el que se quiere ejecutar ya sea por requisitos de espacio de memoria, por instalación de drivers y configuraciones del hardware específicas (tarjeta gráfica, tarjeta de red, placa base), por configuración de red y dispositivos USB.
- Adaptar el sistema operativo a las necesidades de las aplicaciones que el usuario utiliza.

Si todos los equipos tuviesen el mismo hardware, el mismo sistema operativo, y todas las aplicaciones requeridas funcionasen con el mismo sistema operativo, se podría utilizar una sola instalación configurada para todos los equipos.

Obviamente, en el caso del Departamento de Organización de Empresas esto no sucede así, como en otros muchos centros, se presentan más de 8 configuraciones de hardware distintas, 3 sistemas operativos distintos, y más de 50 aplicaciones necesarias en un total de 140 equipos, lo que implica aproximadamente 25 configuraciones diferentes que hay que mantener y gestionar de forma individualizada.

La adaptación de las aplicaciones a los sistemas operativos es factible, ya que en nuestro caso la mayoría de aplicaciones pueden funcionar en el mismo sistema operativo, pero la adaptación de este sistema operativo a la variedad de hardware disponible resulta inviable.

Con el fin de salvar estos inconvenientes se propone el uso de la tecnología de virtualización, la cual agrega un nuevo nivel de abstracción entre software y hardware que nos permite, entre otras muchas ventajas, tener un mismo sistema operativo con todas las aplicaciones de forma independiente al hardware del que dispongamos para ejecutarlo. Cabe señalar que una de las premisas a la hora de afrontar el proyecto, ha sido hacer uso de tecnologías y desarrollos software de uso gratuito. Este hecho ha supuesto un importante ahorro económico en licenciamiento software si bien ha requerido de un importante sobre-esfuerzo en horas de trabajo de las personas implicadas.

FASES DE DESARROLLO.

Se afronta el proyecto en tres fases claramente diferenciadas, acometiendo y recogiendo las dos primeras en el presente documento. Estas dos primeras fases, cerradas con la elaboración del presente documento, se desarrollaron: desde su etapa de investigación, hasta su implementación en los laboratorios del departamento, entre julio de 2014 y mayo de 2015.

- Fase 1: Virtualización en host local



Figura 1 Esquema virtualización basada en host local

El hardware tiene instalado un sistema operativo anfitrión “ligero” en el que se instala un software que realiza las tareas de virtualización con los recursos del propio equipo. Obtenemos algunas ventajas que aporta la virtualización (relativamente fácil de gestionar, independencia del hardware, reutilización de dispositivos), pero también con algunos inconvenientes, principalmente la reducción en el rendimiento general del equipo.

- Fase 2: Virtualización en host remoto:

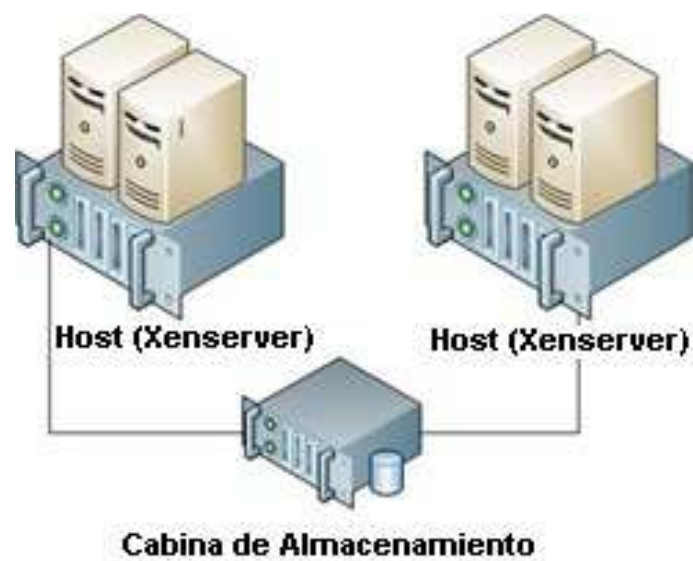


Figura 2 Esquema virtualización basada en host remoto

El equipo tiene instalado un cliente RDP (Conexión a Escritorio Remoto) que le permite acceder a un sistema operativo virtualizado que es accesible a través de la red. Este sistema virtualizado existe una infraestructura virtual o pool de servidores dedicados a ejecutar las máquinas virtuales.

- Fase 3 (en estudio): Virtualización con acceso desde cualquier localización (BYOD por su acrónimo en inglés de Bring your own device):



Figura 3 En la Fase 3: trae tu propio dispositivo.

En esta futura fase y una vez implementada y probada la fase 2, se plantea la ampliación del servicio prestado en los laboratorios permitiendo el acceso a equipos virtualizados de forma remota desde cualquier dispositivo y desde cualquier lugar. Esto permitirá mayor flexibilidad horaria para el acceso a las prácticas, reducirá la necesidad de mantenimiento y adquisición de puestos de trabajo y permitirá la personalización de los escritorios de trabajo para distintos perfiles de usuario. En el momento de la redacción de este texto, esta fase se encuentra en estudio.

Para cada fase describiremos en principio los detalles generales, y posteriormente entraremos a la descripción específica de nuestro caso práctico, detallando las configuraciones y los problemas con los que nos hemos ido encontrando (y generalmente resolviendo) durante su implantación en el Departamento de Organización de Empresas de la Universitat Politècnica de València.

1 Capítulo 1: IMPLEMENTACIÓN DE LA FASE 1.

1.1 Antecedentes

Cuando se inició el proyecto de virtualización de aulas, necesitábamos una solución rápida y fácil de implementar con la que empezar a centralizar la gestión de los laboratorios en el menor tiempo posible, que cuyo coste además, fuese el menor posible. Tal y como vimos en la figura 1 de la introducción, en esta fase usamos exclusivamente los recursos hardware de cada ordenador de laboratorio, con lo que no necesitamos la adquisición de ningún hardware adicional. La idea es que a medida que implantemos las siguientes fases, este modo de virtualizar quede como "backup" o solución de emergencia ante una posible caída en el sistema alojado en los servidores.

En la siguiente tabla resumimos las ventajas y desventajas de esta fase, los equipos siguen ofreciendo al usuario final exactamente el mismo software que tenían hasta ahora, pero a efectos de mantenimiento, todo está contenido en un solo fichero.

doe
Departamento de Organización de Empresas.

Laboratorios DOE. 1ª ETAPA

Ventajas	Desventajas
No requiere compra hardware adicional, excepto alguna ampliación RAM.	Se ejecuta con los recursos "locales", dependemos de procesador y ram local.
La actualización se hace de manera rápida por red.	Menor rendimiento en aplicaciones muy exigentes (simulación)
Un solo fichero contiene el equipo completo.	No reproduce audio en local. Bajo rendimiento video.
Misma configuración para todo el hardware.	Mantenemos ordenador profesor sin virtualizar.
Independencia de la red.	

Universidad Politécnica de Valencia - Camino de Vera, s.n. - 46.022 Valencia | 96 387 76 80 | 96 387 76 89

Tabla 1 Ventajas y desventajas de la Fase 1

La solución elegida se basa en la utilización de un hipervisor de tipo 2, es decir, un sistema de virtualización que se ejecuta sobre un sistema operativo:



Figura 4 Esquema virtualización con hipervisor de tipo 2

Siguiendo este esquema, los componentes software seleccionados han sido:

Sistema operativo anfitrión: Se ha seleccionado Linux CentOS 6.5 por ser un sistema operativo gratuito, con soporte de Red Hat y por su bajo consumo de recursos, lo que nos permite disponer de mayor cantidad de memoria RAM para la máquina virtual que va a ser ejecutada. Además es un operativo independiente del hardware, lo que nos permite poder utilizar la misma instalación en distintos tipos de máquinas, simplemente copiando de uno a otro.

Software de virtualización: VirtualBOX de Oracle, ofrece buen rendimiento, flexibilidad en la gestión y su uso también es gratuito. Sobre esta aplicación ejecutaremos una máquina virtual con S.O. Windows XP (con menos requerimientos de hardware) con todas las aplicaciones que las prácticas docentes del departamento necesitan.

En realidad se debe aclarar que el alumno o usuario de laboratorio no va a ejecutar nunca directamente esta máquina virtual, sino una "instantánea" de la misma. Aprovechamos la funcionalidad de las instantáneas de VirtualBox para generar, cada vez que iniciamos el ordenador, una copia o "clon" de la máquina virtual, "arrancando" posteriormente este clon. De esta forma, preservamos la máquina virtual base de los cambios indeseados que pudiera hacer el usuario. Por tanto al usuario siempre le presentamos un sistema "limpio" donde realizar sus prácticas.

El hecho es que el sistema no mantiene ningún dato entre reinicios. El usuario es informado de esta circunstancia mediante un mensaje presentado como fondo de escritorio de Windows y por medio de carteles en los laboratorios.

1.2 Instalación.

1.2.1 Consideraciones y requisitos previos

Para la instalación vamos a necesitar:

- Herramientas de particionado de discos (parted, fdisk), y extracción de sistemas de archivos (clonezilla, fsarchiver, dd). Usamos SystemRescueCD que lleva gparted y fsarchiver de serie.
- Distribución linux ligera (como Ubuntu, OpenSuse xfce, Fedora xfce) instalado en una máquina de pruebas con aplicación de virtualización (como virtualbox, qemu, kvm) ya instalado y configurado, y la máquina virtualizada configurada con las aplicaciones de uso en el aula. En nuestro caso usamos Linux CentOS, instalado en una máquina de pruebas con VirtualBox configurado con la máquina virtualizada y con las aplicaciones de uso en el aula.
- Soporte externo USB de almacenamiento para mover los datos desde el equipo de pruebas a los equipos de laboratorios destino.

En nuestros laboratorios hemos encontrado dos tipos de hardware, el equipamiento más antiguo no incluía en sus procesadores el conjunto de instrucciones para virtualización (VT-x), lo que nos obliga a hacer algunos ajustes según disponemos de esta funcionalidad o no. Hemos distinguido por tanto dos “bloques” generales de configuración que pueden ser de utilidad para el lector si comparte el caso de utilizar equipamiento con algunos años de antigüedad:

1.2.2 Preparación de los sistemas con VT-x

En caso de que los equipos destino tengan el mismo espacio en disco duro, la migración es tan sencilla como extraer una copia completa del disco duro de la máquina de pruebas donde hemos preparado todo y replicarla en los equipos destino, bien a mano o utilizando un sistema de distribución por red tipo Clonezilla o Transmission (Torrent)+dd.

En caso de que no sean posibles las copias completas de disco por imágenes, hay que realizar el particionado y la configuración del sistema de arranque a mano. Este es el caso de nuestro departamento, donde hay distintos tipos de equipos y para mantener la homogeneidad de la configuración se ha realizado un particionado específico del disco duro, siguiendo estos pasos:

Primero “extraemos” la imagen con el sistema desde el equipo de pruebas o “base”:

- Arrancar el equipo de pruebas con una distribución Linux, nosotros hemos usado SystemRescueCd, conectar un dispositivo externo y montarlo en el sistema, tecleamos:

```
mkdir /mnt/origen
```

```
mount /dev/sdb4 /mnt/origen
```

- Con la utilidad linux fsarchiver almacenamos la partición "/" en un dispositivo externo:

```
fsarchiver savefs /mnt/origen/hypervisor.fsa /dev/sda3
```

Con "*fsarchiver savefs*" ejecutamos la utilidad fsarchiver con el parámetro de almacenar el sistema de archivos

"/mnt/origen/hypervisor.fsa" es el contenedor donde queremos almacenar el sistema operativo completa del equipo “base”, todavía sin la máquina virtual.

"/dev/sdb4" es la partición origen en el equipo base de la que se pretende extraer el sistema de archivos.

Segundo, en los equipos de nueva instalación:

- Arrancar con el mismo "SystemRescueCd" y elegir la opción de arranque por defecto de boot (se detiene el arranque unos segundos e introducimos el código de teclado, teclear 13 para el código de teclado español). Al mostrar el prompt, iniciar el entorno gráfico ejecutando **startx**.
- Ejecutar el editor de particiones en un shell: **gparted &**

Gparted es una conocida herramienta gráfica de Linux para crear particiones en el disco duro del equipo. Hemos hecho en un sistema MBR todas de tipo primario (4 particiones):

1ª Partición: La creamos con 10 Gb por si en el futuro queremos instalar un Windows (o más grande, si el disco lo permite), será fat32, no necesita etiqueta. La opción del sistema Windows local se puede mantener por si hay algún problema con la virtualización local y remota.

2ª Partición: De 2Gb de tipo linux-swap.

3ª Partición: Aloja el sistema raíz "/" de Linux CentOS, le asignamos 10 Gb), tipo ext4.

4ª Partición: Con el resto del tamaño del disco duro, del tipo XFS, en la que posteriormente se montará "/virt" y servirá como almacén de las distintas máquinas virtualizadas que se vayan creando y utilizando las máquinas temporales que se puedan ir creando. Se elige un sistema XFS por su eficiencia, robustez y capacidad de ampliación.

- Montamos el dispositivo externo donde habíamos guardado el sistema con fsarchiver y lo montamos:

```
mkdir /mnt/origen
```

```
mount /dev/sdb4 /mnt/origen
```

- Descomprimos y volcamos el sistema con fsarchiver:
fsarchiver restfs /mnt/origen/hypervisor.fsa id=0,dest=/dev/sda3

"fsarchiver restfs" ejecuta el programa fsarchiver con el parámetro de recuperar filesystem.

"/mnt/origen/hypervisor.fsa" es la ruta al contenedor donde llevamos el sistema operativo del que habíamos extraído del equipo base, ese es el nombre de archivo comprimido.

"id=0,dest=/dev/sda3" el id=0 es el primer trozo, que podrían ser varias particiones las que contuviera el archivo comprimido y dest es la ruta de la partición sobre la que vamos a desplegar el filesystem.

- Podemos ejecutar, opcionalmente para comprobar el resultado:

```
fdisk -l /dev/sda #listamos las particiones del disco, la "/" tiene que tener "id=83".
```

- Preparamos el arranque del disco destino instalando GRUB:

Tecleamos **grub** y nos abre la consola de GRUB, ejecutamos:

```
root (hd0,2) #tabulamos 2 veces para que se autocomplete y vemos las particiones, elegimos la 2 que es de donde arrancará.
```

```
setup (hd0) #para que instale el GRUB en el MBR del disco.
```

```
quit # para salir de grub.
```

- Ya podemos arrancar desde el disco duro, pero solo funcionará la opción del modo MANTENIMIENTO, un modo consola al que accedemos desde una opción del menú de inicio.

- Iniciar sesión como root + contraseña.

- Iniciar el entorno gráfico con "startx" y copiar en "/virt" (la cuarta partición en formato XFS estará montada en esa carpeta) la máquina virtual así como el fichero "hypervisor.fsa" que como hemos visto es el que contiene todo el sistema operativo del equipo.

- Iniciamos VirtualBox y borramos las máquinas virtuales que hubiera (solo borrar, no eliminar ficheros), (es equivalente a des-registrarla).

- Volvemos a registrarla de nuevo desde la línea de comandos con:

```
VBoxManage registervm /virt/Nombre_VM/Nombre_VM.vbox
```

Una vez hemos terminado estos pasos, ya podemos reiniciar y arrancar normalmente.

1.2.3 Preparación de los sistemas sin VT-x

Como hemos mencionado anteriormente, hemos necesitado realizar ajustes en los equipos más antiguos del departamento cuya placa no tiene soporte para la arquitectura VT-x de virtualización. Dicho

ajuste consiste en modificar el fichero `/etc/rc.d/rc.local`, para añadir los siguientes parámetros que anulan el uso de esta tecnología:

```
--hwvirtex off --vtxvpid off --vtxux off --cpus 1
```

Quedando así:

```
/usr/bin/vboxmanage modifyvm Nom_linked_clon--bioslogodisplaytime 0 --bioslogofadein off --  
bioslogofadeout off --biosbootmenu disabled --hwvirtex off --vtxvpid off --vtxux off --cpus 1
```

El resto del proceso es idéntico al del apartado anterior.

1.3 Gestión y Actualización de las máquinas virtuales locales.

1.3.1 Preparación de la nueva máquina virtual.

Cuando necesitamos crear, hacer una modificación o añadir nuevo software a las máquinas virtuales que se ejecutan de forma local en los ordenadores de los laboratorios, comenzamos el proceso preparando una nueva versión de la máquina virtual a desplegar o instalar en cada uno de los discos duros. Para ello conviene que dispongamos de un ordenador en el que tengamos VirtualBox instalado y con hardware lo más compatible posible con la mayoría de los equipos que tenemos en nuestras instalaciones.

En el VirtualBox de este equipo arrancaremos la máquina virtual objeto de modificaciones, a la que nos vamos a referir como "modelo_base". Esta será la que tendremos que copiar luego en todos los ordenadores de los laboratorios para su utilización.

Una vez creada, implementados los cambios y/o añadido software, ejecutaremos los siguientes pasos previos a su "despliegue":

a) En VirtualBox, pinchamos en "Instantáneas" (snapshots). Recordemos el apartado "antecedentes" donde se explicaba el uso de instantáneas y como la máquina virtual que se ejecuta en los laboratorios en realidad es un clon para preservar la integridad de la máquina base. Dicha instantánea debe llamarse siempre sn01, ya que con ese nombre se lanza desde `/etc/rc.d/rc.local`.

b) Tenemos que generar una instantánea nueva que recoja los cambios que acabamos de implementar en nuestra máquina virtual "modelo_base", pero ya que la instantánea que se ejecuta siempre se tiene que llamar "sn01", necesitamos antes de generar una nueva con el software recién instalado liberar este nombre de la anterior "sn01". Para ello pulsamos con el botón derecho sobre la instantánea sn01 y pinchamos en la opción mostrar detalles:

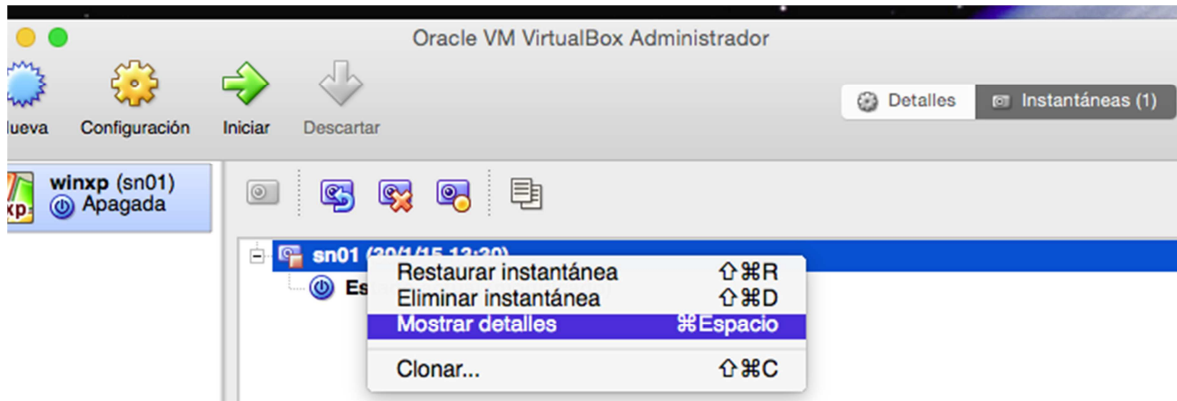


Figura 5 Mostrar detalles de una instantánea en VirtualBox

En la ventana de detalles de la instantánea, cambiamos el nombre sn01 por otro. El criterio que estamos siguiendo para asignar estos nombres, es el de poner la fecha de creación original de esa instantánea en formato año/mes/día (yyyy/mm/dd), este dato lo podemos encontrar en el apartado detalles. Una vez "liberado" el nombre sn01 podemos proceder a generar la nueva instantánea a la que nombraremos "sn01" y en cuyo apartado "detalles" procuraremos describir los cambios realizados a modo de registro, podremos copiar-pegar de la anterior instantánea los cambios anteriores para que no se pierda nada. De esta forma llevamos un registro manual de cada modificación sobre una máquina virtual concreta.

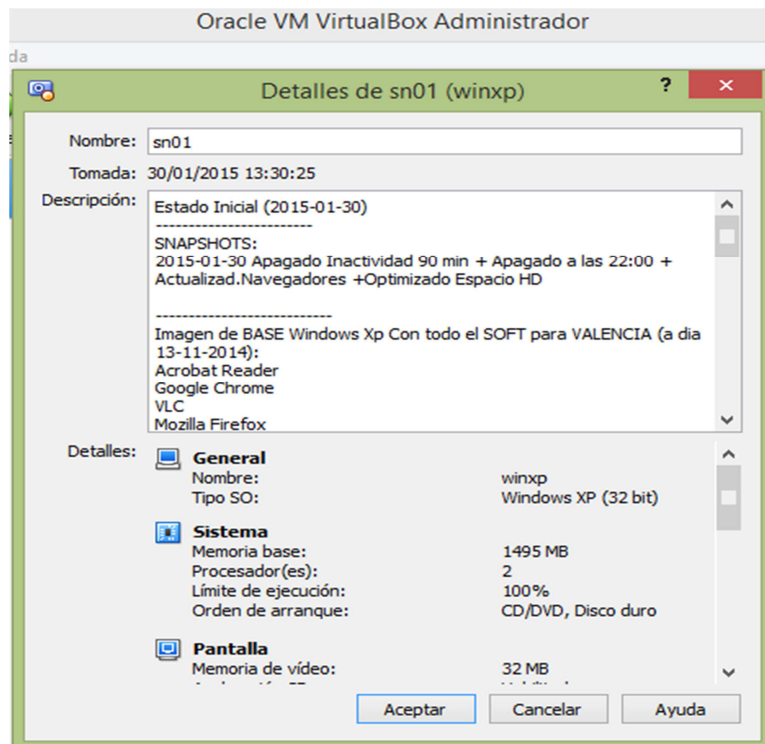


Figura 6 Detalle del registro de cambios en las instantáneas de VirtualBox

Terminada la preparación de nuestra nueva máquina virtual "modelo_base", o mejor dicho de su instantánea "sn01", podremos copiarla a cada uno de los ordenadores de los distintos laboratorios.

1.3.2 Servidor de actualización de máquinas virtuales locales: "actualizador".

Una vez terminado el proceso del apartado anterior de preparación del "modelo_base", se ha diseñado un procedimiento para, en la medida de lo posible, facilitar al máximo su despliegue en todos los equipos de los laboratorios. Para ello nos apoyamos en un servidor que hemos llamado "actualizador", del que nos valdremos para realizar una gestión de forma centralizada y cómoda de las actualizaciones de las máquinas virtuales que usamos de forma local en los ordenadores de los laboratorios.

1.3.2.1 Configuración de "actualizador":

Hemos instalado un servidor virtual con S.O. Centos 6.6, desde este servidor, exportamos mediante protocolo NFS un recurso compartido, donde estará lo que tenemos que copiar a los equipos de laboratorio. Además exportamos un par de recursos que hemos llamado /virt/Almacen de Snapshots/ donde guardaremos un histórico de máquinas virtuales.

Instalaremos el protocolo NFS y lo activamos:

```
yum install nfs nfs-tools
```

```
chkconfig nfs on
```

Configuramos el servicio para que exporte o comparta las carpetas deseadas modificando "/etc/exports". Debería quedarnos de esta forma:

```
#Acceso solo para nuestra Subred de AULAS(158.42.x.x .. hasta .. 158.42.x.x)  
/virt/vm-actualizando/ 158.42.x.0/23(ro1,async) 158.42.x.0/23(ro,async)  
#Acceso solo para Subred del dpto. (158.42.x.x .. hasta ..158.42.x.x)  
/virt/ 158.42.x.0/23(rw,no_root_squash2)
```

Tabla 2 Modificación realizada en /etc/exports de "actualizador"

¹ En la Subred de AULAS solo necesitamos exportar en modo "solo lectura" = ro.

² Como se puede apreciar en la segunda línea se pone la opción "no_root_squash" cuya función que que el usuario "root" remoto tenga permisos de edición como si fuera "root" local.

Con el siguiente comando aplicamos los cambios:

exportfs -ra

Instalación de Apache y PHP-MySQL y activación de servicio.

yum install httpd php php-mysql

chkconfig httpd on

Creamos el script en `"/var/www/html/scripts/"` y reiniciamos

service httpd restart

1.3.2.2 Control de acceso de los equipos a actualizar.

Una vez tenemos configurado este servidor de actualización, el proceso de “despliegue” se pondrá en marcha o no dependiendo de si el equipo que queremos que reciba la actualización tiene acceso a “actualizador” o no.

Cuando se encienden los equipos de los laboratorios, realizan un proceso automático que intenta comunicarse con el servidor de actualizaciones “actualizador”, si reciben respuesta se iniciará el proceso de instalación de una VM nueva, en caso contrario, el equipo iniciará con normalidad.

Para que todo esto suceda, hemos utilizado el comando “ping” combinado con reglas configuradas en el cortafuegos del “actualizador” (iptables). Los técnicos encargados deciden si se lleva a cabo la actualización o no simplemente permitiendo o no que los equipos que deseamos actualizar sean capaces de recibir respuesta al ping que hacen a “actualizador”.

Por defecto, el servidor “actualizador” rechaza todas las peticiones con esta regla que tiene implementada en su cortafuegos:

-A INPUT -p icmp -j REJECT --reject-with icmp-port-unreachable

Si queremos que “responda”, primero hay que eliminar las líneas de rechazo con:

iptables -D INPUT -p icmp -j REJECT --reject-with icmp-port-unreachable

Para ir añadiendo una línea con cada máquina a la que queremos que responda:

iptables -A INPUT -s x.x.x.x -p ICMP --icmp-type 8 -j ACCEPT;

Y se agrega otra regla para que bloquee el resto de máquinas:

```
iptables -A INPUT -p icmp -j REJECT --reject-with icmp-port-unreachable
```

Guardamos los cambios con:

```
iptables-save
```

Con el script siguiente habilitamos la respuesta al ping de las direcciones IP que están almacenadas en un fichero pasado como parámetro, y bloquea el resto:

```
for line in $(cat $1);  
  
do  
  
iptables -A INPUT -s $line -p ICMP --icmp-type 8 -j ACCEPT;  
  
done  
  
iptables -A INPUT -p icmp -m icmp --icmp-type 8 -j REJECT --reject-with icmp-port-  
unreachable  
  
iptables-save;
```

Tabla 3 Script para habilitar la respuesta al ping en el servidor de actualizaciones “actualizador”.

Cuando terminamos el proceso de actualizar equipos de laboratorio, se vuelve a establecer las reglas del cortafuegos a su estado por defecto, de tal manera que ya no responda a los pings de ningún equipo y por tanto no se inicie de nuevo el proceso de actualización:

```
service iptables restart
```

1.3.2.3 Gestión web del proceso de actualización.

Todo el proceso de actualización descrito en el apartado anterior, se ha simplificado en buena medida y hecho más “amigable” mediante el desarrollo de una aplicación web que se encarga, entre otras cosas, de la modificación en el cortafuegos del servidor de actualizaciones.

Para ver la funcionalidad con detenimiento de esta aplicación web se puede consultar el Anexo III de este documento.

1.3.3 Despliegue de una nueva máquina virtual en el laboratorio.

Una vez hemos preparado la nueva máquina virtual y hemos seleccionado que equipos vamos a actualizar en el servidor de actualizaciones, solo nos falta el proceso de copia propiamente dicho.

Los pasos a dar para que los equipos de los laboratorios ejecuten las acciones necesarias al arrancar y reciban la actualización son:

1º) Nos aseguramos de que el equipo tenga su tarjeta de red operativa y pueda resolver el nombre del “actualizador” con:

```
Nslookup actualizador.dominio.compañia.es
```

2º) Si el equipo de laboratorio recibe respuesta al “ping” desde el servidor de actualizaciones se descarga un fichero desde allí (ConfigHypervisor.sh) que lo guarda en el directorio /root con el nombre “bootscript”, este fichero es el que contiene las instrucciones que realizará la actualización o copia.

```
ping actualizador.dominio.compañia.es -c 1 -w 1  
if [ $?=='0']; then  
# Descarga fichero instrucciones actualización desde el servidor  
wget --no-check-certificate -q -O /root/bootscript  
https://actualizador.dominio.compañia.es/rsync/scripts/ConfigHypervisor.sh  
# Ejecuta el fichero de instrucciones de actualización:  
sh -x /root/bootscript  
#Después de ejecutarlo lo elimina:  
rm -f /root/bootscript  
fi
```

Tabla 4 Extracto con las modificaciones del fichero /etc/rc.d/rc.local de los equipos de laboratorio para iniciar copia nueva VM.

Como vemos en la Tabla 4, una vez ejecutado “bootscript” lo elimina para que no exista la próxima vez que realicemos una actualización. La siguiente tabla muestra un contenido de ejemplo del fichero “bootscript” que se descargan y que realiza la actualización de la máquina virtual local:

#Des-registrar y eliminar los linked-clones:

vboxmanage unregistervm Nom_linked_clon --delete

#Esta línea evita un MAL BORRADO del CLON

rm -R -f /virt/Nom_linked_clon

#Des-registrar y eliminar la máquina virtual

vboxmanage unregistervm Nombre_VM

#Montar un recurso NFS almacén con VMs nuevas

mkdir /mnt/virt

mount -t nfs -o vers=3,rw,hard,bg,intr,nolock actualizador.dominio.compañia.es:/virt/vm-actualizando/ /mnt/virt

#Realizar la copia con rsync desde el recurso NFS.

rsync -avz /mnt/virt/ /virt/

#Registrar la VM recién copiada:

vboxmanage registervm /virt/Nombre_VM/Nombre_VM.vbox

Desmontar recurso:

umount /mnt/virt

Tabla 5 Fichero “bootscrip” de ejemplo con proceso de copia y/o actualización de VM.

Cuando termina la ejecución de este fichero, termina la actualización y podremos reiniciar los equipos para comprobar que se ha realizado correctamente, en nuestro caso en el fondo de escritorio vamos cambiando una leyenda que indica el número de versión de la máquina virtual a modo de control.

2 Capítulo 2: IMPLEMENTACIÓN DE LA FASE 2.

2.1 Descripción de la arquitectura elegida.

Una vez implementada la fase inicial del proyecto, en la que como vimos utilizamos la virtualización de forma local para dar una solución rápida al problema que teníamos en la gestión de los laboratorios, dimos paso a la implementación de la fase 2, en la que conseguiremos centralizar aún más la gestión de los laboratorios.

En esta fase se “trasladan” las máquinas virtuales, que en la fase anterior estaban alojadas en el disco duro de cada ordenador de laboratorio, hasta un conjunto o “pool” de servidores. Los equipos físicos de los laboratorios van a pasar a convertirse en una especie de “clientes ligeros” (del inglés: thin-clients), cuya única finalidad es servir de puerta de acceso a esas máquinas virtuales alojadas en los servidores del departamento. Por tanto, la instalación de software que van a necesitar los equipos cliente es mínima: aprovechamos el sistema operativo Linux Centos que tienen instalado desde la primera fase del proyecto y se añade una opción en su menú de inicio para cargar un “Cliente de Escritorio Remoto” (protocolo RDP de Microsoft Windows) con el que conectar con las máquinas virtuales de la infraestructura.

La virtualización local se mantiene, pero su papel pasa a ser de “backup” preparado para entrar en acción ante alguna incidencia en los servidores.

Con el planteamiento expuesto, necesitamos algún mecanismo que nos vincule cada PC de laboratorio con una máquina virtual de las que ahora existen en la infraestructura. En su versión más simple y en lenguaje de virtualización, al componente software que se encarga de esta “puesta en contacto” entre máquina virtual y máquina real se denomina habitualmente: “bróker”.

En nuestro proyecto el papel del “bróker” ha ido evolucionando y ganando complejidad, podemos distinguir dos etapas:

[“Bróker” estático o v1.0](#), una primera versión en la que simplemente mantenía unidos un PC físico de laboratorio con una máquina virtual, cada uno con la suya, y la máquina virtual arrancada y esperando a que el PC del mundo real se le conectara para trabajar... y:

[“Bróker” dinámico o v2.0](#), la evolución del primero cuyas mejoras más interesantes son el hecho de que ya no necesita que la máquina virtual permanezca siempre arrancada sino que lo hace bajo demanda, permitiendo además vincular más de una VM con cada PC físico.

Aunque su cometido resulta sencillo de explicar, su implementación merece un extenso desarrollo que veremos de forma específica en el presente documento.

En una futura Fase 3, el papel de este componente será más complejo y tendrá que encargarse también (o delegar en algún otro software para que lo haga) de validar el usuario, por lo que ya no se limitará a vincular PC's físicos con la infraestructura sino que deberá gestionar usuarios y ofrecerles las máquinas virtuales que los administradores decidan que puede utilizar.

2.1.1 Estructura lógica.

La arquitectura planteada en el apartado anterior se implementa en su forma lógica, con la distribución que podemos ver en la siguiente figura, donde tenemos una visión general que recoge todas las partes que intervienen en esta fase del proyecto:

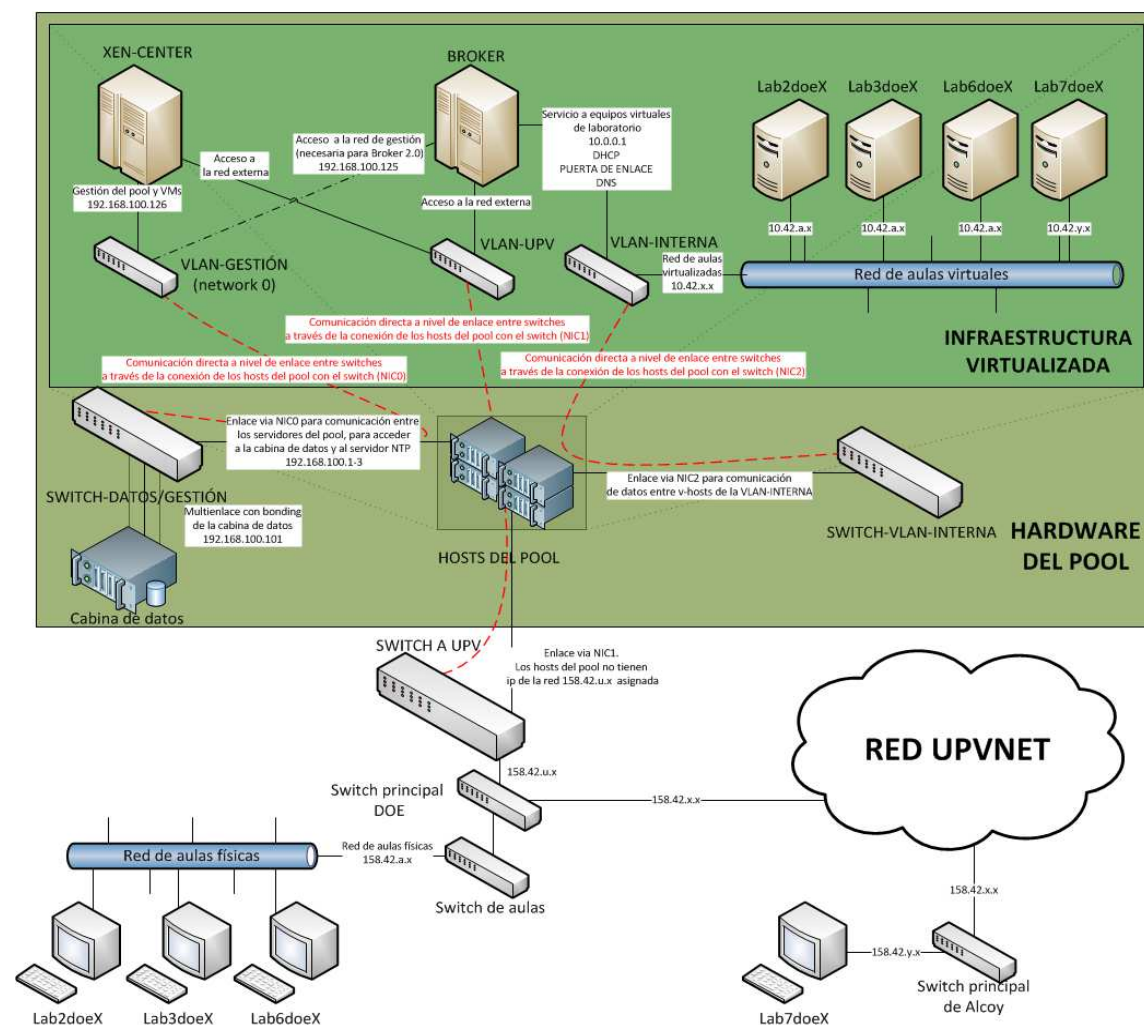


Figura 7 Gráfica de la estructura lógica Fase 3

En esta gráfica, el área con fondo verde oscuro muestra los componentes que forman parte de la infraestructura virtual. La zona verde claro: el hardware que compone el “pool” o grupo de servidores (y el dispositivo de red para comunicarse entre ellos), y en la zona exterior (fondo blanco), el resto de dispositivos de la infraestructura del sistema (cabina de almacenamiento y switches de red para acceso a la red interna y al exterior).

Veamos los componentes de esta configuración:

- **Pool de servidores** con XenServer 6.5. XenServer está basado en Xen-Project, una completa plataforma open-software de virtualización en la que se basa el producto comercial Citrix Xenserver, ofreciendo prácticamente las mismas prestaciones pero sin soporte técnico. El pool de servidores será un conjunto de recursos listos para ser usados por la infraestructura de virtualización, en nuestro caso se compone de:
 - Los propios servidores.
 - Los almacenes de datos o SR (storage resources) que les ofrece a los servidores la cabina de almacenamiento de datos.
 - El switch virtual o "Vlan-UPV" asociado a la segunda tarjeta de red de los servidores.
 - El switch virtual o "Vlan-interna" asociado a la tercera tarjeta de red de los servidores.
 - El switch virtual: “network0” (Vlan-Gestión) asociado a la primera tarjeta de red de los servidores y que se crea por defecto al crear el pool en XenServer.
 - Una máquina virtual con sistema operativo Windows y un adaptador de red en cada una de las anteriores redes virtuales que hemos definido. Su función es administrar la infraestructura de XenServer por lo que tendrá XenCenter instalado (el cliente gráfico de gestión de XenServer).
 - Otra máquina virtual con sistema operativo Linux Centos y adaptador de red conectado en la red pública de la UPV y otro conectado a la red interna para realizar tareas de “bróker”, mencionado anteriormente.
 - Un servicio web instalado en un servidor virtual llamado “actualizador” con un desarrollo web propio que nos va a servir para facilitar y/o automatizar algunas tareas y cuya funcionalidad ya empezamos mencionar en el capítulo anterior.

- **Cabina de almacenamiento**, configurada para ofrecer datos al pool usando el protocolo de nivel de aplicación NFS (Network File System). En cualquier documento sobre buenas prácticas, el almacenamiento de las máquinas virtuales (VM's) aparece separado físicamente de los servidores. La ventaja de esta desvinculación, es dotar de independencia a las máquinas virtuales, de esta forma pueden ser iniciadas en cualquier servidor del pool. Tengamos en cuenta que el sistema de virtualización “Xen” (al igual que otros) dispone de un algoritmo por el que buscará aquel hardware en el que tenga menos carga de trabajo en ese momento para ir iniciando máquinas virtuales.

2.1.2 Infraestructura hardware.

Para poder implementar la estructura lógica que hemos definido en el apartado anterior ha sido necesaria la adquisición de nuevos elementos hardware que se describen brevemente en la siguiente tabla:

Uds.	Descripción:
3	Servidor HP Dual Xeon E5-2620v3 (24 cores) 128 Gb RAM
1	Cabina almacenamiento Asus RS12 30Tb.
1	Switch (A) 24-Port 10/100/1000Mbps. Servirá para conectar los servidores y la cabina de almacenamiento. Red 192.168.x.x., la identificamos como red de almacenamiento.
1	Switch (B) 24-Port 10/100/1000Mbps. En el que se puedan configurar vlans. Una vlan estará conectada a la red externa (red 158.42.x.x) y servirá para tener acceso a la red externa en las máquinas físicas y virtuales (en principio sólo las virtuales tendrán acceso), otra vlan servirá para separar el tráfico de la vlan interna entre máquinas virtualizadas (red 10.x.x.x)

Tabla 6 Elementos hardware adquiridos Fase 2.

En la siguiente tabla mostramos las tres redes que se han creado con ellos, así como su distribución y segmentación:

Nombre red:	Componentes:	Funcionalidad:	Rango direcciones:
Almacenamiento (network0 o de gestión).	Switch A + NIC0 de servidores HP + Cabina almacenamiento Asus	Crear una red privada entre los servidores y la cabina de discos.	192.168.0.1 to 192.168.255.255
VLAN-Interna	Switch B (VLAN ports 13-24) + NIC2 de servidores HP	Red de direccionamiento privado para máquinas virtuales de los laboratorios.	10.0.0.1 to 10.255.255.255

VLAN-UPV	Switch B (VLAN ports 1-12) + NIC1 de servidores HP + Toma de conexión UPVNET.	Red de direccionamiento público unida con UPVNET	158.42.XXX.XXX
----------	---	--	----------------

Tabla 7 Redes creadas en la Fase 2

El siguiente gráfico muestra de una forma visual los componentes de las anteriores tablas:

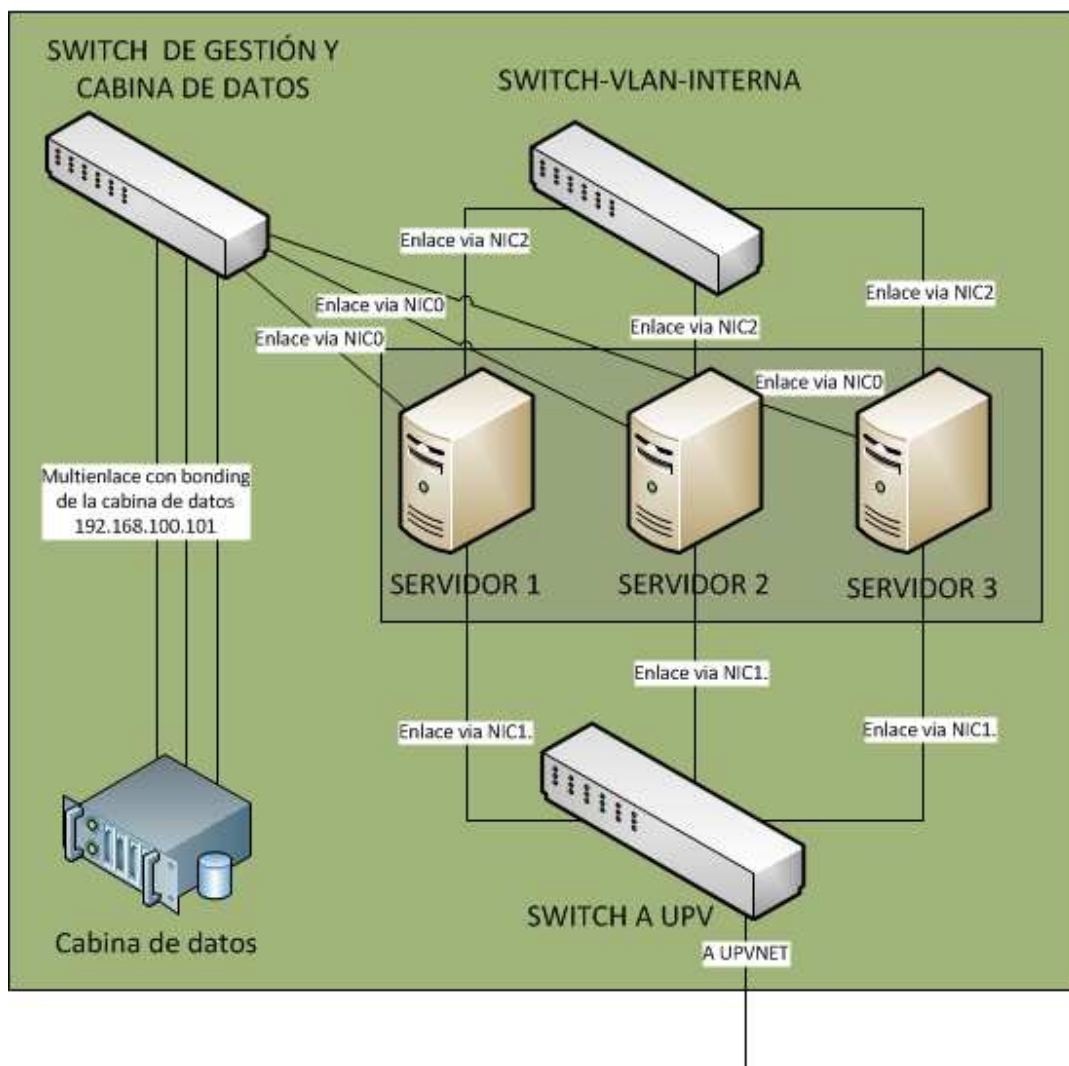


Figura 8 Gráfico de distribución elementos adquiridos en la Fase 2.

2.2 Instalación de la Cabina de Almacenamiento.

Como hemos visto en apartados anteriores, el objetivo es tener las máquinas virtuales y la mayor parte de los servicios en un almacén remoto de forma que podamos levantar un servicio o máquina virtual en cualquiera de los servidores de manera cómoda, inmediata y con garantías de obtener un buen rendimiento. Para llevar a cabo esto necesitamos que todos nuestros servidores puedan acceder a un almacenamiento de manera concurrente. Los protocolos que nos ayudan a conseguirlo pueden ser NFS o iSCSI.

Entre ambos, probablemente el que mejor se ajusta a nuestras necesidades es NFS, la razón es por el hecho de que al crear máquinas virtuales únicamente se utiliza el espacio real que ocupa la máquina mientras que en iSCSI, la máquina ocuparía todo el espacio provisionado o dicho de otro modo, todo el espacio reservado para ella y no solo el ocupado.

Ejemplo: Vamos a crear una máquina virtual con Windows7 y un disco duro virtual de 40 Gb., una vez instalado su sistema operativo ocupa 7 Gb.

	Tamaño provisionado Máximo	Tamaño ocupado en windows	Tamaño Real que OCUPA el Fichero en el Sistema de Almacenamiento
NFS	40 GB	7 GB	7 GB
iSCSI	40 GB	7 GB	40 GB

Tabla 8 Comparación ocupación espacio en disco para NFS vs iSCSI

Como vemos en este ejemplo, con iSCSI perdemos mucho espacio ya que iremos llenando el almacén aunque no lo usemos a media que creamos nuevas máquinas virtuales. A pesar de ello, esta característica puede resultar muy útil en otros casos, por ejemplo, cuando necesitamos crear máquinas/servicios CRÍTICOS a los que queramos garantizar que van a disponer de espacio en disco suficiente.

En nuestro caso lo que necesitamos **maximizar** es el número de máquinas virtuales disponibles para usar en nuestros laboratorios, teniendo en cuenta que el espacio que ocupa cada una apenas va a crecer. Nuestros usuarios (Alumnos), únicamente van a usar los programas ya instalados por nosotros y no añadirán más datos, con lo que sabemos a priori cuales son nuestras necesidades de espacio para almacenamiento.

2.2.1 Configuración almacenamiento en Asus RS12, caso particular DOE.

En el caso del departamento de Organización de Empresas se ha adquirido un producto de almacenamiento de gama profesional Asus RS12, con la siguiente configuración de discos:

- RAID1 con capacidad de 3Tb para sistema y almacenamiento de herramientas de los administradores.
- RAID5 con capacidad de 30Tb con los discos restantes (una vez montado el RAID5 nos quedan 24Tb. disponibles para almacenamiento de máquinas virtuales.

La BIOS de estos productos lleva un asistente de gran ayuda para la inicialización de los discos. Los pasos que hemos realizado han sido:

1º. Instalar en el RAID1 de 3Tb. (/dev/sda) el sistema operativo Linux Centos (en este caso se ha elegido la última versión del momento, 7.0)

2º. Instalar en el RAID5 de 30Tb. (/dev/sdb) el almacenamiento propiamente dicho para las máquinas virtuales mediante el uso de LVM (volúmenes lógicos).

2.2.2 Instalación de Centos 7 server “minimal” en la cabina de almacenamiento.

La BIOS de estos productos ya es UEFI, por lo que el instalador de Centos daba una alerta si intentábamos particionar los discos utilizando el modelo clásico de Master Boot Record (MBR), se ha usado particionado GPT y hemos creado una pequeña partición para el arranque de apenas 100 Mb.

Se ha realizado el siguiente particionado del disco en RAID1:

- Partición arranque de 100 Mb. Bios Boot (/dev/sda1)
- Partición Swap de 4Gb. (/dev/sda2)
- Partición “/” de 20Gb. (/dev/sda3)
- Partición con el resto de espacio disponible para almacenar imágenes ISOS que nos harán falta más adelante, casi 3Tb. (/dev/sda4).

Realizamos una instalación típica y configuramos conforme a nuestras necesidades (usuario root, contraseña, DHCP,...). Después de reiniciar la versión “minimal” no dispone de algunos paquetes que tendremos que añadir de forma manual para, entre otras cosas, disponer de las utilidades que necesitamos para exportar volúmenes NFS.

Teclamos:

```
yum install vim vim-X11 vim-common vim-enhanced vim-minimal nfs-utils
```

También debemos configurar el protocolo seguro “ssh” para disponer de acceso remoto al dispositivo, editamos el archivo de configuración del servicio:

```
vim /etc/ssh/sshd_config
```

... y le cambiamos el parámetro de acceso por sftp para poder transferir datos al almacenamiento desde nuestros pc's":

```
Subsystem sftp internal-sftp
```

...después de reiniciar los servicios, ya podremos acceder.

2.2.3 Configurar recursos NFS mediante LVM

Como hemos visto en el apartado 1.1, la cabina de almacenamiento dispone de 2 RAIDs independientes, uno que se preparó para alojar el Sistema Operativo y como almacén de imágenes de sistemas operativos (ISO's) en /dev/sda, y el otro, que sería el almacén propiamente dicho: el que contendrá las máquinas virtuales (/dev/sdb). Después de configurar el disco de sistema en el punto anterior, en este apartado vamos a configurar este disco de almacenamiento (/dev/sdb).

Para ello vamos a utilizar volúmenes lógicos “LVM”, gracias a los cuales podemos ir agregando en caliente volúmenes según vayan creciendo nuestras necesidades, para lo cual no necesitamos utilizar toda la capacidad de almacenamiento de la cabina desde el principio. Empezamos inicialmente creando uno solo con 2Tb., si más adelante queremos tener más espacio disponible, bastará con añadir un nuevo volumen, siguiendo desde el paso 5 de la guía que se muestra a continuación:

Paso 0. Marcar la partición con un label gpt:

```
parted -a optimal /dev/sdb mklabel gpt
```

Paso 1. Se ha creado una partición GPT:

```
parted -a optimal /dev/sdb mkpart primary 0% 100%
```

Paso 2. Activarlo:

```
parted /dev/sdb set 1 lvm on
```

Paso 3. Visualizamos si hay algun pv (physical volume):

```
pvdisplay
```

Paso 4. Se ha creado un grupo de volúmenes con toda la capacidad del nodo:

```
vgcreate VG_ALMACEN /dev/sdb1
```

Paso 4.1 Lo visualizamos con las órdenes:

```
vgs
```

```
pvdisplay
```

Paso 5. Las LUNs se corresponden a lvms, ejemplo de 2TB:

```
lvcreate -L 2T -n LV_VMS VG_ALMACEN
```

Paso 5.1 Para visualizarlo usamos:

```
lvs
```

```
LV      VG      Attr      LSize Pool Origin Data% Move Log Cpy%Sync Convert
LV  VMS      VG_ALMACEN -wi-ao---- 2,00t
```

Paso 6. Formateamos el volumen lógico utilizando el sistema de ficheros XFS (recomendado por RedHat para “Glusterfs” o simplemente para servirlo por NFS):

```
mkfs.xfs -i size=512 -f /dev/VG_ALMACEN/LV_VMS
```

Paso 7. Creamos el directorio /VMS/LV01 y lo montaremos en el “fstab”. El contenido del fichero /etc/fstab quedará así:

```
/dev/VG_ALMACEN/LV_VMS /VMS/LV01 xfs defaults 1 1
```

Paso 8. Ahora lo exportamos para que sea accesible por “NFS” editando y añadiendo en el fichero /etc/exports:

```
/VMS/LV01 IP_Servidor_Destino_1(rw,async,no_root_squash)  
IP_Servidor_Destino_2(rw,async,no_root_squash)  
  
/isos IP_Servidor_Destino_1(rw,async) IP_Servidor_Destino_2(rw,async)3
```

Paso 9. Actualizamos:

```
exportfs -ra
```

Paso 10. Ver servicios activos:

```
Systemctl
```

Paso 11. Desactivamos firewall:

```
service firewalld status  
  
service firewalld stop  
  
systemctl disable firewalld.service
```

Paso 12. Activamos el servicio NFS:

```
service nfs status  
  
service nfs start  
  
systemctl enable nfs-server.service
```

³ La línea de “/isos” también exporta, en este caso es el directorio donde alojamos las imágenes que usaremos para instalar los distintos sistemas operativos a virtualizar.

2.2.4 Supervisión y control de los Discos del Raid

Para asegurarnos de que si falla algún disco de la cabina de almacenamiento, lo podamos sustituir lo antes posible, debemos disponer de un mecanismo por el cual en el momento que se produzca el fallo se nos reporte inmediatamente a los técnicos responsables.

Con la cabina se nos ha suministrado software de utilidades y controladores del fabricante del adaptador LSI MegaRaid que incorpora y que han sido instalados en la carpeta /root/SOFT, pero no resulta suficiente para nuestro propósito de monitorizar los discos, tendremos que configurar el script: "**Isi.sh**" (obtenido gracias a: https://calomel.org/megacli_lsi_commands.html) que mediante el uso de la opción "checkNemail" chequea los volúmenes y manda un email solo si encuentra algún error.

Este script se ejecuta automáticamente cada hora gracias a la utilidad "crontab". Así queda su programación:

```
crontab -l
```

```
@hourly /root/SOFT/Isi.sh checkNemail
```

El contenido del script "**Isi.sh**" se muestra en el Anexo I: "Fichero script "Isi". Lo más importante respecto a este script es que recordemos que es necesario modificar la dirección de correo electrónico a quien envía el aviso:

```
EMAIL="tecnicos@dominio.compañia.es"
```

...y en la línea donde se llama al programa de envío de correo:

```
$MegaCli -PDlist -aALL -NoLog | egrep 'Slot/state' | awk '/Slot/{if (x)print x;x="";}{x=(!x)?$0:x" -  
"$0;}END{print x;}' | sed 's/Firmware state://g' | mail -r tecnicos@dominio.compañia.es -s `hostname` -  
RAID Notification' $EMAIL
```

2.3 Instalación de los servidores con XenServer

2.3.1 Instalación de un servidor con XenServer 6.5:

Si bien en el comienzo de la implementación de la fase 2 de este proyecto, la versión disponible de XenServer era la 6.2 y la instalamos en los primeros servidores para poder comenzar a hacer pruebas, muy poco tiempo después se liberó la versión 6.5 y actualizamos rápidamente, el tercer servidor incorporado al "pool" ya se instaló directamente con la versión 6.5. Dado que la instalación y cambio de versiones resulta bastante intuitiva, no hemos documentado las tareas de instalación de la versión 6.2 y posterior actualización a la 6.5 para no extendernos de forma innecesaria. A fecha de hoy la instalación se haría directamente con una versión 6.5 o superior siguiendo estos pasos:

1. Arrancamos desde el CD de instalación cuya ISO nos hemos descargado previamente de <http://xenserver.org/open-source-virtualization-download.html> y seleccionamos el destino para la instalación.
2. Seguimos el asistente, nos ofrecerá cargar un driver de acceso a disco por si el hardware así lo requiere, aceptamos el acuerdo de licencia (EULA) y nos muestra los discos duros que tenemos disponibles para realizar la instalación, elegimos el deseado y no marcamos "Enable thin provisioning...", esta opción es apropiada cuando el almacenamiento de las máquinas virtuales se hace localmente en el propio servidor, pero como vimos en el apartado [1.1](#) están almacenadas en la cabina de almacenamiento.
3. Elegimos "Local Media" como origen de la instalación para que la tome desde nuestro CD-Rom. A la pregunta de si nos gustaría instalar algún "supplemental pack" respondemos que no, al igual que podemos saltar la verificación del CD de instalación para ahorrar tiempo.
4. Necesitamos asignar una contraseña al usuario root para XenServer.
5. Nos preguntará, de las tarjetas de red que ha detectado la instalación en nuestro sistema, cual queremos asignar a la red de gestión "[network0](#)", nosotros hemos elegido la primera detectada (eth0) por costumbre de nuestra anterior infraestructura de virtualización con Vmware.

6. Configuramos la red del servidor de forma estática, dentro de la red de gestión este servidor será el primero: 192.168.100.1 y lo encamina dentro de esta red privada con la cabina de almacenamiento y los otros servidores. En la ventana siguiente lo "bautizamos" con el nombre que deseemos y de servidor de nombres DNS= 192.168.100.101 que es al cabina de almacenamiento y nos sirve a la vez como almacén de imágenes.
7. Configuramos un NTP (Network Time Protocol⁴) server para que le asigne la hora. En nuestra cabina de almacenamiento hemos implementado esa función, por tanto introducimos la dirección IP de la misma 192.168.100.101 como NTP Server 1.
8. Reiniciar el sistema. Con el hardware utilizado en nuestro caso particular tuvimos un problema en este reinicio: no arrancaba el sistema y se tuvo que modificar el fichero de configuración de /boot/extlinux.conf, comentando o borrando la línea "serial 0 115200" pues se quedaba parado el arranque esperando alguna señal en ese puerto serie. En un tercer servidor no se presentó este problema (hardware ligeramente distinto) y no fue necesario modificar la línea donde se detenía el arranque.

2.3.2 Instalación y configuración de XenCenter.

XenCenter es la aplicación gráfica de control y administración del entorno de virtualización de XenServer. Desde él podremos realizar tareas sobre nuestra infraestructura Xen de una forma intuitiva y sencilla, si bien algunas tareas sólo pueden llevarse a cabo desde la potente línea de comandos de XE-CLI.

La instalación de Open XenManager que sería el equivalente de XenCenter para Linux requiere de su descarga desde:

http://sourceforge.net/projects/openxenmanager/files/openxenmanager_rev6.tar.gz/download

Este proyecto para su instalación en Linux está en etapas tempranas de desarrollo, por lo que la versión que utilizaremos será la de Windows.

La instalación de XenCenter bajo sistema operativo Windows, comienza con su descarga desde: <http://xenserver.org/open-source-virtualization-download.html>

Una vez instalado, tendremos que asociar nuestros servidores XenServer con la opción: "ADD a Server" indicando el nombre o la dirección IP, así como las credenciales de los mismos.

⁴ En la siguiente url visitada el 6/7//2015 se extrajo la documentación necesaria para su implementación; <http://www.tecmint.com/install-ntp-server-in-centos/>

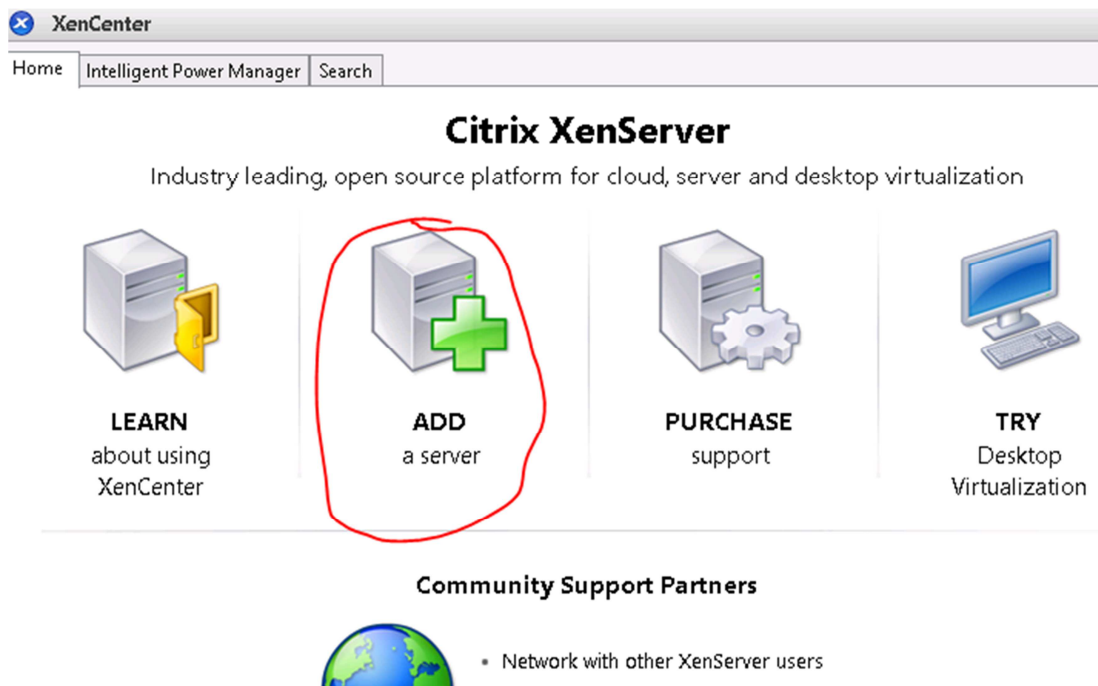


Figura 9 Agregando un servidor en XenCenter.

La primera vez que usamos XenCenter tendremos que crear el “pool” del que hablábamos en el apartado “Estructura Lógica”, así una vez que hemos añadido el primer servidor (normalmente el que hemos elegido para realizar funciones de “Máster”) tendremos que crear el “pool” propiamente dicho, para ellos pulsamos botón derecho sobre la etiqueta “XenCenter” y luego sobre la opción “New Pool”:

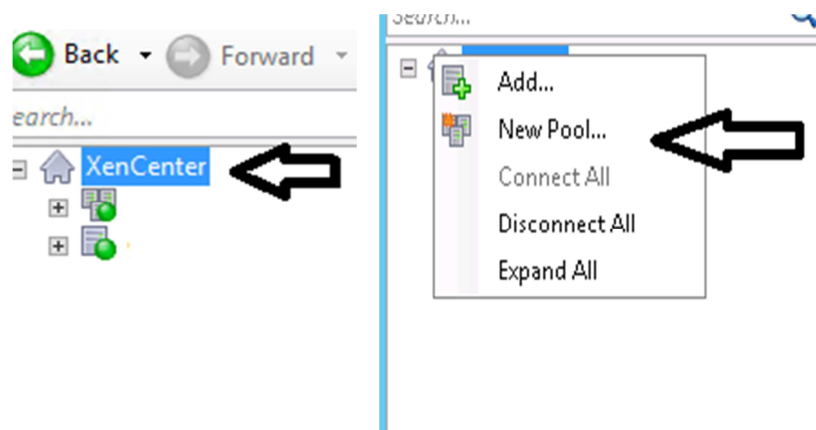


Figura 10 Crear un nuevo “Pool” en XenCenter.

Mostrándonos el siguiente cuadro de diálogo:

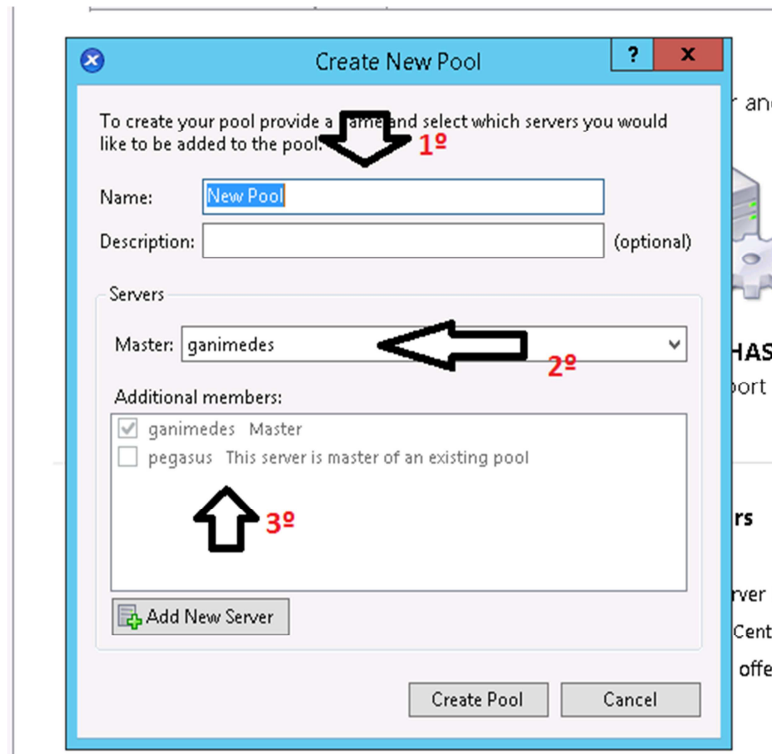


Figura 11 Cuadro de diálogo de XenCenter en la creación del "Pool"

Introduciremos (1º) el nombre con el que queremos identificar nuestro "pool", a continuación (2º) elegimos el servidor que será "Master" del "pool" y por último (3º) podemos añadir desde aquí mismo más servidores como miembros del "pool", siempre y cuando previamente los hayamos añadido en XenCenter como hacíamos en párrafos atrás con la opción "Add a Server" (si solo habíamos añadido uno, desde aquí solo veremos ese servidor).

Es importante tener en cuenta lo que dice la documentación oficial de XenServer respecto a los requerimientos necesarios de compatibilidad hardware entre los distintos servidores que queramos reunir en un mismo "pool", donde por ejemplo se especifica la necesidad de que sus CPU's sean en general compatibles, no pudiendo mezclarse en el mismo "pool" aquellas basadas en AMD con Intel. En el siguiente enlace se recoge amplia información sobre este asunto:

<http://support.citrix.com/proddocs/topic/XenCenter-61/xs-xc-pools-requirements.html>

A continuación añadiremos los recursos de almacenamiento al pool que acabamos de crear, recordemos que previamente habremos creado los volúmenes lógicos necesarios en la cabina de almacenamiento (ver apartado:)

Seleccionamos el nombre del "pool" recién creado y pichamos en la pestaña "Storage".

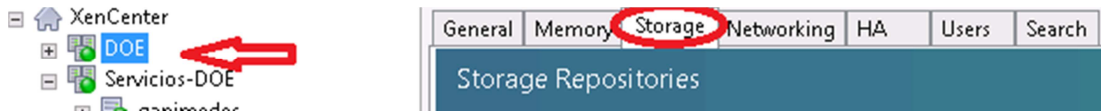


Figura 12 Tras crear el "Pool" creamos el almacenamiento de datos.

Donde podremos añadir el "Storage Resource" (SR) con la opción "New SR...".

Este es el aspecto del asistente que nos mostrará:

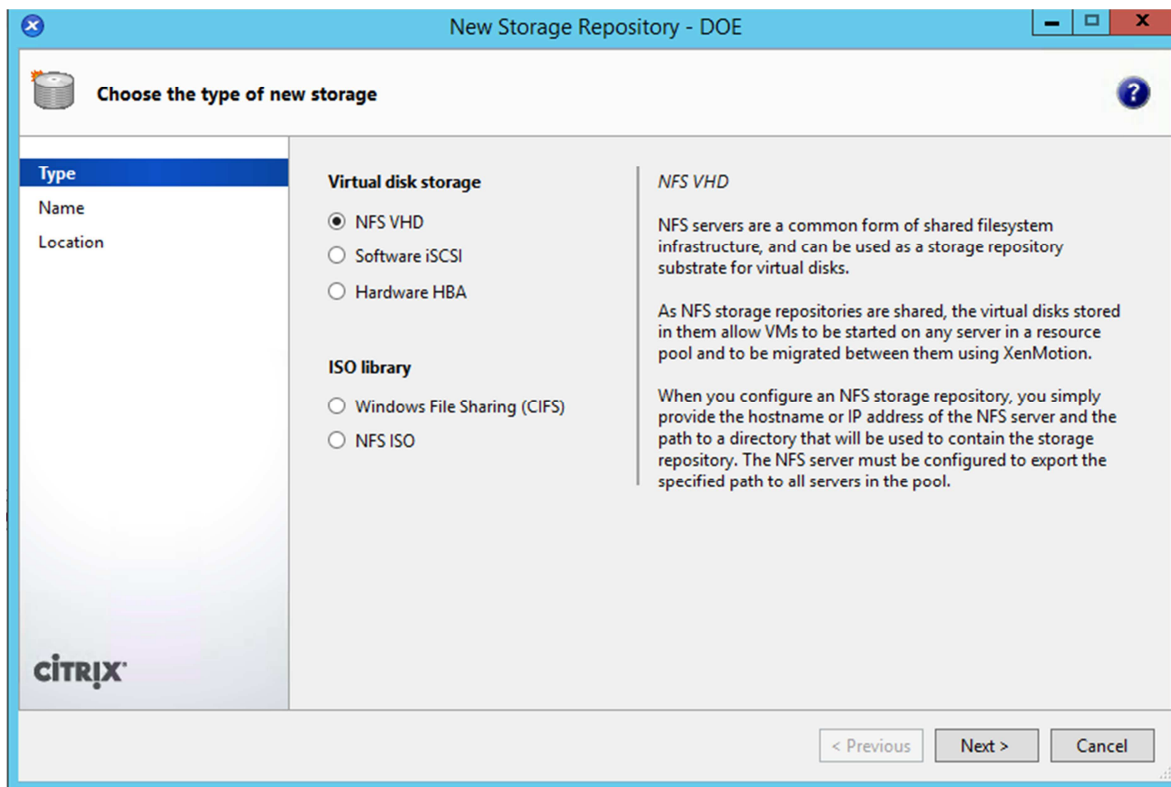


Figura 13 Asistente de creación de nuevo almacenamiento de datos. Pantalla de inicio.

No cambiamos ningún valor de los mostrados y pulsamos siguiente..."Next".

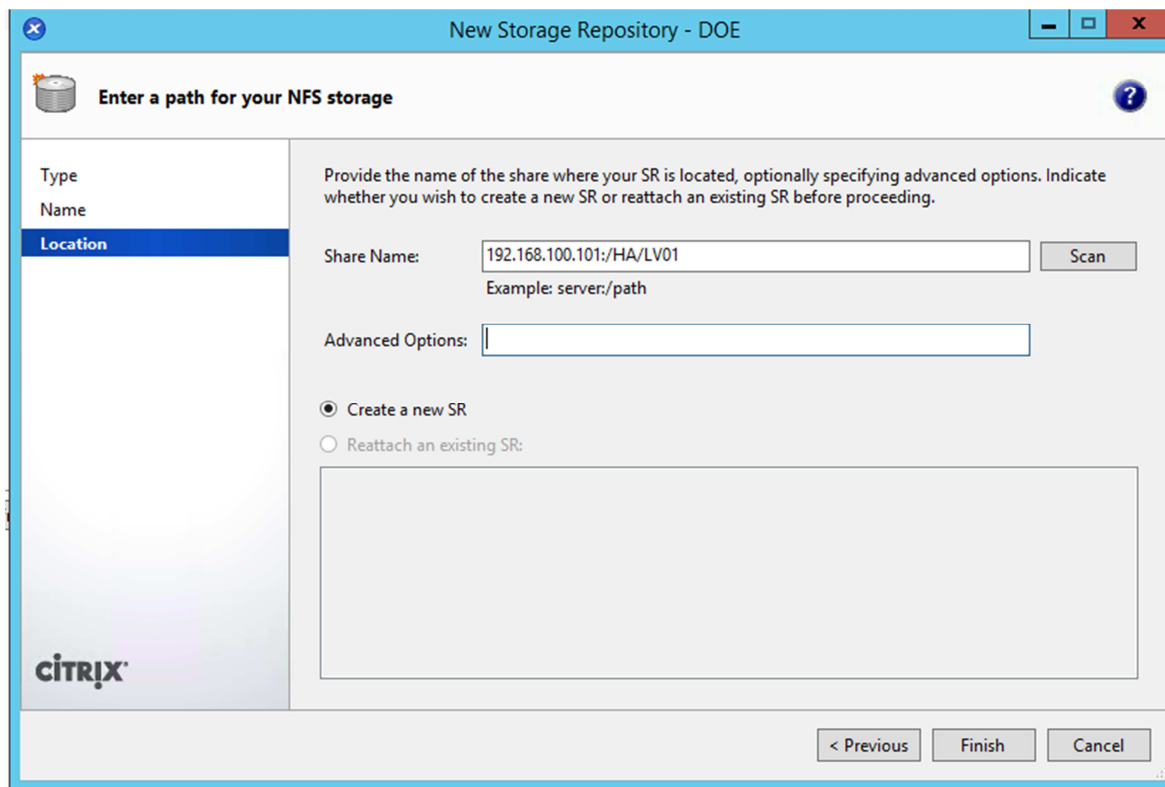


Figura 14 Asistente de creación SR. Introducción datos configuración.

Una vez hemos puesto un nombre para identificar el nuevo SR, le damos a "Next" y nos preguntará sobre los datos de conexión a la cabina de almacenamiento para este nuevo SR, escribimos el Share Name: 192.168.100.101:/HA/LV01 (viene de la dirección IP de la cabina de almacenamiento y del nombre de carpeta que hemos exportado mediante NFS en pasos anteriores. En Advanced Options no cambiamos nada y pulsamos "Finish", para terminar. Después de un breve tiempo de proceso, ya estará montado el SR. Seguiremos el mismo proceso con todos los SR que queramos utilizar, por ejemplo uno para el almacén de máquinas virtuales y otro para almacenar los ISO's o instaladores de sistemas operativos que ya hemos mencionado.

2.4 Creación y migración de máquinas virtuales.

2.4.1 Creación de una máquina virtual desde XenCenter.

El proceso para crear una nueva máquina virtual desde XenCenter es una tarea que consta de los siguientes pasos, siguiendo un intuitivo asistente que nos proporciona XenCenter:

- Teniendo seleccionado el nombre del "pool", en la barra de menú superior: "VM - New VM".

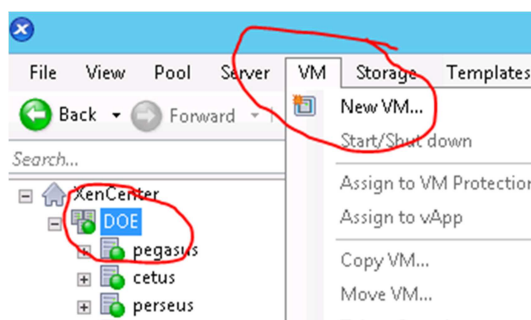


Figura 15 Creación VM desde XenCenter.

- En el asistente que nos muestra, seleccionaremos la plantilla correspondiente al sistema operativo con el que queremos crear la máquina virtual.
- Le damos a la máquina virtual un nombre representativo y una descripción.
- En la siguiente ventana del asistente nos pedirá que seleccionemos el método que vamos a utilizar para llevar a cabo la instalación del S.O. (bien con un DVD físico o con una imagen ISO). Normalmente usaremos un ISO que previamente habremos copiado a nuestro repositorio, en la misma lista podemos ver las imágenes que tenemos en el recurso de ISO's.
- En el siguiente paso podremos o bien asignar un servidor fijo en el que ejecutar esta máquina virtual o bien dejar que sea el pool de XenServer quien le asigne un servidor de forma dinámica, según un algoritmo de balanceo de carga interno de XenServer.
- A continuación le asignamos a nuestra máquina virtual los recursos de CPU y de memoria RAM que le vamos a asignar o reservar.

- Crearemos un nuevo disco duro virtual y se lo asignamos. En este punto, según la plantilla que hayamos elegido de sistema operativo, XenCenter nos sugiere ya un tamaño para el disco, aunque podemos modificarlo pulsando el botón “properties”.
- Asignamos la interfaz de red correspondiente a la red en la que queremos hacer visible esta máquina.
- Finalizamos este asistente de creación marcando la opción de encender la máquina virtual, de esta forma se iniciará con el proceso de instalación desde la ISO o el DVD seleccionado. Pinchando en la pestaña consola de la máquina recién creada podremos continuar con la instalación habitual del sistema operativo que hayamos seleccionado.

2.4.2 Generación de máquinas virtuales y "clones" para aulas.

Una vez hemos creado una máquina virtual y consideramos que está correctamente configurada, el siguiente paso será su despliegue para su uso en las aulas. A nivel de máquina virtual el procedimiento es similar al ya empleado en la virtualización en modo "local", es decir preparamos una máquina virtual "base" y de ella generamos "clones" que serán las VM's con las que realmente trabajan nuestros usuarios. La creación de estos "clones" se lleva a cabo de forma automatizada mediante el uso del siguiente script: */home/xen-scripts/CreaLAB-v3.sh*:

<pre>#!/bin/bash echo "Necesito la clave del servidor: " read -s clave echo "Introduce laboratorio a Generar(solo el numero[1..7]) Lab:" read num_lab rm -f genesis.csv && wget http://actualizador.dominio.compañia.es/scripts/genesis.csv</pre>	<pre># Lo ejecutamos por cada laboratorio.</pre>
--	--

<pre> sed 's/,158/,10/' genesis.csv > fichero.csv echo "" > lab\$num_lab.csv XE="/home/xen-scripts/XenCenter/xe.exe -s 192.168.100.1 -u root -pw \$clave" BASE="Labdoe_XP_Valencia_BASE" NETUUID="ae35d10e-a70d-cdd0-0426-41bbf96d7426" for line in \$(cat ./fichero.csv) do HOST=\$(echo \$line cut -d ',' -f 1 cut -d '.' -f1) MAC=\$(echo \$line cut -d ',' -f3) vmuuid=\$((\${XE} vm-clone vm="\$BASE" new-name- label=\$HOST) \${XE} vm-param-set uuid=\${vmuuid} ha-always-run=true uvdi=\$((\${XE} vm-disk-list name-label=\$HOST --minimal cut -d ',' -f 1 cut -d '.' -f1) \${XE} vdi-param-set on-boot=reset uuid=\$uvdi echo "\$HOST,id_maquina=\$vmuuid,id_disco=\$uvdi " >> lab\$num_lab.csv echo "" >> lab\$num_lab.csv </pre>	<pre> #Extrae el nombre de la linea del CSV #Extrae la MAC de la linea del CSV #Crea el clon, de tipo linked de la máquina base #Incluye el nuevo clon dentro de la alta disponib. de la infraestructura #Extrae el nombre del disco correspondiente a ese clon. #Marca ese disco en modo reset, para que no se guarden los cambios en cada reinicio. #Extrae el UUID nuevo de la tarjeta de red que tiene el clon creado #Destruye la tarjeta de red que tiene el clon creado #Le asigna la MAC de la línea del CSV y además mete al clon en la VLAN interna </pre>
---	---

<pre> vifuuid=\$(\${XE} vm-vif-list vm=\$vmuuid --minimal) \${XE} vif-destroy uuid=\$vifuuid \${XE} vif-create mac=\$MAC network-uuid=\$NETUUID vm- uuid=\$vmuuid device=0 echo "Creado clon \$HOST" # \${XE} vm-start vm=\$HOST done </pre>	<pre> #Opcionalmente podemos arrancar las máquinas después de creadas. </pre>
--	---

Tabla 9 Ejemplo de `/home/xen-scripts/CreaLAB-v3.sh` para crear clones para laboratorios.

Como se puede ver en el inicio de este script, nos traemos de un servidor (actualizador.dominio.compañia.es) el fichero **genesis.csv**. En este servidor tenemos alojada una base de datos en la que hemos dado de alta previamente todos los ordenadores físicos que tenemos en los laboratorios. “Actualizador” actúa como máquina de apoyo a la gestión de toda la infraestructura, en la aplicación web que se ha desarrollado en él podremos hacer selección de un grupo de equipos físicos o de un laboratorio completo para “extraer” de su base de datos un fichero formateado con la estructura de datos que vamos a necesitar para pasarlo como argumento a **genesis.csv**. El resto de líneas del script están explicadas y comentadas en su propia tabla.

Otra tarea que lleva a cabo este script, es transformar el primer octeto de la dirección IP de cada máquina física (recordemos que el script está trayendo datos de máquinas físicas o reales), por una dirección del rango de direcciones que se corresponde con la VLAN interna donde tenemos alojadas las máquinas virtuales y lo vuelca todo en el fichero **fichero.csv** que es el que realmente se utiliza en el script para la generación de las máquinas virtuales usando todos los datos de las reales (MAC, nombre, etc.) pero cambiando las IP al rango de la VLAN interna. Para entender mejor este proceso de “transformación”, en la siguiente tabla vemos comparado el contenido ejemplo de **genesis.csv** tal y como viene de la BBDD de equipos y el resultado en fichero.csv del que se alimenta realmente el script:

Contenido en genesis.csv	Contenido en fichero.csv
Nom_equipo.dominio, IP_maq_real, MAC_maq_real,	Nom_equipo.dominio, IP_maq_virtual, MAC_maq_real,
p.e.: lab6doe30, 158 .42.xxx.yyy, 00:1a:92:c9:05:bc,	p.e.: lab6doe30, 10 .42.xxx.yyy, 00:1a:92:c9:05:bc,

Tabla 10 Ficheros "csv" de ayuda en la creación de nuevos clones de VM's de laboratorios.

Una vez hemos llegado a este punto, tenemos los clones o máquinas virtuales creadas en nuestra infraestructura de virtualización, cada una de ellas lista y a la espera de que su máquina asignada en el "mundo real" se le conecte y la ponga en marcha para su uso mediante un cliente de RDP ligero equivalente a "Conexión a escritorio remoto" de Windows que tendremos instalado en cada equipo físico de los laboratorios.

2.4.3 Modificar una máquina virtual para "congelar" su disco duro.

En los laboratorios veníamos usando un software de terceros para proteger los equipos llamado Deepfreeze. Este software permite un modo de trabajo en el que el ordenador de laboratorio se queda en un estado que llaman "congelado" y en el que cada vez que el usuario reinicia el equipo, éste vuelve al estado inicial en el que se congeló, evitando así virus, cambios indeseados en la configuración e instalación de software no autorizado por los operadores. Esta herramienta ha resultado hasta ahora imprescindible para este entorno en donde los usuarios no tienen un ordenador asignado de forma permanente y cada equipo es compartido por distintos grupos de alumnos y prácticas.

Para que con el sistema de virtualización implementado, los equipos de los laboratorios se comporten de modo similar al anterior con Deepfreeze, se tiene que modificar la imagen base (antes de clonar los clientes) para que el disco duro de la máquina virtual no guarde los cambios cuando el usuario termine de utilizarla.

Para conseguir este efecto de equipo "congelado", tenemos que forzar de alguna manera, que cuando el usuario termine su trabajo se apague la máquina virtual en la que estaba conectado, sin su intervención y que después se apague también la máquina real en la que ha trabajado. Para que la máquina virtual siempre se quede "limpia" y preparada para ser usada por un nuevo usuario es imprescindible que haya un reinicio entre cada uso, o que sea apagada tras su utilización. XenServer provee un mecanismo para que los "clones" siempre arranquen "congelados" y no guarden los cambios que haga cada usuario. Este mecanismo consiste en indicar en el proceso de creación del clon que su unidad de almacenamiento se reinicia cada vez que arranca, mediante el uso del parámetro "on-boot"

A continuación un ejemplo, primero extraemos la uuid del disco VDI de una máquina virtual con el comando de XE-CLI `vm-disk-list`:

```
$$ {XE} vm-disk-list name-label=Labdoe_XP_Valencia
```

```
Disk 0 VBD:
```

```
uuid ( RO)      : ab100af8-8e23-f18c-229a-41bd429f8220
```

```
vm-name-label ( RO): Labdoe_XP_Valencia
```

```
userdevice ( RW): 0
```

```
Disk 0 VDI:
```

```
uuid ( RO)      : a2f43fd5-944d-41fa-a474-89f84972335f
```

```
name-label ( RW): Labdoe_XP_Valencia 0
```

```
sr-name-label ( RO): VMS-LV01
```

```
virtual-size ( RO): 32212254720
```

Conociendo el UUID de su disco VDI, podemos ver con qué parámetros y opciones está configurado con `vdi-param-list`:

```
$$ {XE} vdi-param-list uuid=a2f43fd5-944d-41fa-a474-89f84972335f
```

```
uuid ( RO): a2f43fd5-944d-41fa-a474-89f84972335f
```

```
name-label ( RW): Labdoe_XP_Valencia 0
```

```
name-description ( RW): Created by template provisioner
```

```
is-a-snapshot ( RO): false
```

```
snapshot-of ( RO): <not in database>
```

```
snapshots ( RO):
```

```
snapshot-time ( RO): 19700101T00:00:00Z
```

allowed-operations (SRO): forget; generate_config; update; resize; destroy; clone; copy;
snapshot

current-operations (SRO):

sr-uuid (RO): dd44b6c5-b2af-1903-d389-7b1eeca17e32

sr-name-label (RO): VMS-LV01

vbd-uuids (SRO): ab100af8-8e23-f18c-229a-41bd429f8220

crashdump-uuids (SRO):

virtual-size (RO): 32212254720

physical-utilisation (RO): 66048

location (RO): a2f43fd5-944d-41fa-a474-89f84972335f

type (RO): System

sharable (RO): false

read-only (RO): false

storage-lock (RO): false

managed (RO): true

parent (RO): <not in database>

missing (RO): false

other-config (MRW): content_id: 7c093dbe-6b24-9332-985e-0e6011ce9052

xenstore-data (MRO):

sm-config (MRO):

on-boot (RW): persist

allow-caching (RW): false

metadata-latest (RO): false

metadata-of-pool (RO): <not in database>

tags (SRW):

En negrita y subrayado podemos ver con qué opción tenemos activado el parámetro on-boot del disco VDI de esta máquina. Si queremos cambiarlo para que cada vez que se reinicie se pierdan los datos, en lugar de mantenerlos, tendremos que cambiar el valor de ese parámetro a on-boot=reset, mediante:

```
$ ${XE} vdi-param-set uuid=a2f43fd5-944d-41fa-a474-89f84972335f on-boot=reset
```

Ahora el valor del parámetro on-boot aparecerá en modo "reset":

```
$$$ ${XE} vdi-param-list uuid=a2f43fd5-944d-41fa-a474-89f84972335f
```

```
uuid ( RO): a2f43fd5-944d-41fa-a474-89f84972335f
```

```
name-label ( RW): Labdoe_XP_Valencia 0
```

```
name-description ( RW): Created by template provisioner
```

```
is-a-snapshot ( RO): false
```

```
snapshot-of ( RO): <not in database>
```

```
snapshots ( RO):
```

```
snapshot-time ( RO): 19700101T00:00:00Z
```

```
allowed-operations (SRO): forget; generate_config; update; resize; destroy; clone; copy;  
snapshot
```

```
current-operations (SRO):
```

```
sr-uuid ( RO): dd44b6c5-b2af-1903-d389-7b1eeca17e32
```

```
sr-name-label ( RO): VMS-LV01
```

```
vbd-uuids (SRO): ab100af8-8e23-f18c-229a-41bd429f8220
```

```
crashdump-uuids (SRO):
```

```
virtual-size ( RO): 32212254720
```

```
physical-utilisation ( RO): 66048
```

```
location ( RO): a2f43fd5-944d-41fa-a474-89f84972335f
```

```
type ( RO): System
```

sharable (RO): false
read-only (RO): false
storage-lock (RO): false
managed (RO): true
parent (RO): <not in database>
missing (RO): false
other-config (MRW): content_id: 7c093dbe-6b24-9332-985e-0e6011ce9052
xenstore-data (MRO):
sm-config (MRO): vhd-parent: d893f463-d902-4686-b106-36c3ae9a08c4
on-boot (RW): reset
allow-caching (RW): false
metadata-latest (RO): false
metadata-of-pool (RO): <not in database>
tags (SRW):

2.4.4 Creación de máquinas virtuales y "clones" para otros propósitos.

Para crear una máquina virtual nueva, que vamos a emparejar con un cliente de RDP ligero físico tendremos que dar los siguientes pasos:

1.- Creamos la máquina virtual empezando desde su instalación como veíamos en el punto 4.1 o bien cogiendo una VM base ya existente.

2.- Hacemos los ajustes que necesite esa máquina concreta (añadir - quitar software p.e.).

3.- Incluimos en el fichero /etc/hosts del servidor "[bróker](#)" una línea adicional con el formato IP de la nueva máquina y su nombre en la red interna: .cloud (p.e. 10.42.xx.yy nombre_equipo.compañia.cloud), de esta forma la tendrá en cuenta el servicio DNS en la red privada que gestiona "bróker". Reiniciaremos el servicio DNS en "broker.dominio.compañia.es" con el comando ***service dnsmasq restart***.

4.- Incluir la nueva máquina en el fichero /etc/dhcp/VMS-BASE.conf del servidor "[bróker](#)" , añadiendo una línea más con el formato:


```
host NOMBREHOST {hardware ethernet MAC-HOST; fixed-address IPASIGNADA; option host-name "NOMBREHOST.COMPLETO.cloud";}
```

Reiniciar el servicio DNS en "broker.dominio.compañia.es" con el comando *service dhcpd restart*.

5.- A continuación crearemos su máquina virtual clon, para ello usamos un script muy parecido al de creación de máquinas de los laboratorios al que hemos llamado **crea-una.sh**.

Hay que tener en cuenta, antes de ejecutarlo, que este script se alimentará del fichero **un.csv**, y por tanto tendremos que modificarlo a mano incluyendo en él la máquina que estamos creando para realizar la vinculación máquina "real" con máquina "virtual". En **un.csv** pondremos el nombre de equipo, IP y MAC de la máquina real, el script **crea-una.sh** será quien realice la conversión a la dirección IP de la máquina virtual 10.xxx.xxx.xxx. También tendremos que modificar en este script el nombre de la máquina base que vamos a utilizar ya que el usuario va a trabajar siempre con un "linked clone" de ella (esto nos sirve para que volver al momento en el que la máquina está recién creada es tan sencillo como volver a lanzar este script y crear el clon de nuevo).

Por último tendremos que decidir, según cada caso y las necesidades del usuario de esta máquina virtual, si su disco duro se creará en modo **persist** o **reset**. Si el script lo dejamos con el primer modo, la máquina que utilice el usuario irá guardando los cambios que éste vaya realizando, mientras que en el modo reset, la máquina virtual arranca siempre "limpia" de datos y programas y no guardará los cambios.

2.4.5 Migración desde VirtualBox con XP.

Por último también existe la opción de incorporar a nuestra infraestructura una máquina virtual que previamente haya sido creada con otro hipervisor. En nuestro caso, ligando con la parte del proyecto de virtualización local, tenemos máquinas virtuales que tenemos creadas con VirtualBox y que podrían ser aprovechadas por XenServer para la virtualización remota. Su importación al entorno de XenServer pasa por los siguientes pasos:

1º. Desde VirtualBox.

- Desinstalamos las "Guest Additions" que normalmente tendrá instalada la máquina virtual en VirtualBox y que actúan como "drivers" específicos de VirtualBox que tendremos que sustituir por los de XenServer.
- Descargar y ejecutar la siguiente utilidad en la máquina origen: https://www.virtualbox.org/attachment/wiki/Migrate_Windows/MergeIDE.zip. Con ella realizamos los cambios necesarios para que ésta utilice unos drivers genéricos de acceso a disco duro.

- Si el formato del disco duro virtual no es VHD (nativo en Xen), tendremos que convertirlo utilizando la herramienta que incluye Oracle VirtualBox llamada “Vboxmanage” con los siguientes parámetros:

VBoxManage clonehd "vm_a_migrar.vdi" "vm_a_migrar.vhd" --format VHD

- Necesitamos renombrar el VHD que acabamos de crear con el formato de UUID del disco para que de esta forma XenServer sea capaz de reconocerlo en su infraestructura, encontramos este UUID de nuevo con la utilidad de “Vboxmanage”:

VBoxManage showhdinfo "nom_vm_a_migrar.vhd"

2º. Desde XenCenter.

- Copiamos el fichero VHD en el recurso NFS que tenemos montado en XenCenter y que nos sirve como almacén para nuestras máquinas virtuales.
- En este almacén iremos a la pestaña “storage” y pulsaremos el botón “rescan” para que actualice la lista de discos virtuales que contiene, una vez hecho esto, nos debería aparecer un nuevo disco sin nombre asignado.

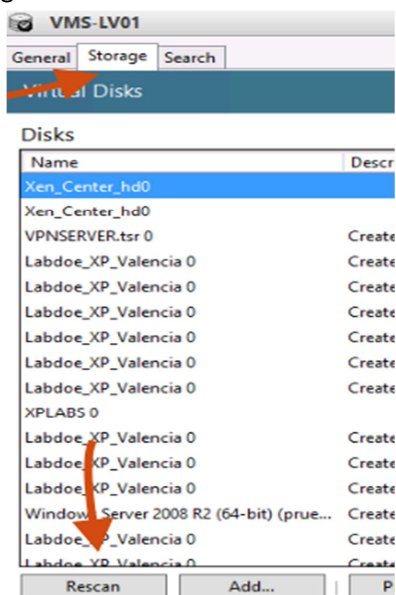


Figura 16 Actualizar lista discos virtuales contenidos en un SR (XenCenter)

- A partir de este punto podremos crear una nueva máquina virtual según el procedimiento descrito anteriormente, pero teniendo en cuenta que en la pestaña de almacenamiento tendremos que eliminar el disco para sustituirlo o asociar el que acabamos de copiar y detectar.

- Por último, iniciar la máquina virtual e Instalar las utilidades o controladores Xen-Tools.

2.5 Bróker estático: V1.0.

En este apartado se recoge el modo en que vamos a vincular los equipos físicos de los laboratorios con las máquinas virtuales alojadas la infraestructura creada con XenServer.

En su versión 1.0 el servidor virtual que hemos denominado "broker.dominio.compañía.es" en nuestra infraestructura realizaba principalmente tres funciones:

1º. Crear un túnel o vincular cada ordenador "físico" de laboratorio con su equivalente virtual, el cual permanece encendido en espera de que un usuario desde su ordenador "hermano" físico se le conecte mediante RDP (Remote Desktop Protocol). Cada una de estas máquinas virtuales, alojadas en el pool de XenServer, tienen el mismo nombre que su "espejo" en el mundo físico, misma subred IP (pero comenzando con una dirección 10.xxx.xxx.xxx) y la misma MAC Address.

2º. Realizar funciones de servidor DHCP y DNS dando servicio a la red virtual privada que hemos llamado [VLAN-Interna](#) (de rango 10.xxx.xxx.xxx).

3º. Actuar como enrutador (router) para que las máquinas virtuales que están en la [VLAN-Interna](#), tengan conexión de red hacia el exterior.

Veamos cada una de estas funciones con mayor detalle:

2.5.1 Vinculación cliente físico - máquina virtual.

Como hemos visto, necesitamos implementar un mecanismo que realice la vinculación entre cada ordenador "físico" de laboratorio con su equivalente virtual. El usuario cuando enciende un ordenador de laboratorio ejecutará un programa de tipo "Conexión de Escritorio Remoto" de Windows, en nuestro caso usaremos "Rdesktop", un cliente ligero que se ejecuta sobre el sistema operativo Linux.

Antes de iniciar "Rdesktop", los equipos cliente intentan abrir una conexión con el equipo virtual que está alojado en una red privada y por tanto, en principio, no van a poder acceder hasta él, por ejemplo accedemos desde lab5doe30.dominio.compañía.es a su máquina virtual lab5doe30.compañía.cloud, cuyas IP's serán algo parecido a 158.42.XX.YY y 10.42.XX.YY, respectivamente, necesitamos algún mecanismo para unir esas dos redes de distinto rango.

Para ello tendremos que realizar ajustes tanto en el servidor "broker.dominio.compañía.es", que actúa como servidor de estos "túneles" como en los equipos físicos o clientes desde los que se conectarán los usuarios en los laboratorios a las distintas máquinas virtuales.

2.5.1.1 Ajustes necesarios en el Servidor "broker.dominio.compañia.es".

En "broker.dominio.compañia.es", o el equipo que actúe como servidor de estos "túneles" que en realidad no son más que conexiones seguras utilizando el protocolo "ssh" para redirigir un puerto local a uno remoto, realizamos los siguientes ajustes:

1º Hemos creado un usuario especial para acceder a este servidor por protocolo seguro "ssh", al que hemos llamado "tunel".

2º Iniciamos sesión en "broker.dominio.compañia.es" con este usuario "tunel".

3º Con este usuario, hemos generado el par clave pública/privada:

```
ssh_keygen -t rsa -f tunel.key
```

Señalar que sería interesante, como "best-practice" de seguridad: crear una clave privada para cada puesto de trabajo, así de esa manera no se comprometería nunca la seguridad de todos sino de solamente la de un equipo.

4º Copiamos la parte pública de la clave al fichero /home/tunel/.ssh/authorized_keys, que es el fichero que contiene las claves con autorización para acceder por "ssh" a "broker.dominio.compañia.es", para ello ejecutamos:

```
cat tunel.key.pub >> /home/tunel/.ssh/authorized_keys
```

5º Modificamos el /home/tunel/.ssh/authorized_keys y añadimos al principio del mismo la modificación que resaltamos en negrita y subrayado:

```
no-X11-forwarding,no-agent-forwarding,command="sleep 30" ssh-rsa  
AAAAB3NzaC1yc2EAAAABIwAAA... (mostramos parcialmente la clave) ...ULv6nLOFvQ==  
tunel@broker.compañia.cloud
```

Con esta modificación protegemos "broker.dominio.compañia.es" limitando las acciones que el usuario tunel puede ejecutar, además y como único comando de entrada se establece el comando "sleep" para que si no hay actividad se cierre la conexión pasados 30 segundos.

2.5.1.2 Ajustes necesarios en el cliente físico.

En cada equipo físico hay que copiar en `/root/.ssh/` el fichero `tunel.key`. Es la parte privada de la clave generada en "broker.dominio.compañia.es" en el apartado anterior. Copiaremos la huella digital del servidor en el fichero `/root/.ssh/known_hosts` para evitar ataques típicos como "Man in the middle".

También necesitamos modificar en los equipos físicos el fichero `"/etc/rc.d/rc.local"` para que cuando los arranquemos se establezca el vínculo tal y como hemos explicado:

```
#extraer el nombre de la máquina

host=$(hostname -s)
#Redirige el puerto local 3389 (RDP) mediante ssh a la maquina virtual gemela

ssh -f -i /root/.ssh/tunel.key -L 3389:$host.compañia.cloud:3389 tunel@broker.compañia.dominio
" "

#Lanza el cliente de escritorio remoto montando también el recurso USB

xinit /usr/bin/rdesktop -f -r disk:usb=/media/usb localhost
```

Tabla 11 Extracto de "rc.local" equipos laboratorios con detalle del establecimiento del vínculo con su VM.

Se ha extraído el nombre corto de la máquina cliente para pasarlo como argumento de la conexión ssh, ya que como hemos explicado anteriormente, los nombres de los equipos físicos y virtuales son iguales. Posteriormente hacemos el port-forwarding a ese equipo remoto utilizando como "puente" "broker.dominio.compañia.es".

2.5.2 Funciones de servidor DHCP y DNS.

2.5.2.1 Servicio DHCP.

Veamos con detalle estos servicios, en primer lugar, la configuración del servicio DHCP que la encontraremos en:

`/etc/dhcp/dhcpd.conf`

```
authoritative;
```

```

ddns-update-style interim;

ignore client-updates;

subnet 10.0.0.0 netmask 255.0.0.0 {
    option routers 10.0.0.1;
    option subnet-mask 255.0.0.0;
    option domain-name-servers 10.0.0.1;
    default-lease-time 28800;
    max-lease-time 43200;
    range 10.0.200.1 10.0.202.240;
}

## IPs fijas de SERVIDORES

include "/etc/dhcp/VMS-BASE.conf";

include "/etc/dhcp/VMS-DHCP.conf";

```

Tabla 12 Archivo dhcpd.conf en "bróker". Configuración servicio DHCP.

Como vemos incluye a los ficheros `/etc/dhcp/VMS-BASE.conf` y `/etc/dhcp/VMS-DHCP.conf` en los cuales están puestas las líneas correspondientes para asignar direcciones IP's de forma estática a cada máquina virtual en este formato:

```

host NOMBREHOST {hardware ethernet MAC-HOST; fixed-address IPASIGNADA; option host-
name "NOMBREHOST.COMPLETO.cloud";}

```

En el fichero `/etc/dhcp/VMS-DHCP.conf` se recogen las máquinas virtuales de los laboratorios y el fichero `/etc/dhcp/VMS-BASE.conf` recoge las máquinas virtuales "especiales" como servidores o de otros propósitos. Por tanto si queremos crear y añadir alguna máquina virtual distinta de las de los laboratorios, la tendremos que añadir en este fichero de forma manual y reiniciar el servicio DHCP con el siguiente comando:

`service dhcpd restart`

Las máquinas virtuales de los laboratorios ya están incluidas en el fichero */etc/dhcp/VMS-DHCP.conf* de forma automática y cada vez que usamos el script */root/DOE/DHCP+DNS.sh*, que se encarga de extraer los datos de nuestra BBDD de equipos de aula pasándolos a un fichero tipo csv que contiene tres campos separados por coma: "nombre.dominio", "IP" y "MAC":

```
DOMINIO=compañia.cloud

mv VMS-DHCP.conf VMS-DHCP.conf.old

for line in $(cat ./fichero.csv)
do
HOST=$(echo $line | cut -d',' -f1 | cut -d'.' -f1)
IP=$(echo $line | cut -d',' -f2)
MAC=$(echo $line | cut -d',' -f3)

echo "host ${HOST} {hardware ethernet ${MAC}; fixed-address ${IP}; option host-name
\"${HOST}.${DOMINIO}\";}" >> VMS-DHCP.conf

done
```

Tabla 13 Extracto del script "DHCP+DNS.sh"

Cuando hagamos cambios de equipos en nuestros laboratorios, bien porque retiremos, cambiemos o añadamos equipos nuevos, necesitaremos actualizar la lista de equipos que "reconoce" nuestro servidor de DHCP y les asigna dirección IP, para ello usaremos el script que acabamos de ver mediante:

```
sh /root/DOE/DHCP+DNS.sh
```

Dentro del script ya hemos implementado el reinicio del servicio como paso final:

```
service dhcpd restart
```

2.5.2.2 Servicio DNS.

De forma similar al anterior servicio, la configuración del servicio DNS la encontraremos en el fichero:

/etc/dnsmasq.conf

```
interface=eth1
server=158.42.250.65
addn-hosts=/etc/dnsmasq.d/VMS-DNS.conf
all-servers
domain-needed
bogus-priv
domain=compañia.cloud
expand-hosts
local=/compañia.cloud/
```

Tabla 14 Archivo dnsmasq.conf en “bróker”. Configuración servicio DNS.

Como podemos observar, incluye el fichero /etc/dnsmasq.d/VMS-DNS.conf, donde se encuentra el listado de máquinas virtuales para la resolución interna de la red, en 2 columnas con formato:

“IP-Maquina” “NombreCompleto.cloud”

```
DOMINIO=compañia.cloud
mv VMS-DNS.conf VMS-DNS.conf.old
for line in $(cat ./fichero.csv)
```



```

do

HOST=$(echo $line | cut -d',' -f1 | cut -d'.' -f1)

IP=$(echo $line | cut -d',' -f2)

MAC=$(echo $line | cut -d',' -f3)

    echo "${IP} ${HOST}.${DOMINIO}" >> VMS-DNS.conf

done

```

Tabla 15 Extracto del script DHCP+DNS.sh

Con el mismo script que usábamos para generar las máquinas que tiene que conocer el servidor de DHCP, generamos de forma automática el fichero `/etc/dnsmasq.d/VMS-DNS.conf` (que es el que contiene el listado de equipos de aula para su resolución de nombres).

Como ocurría en el apartado anterior, cuando hagamos cambios de equipos en nuestros laboratorios, hay que actualizar de nuevo la lista de equipos para el DNS, ejecutando como vimos:

```
sh /root/DOE/DHCP+DNS.sh
```

Al igual que nos sucedía en el servicio de DHCP, las máquinas virtuales especiales, no de laboratorios, se tendrán que incluir manualmente agregando una línea en el fichero `/etc/hosts` de "bróker" con el siguiente formato:

```
IP Nombre.subdominio.dominio
```

Recordemos que para este caso, para que sean efectivos los cambios tendremos que reiniciar el servicio DNS con el siguiente comando:

```
service dnsmasq restart
```

2.5.3 Funciones de enrutamiento.

Como se ha mencionado anteriormente, la tercera función de "bróker" consiste en actuar como enrutador para que las máquinas virtuales que están en la [VLAN-Interna](#), tengan conexión de red hacia internet. Para ello se lanza automáticamente un script en `broker.dominio.compañia.es`: que se llama `/usr/sbin/firewall.sh`.

Este script establece las típicas reglas básicas de un cortafuegos de RedHat (origen de esta distribución Linux), abriendo nada más los puertos que necesitamos que el equipo tenga abiertos (ssh y

DNS) y una regla que permite encapsular las IPS's de rango 10.xx.xx.xx para que las peticiones que hagamos fuera, una vez enrutadas sean efectivas:

```
iptables -A POSTROUTING -t nat -o eth0 -j MASQUERADE  
echo 1 > /proc/sys/net/ipv4/ip_forward
```

En la siguiente figura, se puede observar el recorrido completo del acceso desde una máquina física con cliente RDP a la máquina virtual asociada.

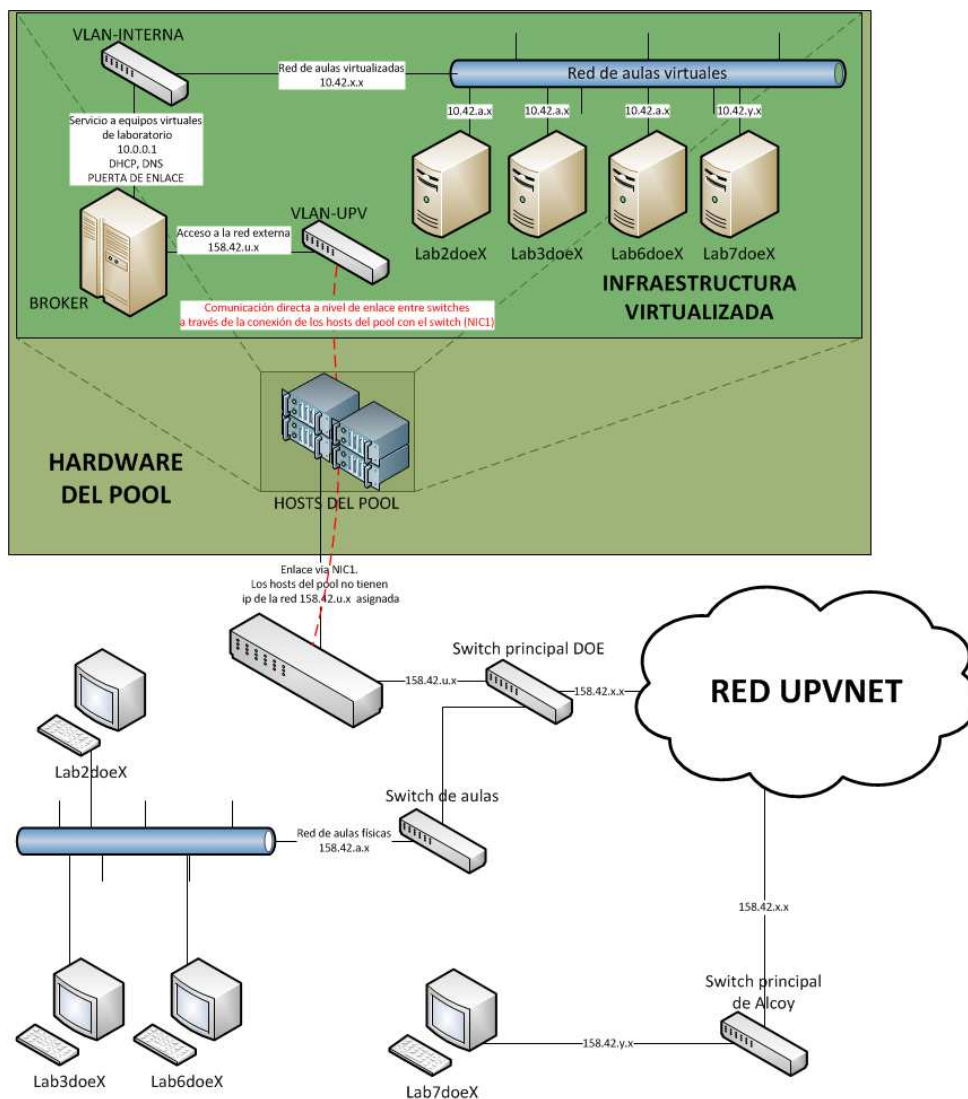


Figura 17 Gráfico ruta entre máquina física – máquina virtual.

2.6 Bróker v2.0: Versión Dinámica.

Hasta este punto del proyecto, para que un cliente de laboratorio pudiese conectarse a la máquina virtual (VM) que tenía asociada, requería que previamente dicha VM estuviese encendida en algún servidor, además, cuando un alumno terminaba la práctica y cerraba su sesión, un script de cierre de sesión apagaba totalmente esa VM, de esta forma nos aseguramos que había un “reinicio” entre cada alumno y devolvemos la máquina virtual a su estado “origen”. Para poder garantizar que dicha máquina estaba operativa de nuevo para la siguiente práctica, utilizamos la opción de levantarla automáticamente con la “Alta Disponibilidad” (HA), como se explica más adelante.

Este modo de trabajo necesitaba que todas las máquinas de todos los laboratorios estuviesen encendidas las 24 horas del día a la espera de ser usadas. Más crítico aún es el hecho que no pudiéramos levantar máquinas diferentes para diferentes tipos de prácticas (por ejemplo, una con Windows 8 y otra con Windows XP y software más antiguo de 16 bits, otra con windows 2008 server, etc.), se quedaba la vinculación limitada a un equipo real = un equipo virtual.

La mejora natural de este sistema pasaba por conseguir que el bróker sea capaz de: estando una VM apagada, la ponga en marcha y además se pueda elegir entre varios modelos de VM según un menú de selección mostrado al usuario. Gracias a este método podremos preparar previamente prácticas que requieran Sistemas Operativos o configuraciones diferentes.

Al no tener que tener encendidas todas las VM de antemano, ahorramos una importante cantidad de memoria RAM en los servidores. Con el nuevo modelo de bróker se aprovecha mejor el hardware, a la par que nos permite obtener un importante ahorro de energía.

2.6.1 Ajustes necesarios en el Servidor "broker.dominio.compañia.es".

Para conseguir este nuevo objetivo, necesitamos crear dos nuevas claves RSA en "broker.dominio.compañia.es". A una le vamos a llamar “**entrada.key**” y a la otra “**nmap.key**”. La primera será utilizada para encender la máquina virtual y la segunda la usaremos en un bucle cuya función será saber cuándo está lista dicha VM para que su equipo físico cliente se pueda conectar a ella. Así en esta versión dividimos el proceso de ajuste en dos partes:

Parte 1: Ajustes para “entrada.key”:

1. Iniciamos sesión con el usuario “tunel” (que ya teníamos creado desde la versión 1.0 de bróker).
2. Creamos el par clave pública/privada con el comando:

```
ssh_keygen -t rsa -f entrada.key
```

3. Copiamos la parte pública de la clave al fichero `/home/tunel/.ssh/authorized_keys`, fichero que contiene las claves con autorización para acceder por "ssh" a "broker.dominio.compañia.es", para ello ejecutamos:

```
cat entrada.key.pub >> /home/tunel/.ssh/authorized_keys
```

4. A continuación modificaremos el fichero `/home/tunel/.ssh/authorized_keys` para que "broker.dominio.compañia.es" sea capaz de ejecutar el script "**tunel.sh**" añadiendo lo que aparece en negrita:

```
no-X11-forwarding,no-agent-forwarding,command="/home/tunel/tunel.sh" ssh-rsa  
AAAAB...resto-de-la-clave-rsa...V/Q== tunel@broker.compañia.cloud
```

El script "**tunel.sh**" que se ejecutará en "broker.dominio.compañia.es", será el encargado de en primer lugar encender la VM su contenido se muestra a continuación:

```
#!/bin/bash  
  
# Primero introducimos en una variable la ruta y credenciales para usar XE-CLI  
XE="/usr/bin/xe -s 192.168.100.1 -u root -pw "La_Clave_que_sea""  
  
VM=$(echo $SSH_ORIGINAL_COMMAND | cut -d":" -f1)  
  
#Comprueba si la VM ya está encendida y si no lo está, la enciende  
if [ -z $(($XE vm-list name-label=$VM power-state=running --minimal) ) ]; then  
    #echo halted  
    $XE vm-start name-label=$VM  
fi  
  
sleep 120
```

Tabla 16 Script "túnel.sh"

Cuando un cliente físico se conecta, como veremos más adelante, envía a "broker.dominio.compañia.es" por ssh, su nombre que está almacenado en la variable `$host`. Para recuperar ese valor en "broker.dominio.compañia.es" necesitamos utilizar la variable reservada de

“ssh”: `SSH_ORIGINAL_COMMAND` y una vez recuperado, nos quedamos solo con el principio del nombre del equipo físico cliente haciendo “cut” hasta el primer punto (valor que introducimos en la variable “VM” que usaremos para arrancar la máquina virtual).

Parte 2: Ajustes para “nmap.key”

Para la segunda parte del proceso, la que se refiere a la clave “nmap” tenemos que crear también su correspondiente par clave pública/privada, así como su script asociado:

1. Iniciamos sesión con el usuario “tunnel” (que ya teníamos creado desde la versión 1.0 de bróker).
2. Creamos el par clave pública/privada con el comando:
`ssh_keygen -t rsa -f nmap.key`
3. Copiamos la parte pública de la clave al fichero `/home/tunnel/.ssh/authorized_keys`, fichero que contiene las claves con autorización para acceder por “ssh” a “broker.dominio.compañia.es”, para ello ejecutamos:
`cat nmap.key.pub >> /home/tunnel/.ssh/authorized_keys`
4. A continuación modificaremos el fichero `/home/tunnel/.ssh/authorized_keys` para que “broker.dominio.compañia.es” sea capaz de ejecutar el script “nmap.sh” añadiendo lo que aparece en negrita:

```
no-X11-forwarding,no-agent-forwarding,command="/home/tunnel/nmap.sh" ssh-rsa AA  
.....resto-de-la-clave-rsa.....J9Q== tunnel@broker.compañia.cloud
```

El script que ejecutamos ahora en “broker.dominio.compañia.es” es “nmap.sh” (Mantiene este nombre pues al principio se pensó utilizar la utilidad “nmap” para este cometido, pero finalmente se ha simplificado mediante el uso de telnet):

```
#!/bin/bash  
  
#  
  
# nmap -p 3389 -oG - 192.168.187.122  
  
#  
  
#/usr/bin/nmap -p $PORT $VM.compañia.cloud | grep $PORT
```

```
VM=$(echo $SSH_ORIGINAL_COMMAND | cut -d":" -f1)
PORT=$(echo $SSH_ORIGINAL_COMMAND | cut -d":" -f2)
echo exit | /usr/bin/telnet $VM.compañia.cloud $PORT | grep Connected
```

Tabla 17 Script "nmap.sh".

De esta manera controlamos si la máquina virtual a la que nos queremos conectar está preparada, cuando el comando telnet nos devuelve la cadena "Connected".

2.6.2 Ajustes necesarios en el cliente físico.

En los clientes hay que copiar en `/root/.ssh/` los ficheros **entrada.key** y **nmap.key**, que los habíamos creado en "broker.dominio.compañia.es". Es la parte privada de la clave generada en broker.compañia.dominio en el apartado anterior.

En el sistema operativo de los Clientes o equipos físicos de los laboratorios, para que funcione todo el proceso, tendremos que modificar de esta forma el fichero `/etc/rc.d/rc.local` (extracto):

```
...

host=$(hostname -s)
# Atrapo el Control-C
trap " 2
/usr/bin/ssh -M -S my-ctrl-socket -q -f -i /root/.ssh/entrada.key -L 3389:$host.compañia.cloud:3389
tunel@broker.compañia.dominio "$host" 2>/dev/null
res=$(/usr/bin/ssh -q -i /root/.ssh/nmap.key tunel@broker.compañia.dominio "$host:3389" 2>/dev/null)
contador=5
while [ -z "$res" ]
do
echo $contador | /usr/bin/dialog --title "Cliente Remoto DOE" --gauge "Conectando..." 10 70 0
plymouth message --text=" Cargando Maquina Virtual ... $contador %"
let contador=$contador+10
res=$(/usr/bin/ssh -q -i /root/.ssh/nmap.key tunel@broker.compañia.dominio "$host:3389"
```

```
2>/dev/null)

    sleep 2

    done

xinit /usr/bin/rdesktop -x 0x8E -f -z -u labdoe -r disk:usb=/media/usb localhost

/usr/bin/ssh -S my-ctrl-socket -O exit -i /root/.ssh/entrada.key tunel@broker.compañia.dominio

for i in $(seq 0 20 100) ; do sleep 1; echo $i | dialog --title "Cliente Remoto DOE" --gauge "Desconectando..."
10 70 0; done

/sbin/shutdown -h 0

...(resto de rc.local)
```

Tabla 18 Extracto del fichero rc.local: inclusión de entrada.key y nmap.key

En primer lugar hemos creado un método “por aproximación” que mediante sleeps y mensajes nos ayuda a mantener al usuario en espera mientras se pone en marcha su máquina virtual asignada mostrándole una barra y/o porcentaje de progreso.

Cuando la máquina ha devuelto el estado de “connected” al telnet por ssh, entonces es el cliente de conexión de escritorio remoto quien se encarga de iniciar ya la sesión de trabajo, mediante “Rdesktop”.

Cuando el usuario cierra la sesión, este script continua en el punto donde se quedó, mostrando al usuario el mensaje “Desconectando...” y además apagará el sistema operativo del cliente físico. La máquina virtual que tenía asociada también se apaga gracias a un programa que se ejecuta al cerrar sesión el usuario. Recordemos nuestro interés o necesidad de que cada nuevo usuario reciba su máquina virtual “limpia” y recién iniciada para que no se mantenga información de un alumno al siguiente que la utiliza.

2.7 Alta Disponibilidad (HA)

En la primera implementación de "broker.dominio.compañia.es" (v 1.0), con el fin de garantizar que las máquinas virtuales o mejor dicho que los clones a los que se conectan los usuarios desde los laboratorios estuviesen arrancados y esperando su uso, necesitábamos algún mecanismo que de forma automática los “vigilara” y volviera a encender en caso de detectar su apagado.

Pensemos que en ciertos casos, como cuando el usuario lanza un apagado remoto, o se produce un incidente eléctrico (caída prolongada de la corriente eléctrica), o cuando ocurre un reinicio inesperado o programado en alguno de los servidores, etc., etc., y queda alguna VM apagada, desde ese momento teníamos un equipo físico de laboratorio “inutilizado” para trabajar (aunque ese puesto de trabajo siempre tiene la opción de utilizar la virtualización “local” que mantiene en su disco como medida de seguridad).

Para dar solución a este problema XenServer nos ofrece la posibilidad de habilitar una funcionalidad de Alta Disponibilidad en el pool (denominada HA acrónimo de High Availability). Esta función permite definir cuáles de nuestras máquinas virtuales queremos que siempre estén en funcionamiento (salvo que nosotros decidamos apagarlas desde XenCenter). Este sistema se encargará de que siempre estén en marcha, incluso ante la eventual caída o fallo de un servidor. Si las máquinas que seleccionemos para estar en modo HA se apagan por la circunstancia que sea, el sistema las volverá a encender y si un servidor deja de funcionar, el sistema se encargará de encender las máquinas virtuales que tenía operando en los otros servidores (siempre y cuando encuentre recursos disponibles para hacerlo).

En nuestro caso, como hemos visto, todas las máquinas virtuales para usar en los laboratorios estuvieron marcadas como de Alta Disponibilidad hasta que se cambió “bróker” a la versión 2.0, momento en el cual ya no necesitábamos tener todas las máquinas virtuales siempre encendidas salvo algunas de ellas cuyo uso específico es crítico, como puede ser el mismo “bróker” que necesitamos que siempre permanezca encendido y en servicio.

Habilitar la alta disponibilidad es un proceso sencillo a continuación veremos los pasos para su habilitación.

2.7.1 Creación de un nuevo recurso de almacenamiento (SR) para HA.

El sistema de alta disponibilidad necesita disponer de forma exclusiva de un recurso de almacenamiento propio, por lo que como primer paso vamos a necesitar crear un nuevo recurso de almacenamiento en nuestra cabina de discos.

Abrimos una consola de comandos por ssh en la cabina de almacenamiento y realizamos los siguientes pasos:

1. Creamos un nuevo volumen lógico, con 1 Gb. de tamaño tendremos suficiente:

```
lvcreate -L 1GB -n LV_HA VG_ALMACEN
```

Podemos listar los volúmenes lógicos con el comando lvs:

```
lvs
```

```
LV VG Attr LSize Pool Origin Data% Move Log Cpy%Sync Convert
```



```
LV_HA VG_ALMACEN -wi-ao---- 1,00g
```

```
LV_VMS VG_ALMACEN -wi-ao---- 2,00t
```

2. A continuación damos formato al nuevo volumen lógico, hemos elegido formato tipo xfs:.

```
mkfs.xfs -i size=512 -f /dev/VG_ALMACEN/LV_HA
```

3. Creamos un nuevo directorio donde montar este nuevo almacén y lo incluimos en

```
mkdir -p /HA/LV01
```

4. Incluimos una nueva línea para este nuevo directorio como punto de montaje en el fichero

```
/etc/fstab usando vi:
```

```
vi /etc/fstab
```

...

```
/dev/VG_ALMACEN/LV_HA /HA/LV01 xfs defaults 1 1
```

5. Editamos con vi el fichero **/etc/export** para añadir una nueva línea que nos permita exportar mediante protocolo **NFS** a nuestros servidores el nuevo recurso:

```
vi /etc/exports
```

...

```
/HA/LV01 192.168.100.0/255.255.255.0(rw,async,no_root_squash)
```

6. Recargamos la exportación de los recursos **NFS** en caliente sin tener que reiniciar la cabina de almacenamiento mediante:

```
exportfs -ra
```

2.7.2 Montar el nuevo SR en XenCenter.

El segundo paso consistirá en agregar como Storage Resource (SR) el recurso NFS que acabamos de crear a nuestra infraestructura XenServer, para ello desde nuestro XenCenter, seleccionamos uno de los servidores de nuestro pool e iremos a la pestaña "Storage" donde podremos añadir el nuevo SR con la opción "New SR...". El asistente de creación de SR ya lo conocemos de pasos anteriores.

Pulsaremos el botón siguiente ..."Next", y escribimos el nombre de nuestro SR, por ejemplo HA-LV01, pulsamos de nuevo "Next" y nos preguntará sobre los datos de conexión a la cabina para este nuevo elemento del Storage Repository (SR). En el cuadro de texto "Share Name", introducimos el nombre con el que está compartido el recurso (path), en nuestro caso: 192.168.100.101:/HA/LV01. No es necesario poner nada en "Advanced Options", pulsamos: "Finish".

Tras una breve espera, el proceso terminará y tendremos montado el nuevo Recurso de Almacenamiento para usar con la HA.

2.7.3 Habilitar HA.

Una vez terminado el proceso de creación, ya podemos habilitar para el "pool" la alta disponibilidad, iremos a la pestaña "HA" para configurarla y se nos mostrará este asistente:

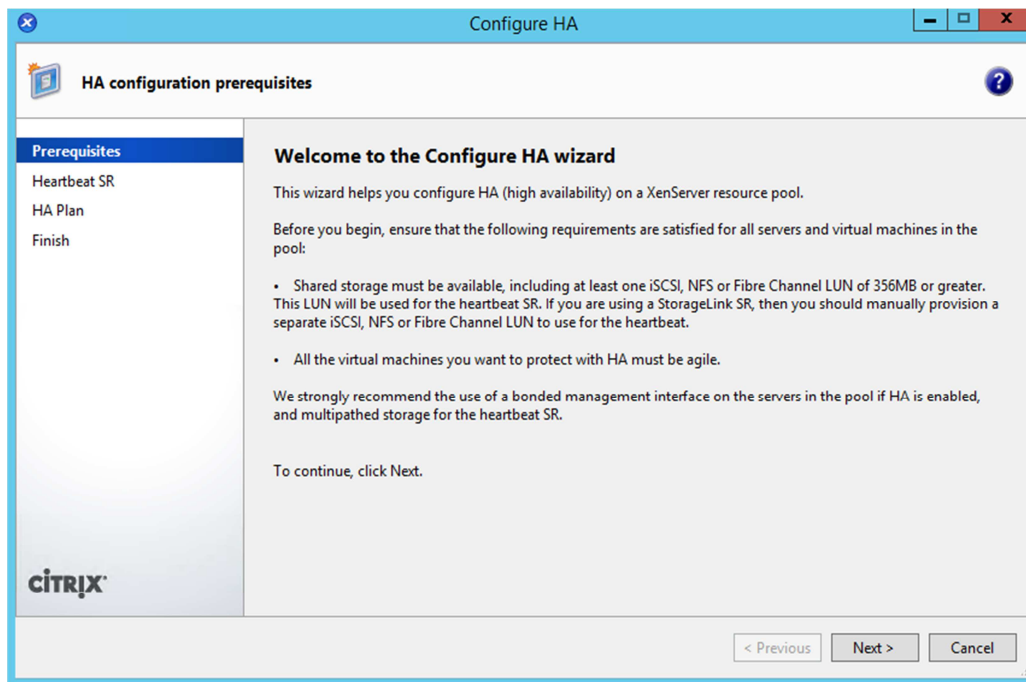


Figura 18 Asistente de configuración de HA en XenCenter, página de inicio.

Pulsamos el botón "Next" para pasar la ventana de inicio y seleccionamos el repositorio que acabamos de crear. Es importante que no seleccionemos el almacén que ya teníamos para las máquinas virtuales, ya que nos destruiría las que ya tuviésemos. Aquí por tanto seleccionamos el que en el paso anterior hemos nombrado: "HA-LV01" (y que indica en Description que es un NFS SR[192.168.100.101:/HA/LV01]):

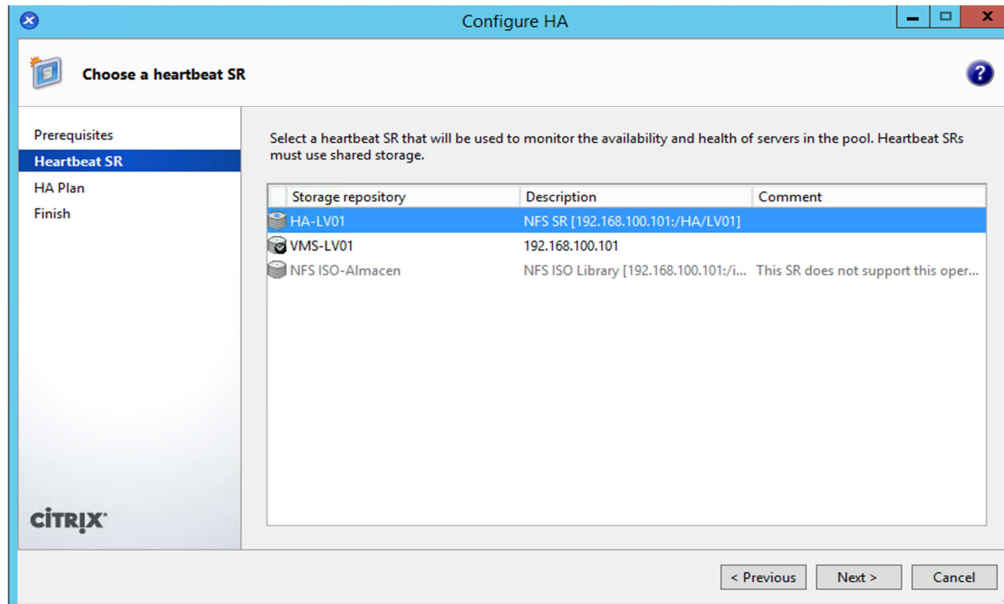


Figura 19 Selección de repositorio de datos (SR) para la función de Alta Disponibilidad (HA).

Por último el asistente nos pregunta en qué máquinas virtuales queremos habilitar la alta disponibilidad. En nuestro caso seleccionamos aquí las que hemos considerado como "críticas", por ejemplo, "broker.dominio.compañia.es", que ya hemos visto que tiene un papel fundamental para que funcionen las prácticas de laboratorio y la que aloja XenCenter" que nos ayuda a gestionar y supervisar toda la infraestructura. Para ello seleccionamos la opción "Restart" para que el sistema la vuelva a arrancar si se apagan en la casilla "HA restart priority", como vemos en el siguiente volcado de pantalla donde las máquinas críticas ya aparecen con la opción RESTART habilitada:

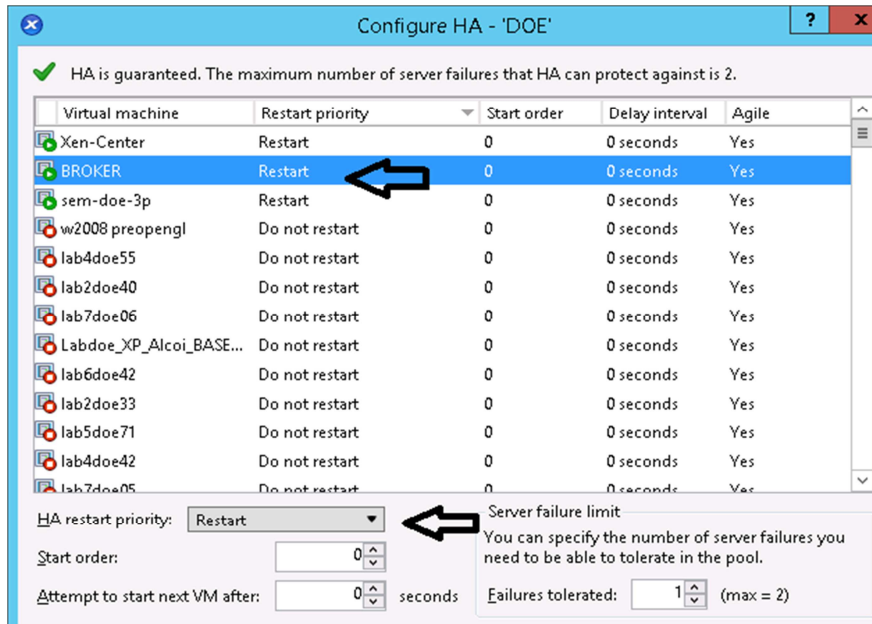


Figura 20 HA. Configuración de máquinas virtuales con Alta Disponibilidad habilitada.

Con respecto a otras opciones que permite la alta disponibilidad, (priority, failures tolerated, order, etc.) no hemos modificado ningún parámetro, puesto que para nuestro caso en concreto no era necesario.

3 ANEXOS

3.1 ANEXO I: "Fichero script "Isi""

```
#!/bin/bash

#

# Calomel.org

# https://calomel.org/megacli\_isi\_commands.html

# LSI MegaRaid CLI

# Isi.sh @ Version 0.05

#

# description: MegaCLI script to configure and monitor LSI raid cards.

# Full path to the MegaRaid CLI binary

#MegaCli="/usr/local/sbin/MegaCli64"

MegaCli="/opt/MegaRAID/MegaCli/MegaCli64"

# The identifying number of the enclosure. Default for our systems is "8". Use

# "MegaCli64 -PDlist -a0 | grep "Enclosure Device"" to see what your number

# is and set this variable.

ENCLOSURE="12"

if [ $# -eq 0 ]

then

echo ""

echo "      OBPG .. Isi.sh $arg1 $arg2"

echo "-----"

echo "status      = Status of Virtual drives (volumes)"
```

```
echo "drives    = Status of hard drives"

echo "ident \$slot = Blink light on drive (need slot number)"

echo "good \$slot = Simply makes the slot \"Unconfigured(good)\" (need slot number)"

echo "replace \$slot = Replace \"Unconfigured(bad)\" drive (need slot number)"

echo "progress  = Status of drive rebuild"

echo "errors    = Show drive errors which are non-zero"

echo "bat       = Battery health and capacity"

echo "batrelearn = Force BBU re-learn cycle"

echo "logs      = Print card logs"

echo "checkNemail = Check volume(s) and send email on raid errors"

echo "allinfo   = Print out all settings and information about the card"

echo "settime   = Set the raid card's time to the current system time"

echo "setdefaults = Set preferred default settings for new raid setup"

echo ""

exit

fi

# General status of all RAID virtual disks or volumes and if PATROL disk check
# is running.

if [ $1 = "status" ]

then

    $MegaCli -LDInfo -Lall -aALL -NoLog

    echo "#####"

    $MegaCli -AdpPR -Info -aALL -NoLog

    echo "#####"
```

```

    $MegaCli -LDCC -ShowProg -LALL -aALL -NoLog

    exit

fi

# Shows the state of all drives and if they are online, unconfigured or missing.

if [ $1 = "drives" ]

    then

        $MegaCli -PDlist -aALL -NoLog | egrep 'Slot|state' | awk '/Slot/{if (x)print x;x=""};}{x=(!x)?$0:x" -
"$0;}END{print x;}' | sed 's/Firmware state://g'

    exit

fi

# Use to blink the light on the slot in question. Hit enter again to turn the blinking light off.

if [ $1 = "ident" ]

    then

        $MegaCli -PdLocate -start -physdrv[$ENCLOSURE:$2] -a0 -NoLog

        logger "`hostname` - identifying enclosure $ENCLOSURE, drive $2 "

        read -p "Press [Enter] key to turn off light..."

        $MegaCli -PdLocate -stop -physdrv[$ENCLOSURE:$2] -a0 -NoLog

    exit

fi

# When a new drive is inserted it might have old RAID headers on it. This
# method simply removes old RAID configs from the drive in the slot and make
# the drive "good." Basically, Unconfigured(bad) to Unconfigured(good). We use
# this method on our FreeBSD ZFS machines before the drive is added back into
# the zfs pool.

if [ $1 = "good" ]

```

```

then

    # set Unconfigured(bad) to Unconfigured(good)

    $MegaCli -PDMakeGood -PhysDrv[$ENCLOSURE:$2] -a0 -NoLog

    # clear 'Foreign' flag or invalid raid header on replacement drive

    $MegaCli -CfgForeign -Clear -aALL -NoLog

exit

fi

# Use to diagnose bad drives. When no errors are shown only the slot numbers
# will print out. If a drive(s) has an error you will see the number of errors
# under the slot number. At this point you can decided to replace the flaky
# drive. Bad drives might not fail right away and will slow down your raid with
# read/write retries or corrupt data.

if [ $1 = "errors" ]

    then

        echo "Slot Number: 0"; $MegaCli -PDlist -aALL -NoLog | egrep -i 'error|fail|slot' | egrep -v '0'

    exit

fi

# status of the battery and the amount of charge. Without a working Battery
# Backup Unit (BBU) most of the LSI read/write caching will be disabled
# automatically. You want caching for speed so make sure the battery is ok.

if [ $1 = "bat" ]

    then

        $MegaCli -AdpBbuCmd -aAll -NoLog

    exit

```



```
fi
```

```
# Force a Battery Backup Unit (BBU) re-learn cycle. This will discharge the  
# lithium BBU unit and recharge it. This check might take a few hours and you  
# will want to always run this in off hours. LSI suggests a battery relearn  
# monthly or so. We actually run it every three(3) months by way of a cron job.  
# Understand if your "Current Cache Policy" is set to "No Write Cache if Bad  
# BBU" then write-cache will be disabled during this check. This means writes  
# to the raid will be VERY slow at about 1/10th normal speed. NOTE: if the  
# battery is new (new bats should charge for a few hours before they register)  
# or if the BBU comes up and says it has no charge try powering off the machine  
# and restart it. This will force the LSI card to re-evaluate the BBU. Silly  
# but it works.
```

```
if [ $1 = "batrelearn" ]
```

```
then
```

```
    $MegaCli -AdpBbuCmd -BbuLearn -aALL -NoLog
```

```
exit
```

```
fi
```

```
# Use to replace a drive. You need the slot number and may want to use the  
# "drives" method to show which drive in a slot is "Unconfigured(bad)". Once  
# the new drive is in the slot and spun up this method will bring the drive  
# online, clear any foreign raid headers from the replacement drive and set the  
# drive as a hot spare. We will also tell the card to start rebuilding if it  
# does not start automatically. The raid should start rebuilding right away  
# either way. NOTE: if you pass a slot number which is already part of the raid
```

```

# by mistake the LSI raid card is smart enough to just error out and _NOT_
# destroy the raid drive, thankfully.

if [ $1 = "replace" ]
then
    logger "`hostname` - REPLACE enclosure $ENCLOSURE, drive $2 "
    # set Unconfigured(bad) to Unconfigured(good)
    $MegaCli -PDMakeGood -PhysDrv[$ENCLOSURE:$2] -a0 -NoLog
    # clear 'Foreign' flag or invalid raid header on replacement drive
    $MegaCli -CfgForeign -Clear -aALL -NoLog
    # set drive as hot spare
    $MegaCli -PDHSP -Set -PhysDrv [$ENCLOSURE:$2] -a0 -NoLog
    # show rebuild progress on replacement drive just to make sure it starts
    $MegaCli -PDRbld -ShowProg -PhysDrv [$ENCLOSURE:$2] -a0 -NoLog
exit
fi

# Print all the logs from the LSI raid card. You can grep on the output.
if [ $1 = "logs" ]
then
    $MegaCli -FwTermLog -Dsply -aALL -NoLog
exit
fi

# Use to query the RAID card and find the drive which is rebuilding. The script
# will then query the rebuilding drive to see what percentage it is rebuilt and
# how much time it has taken so far. You can then guess-ti-mate the

```

```

# completion time.

if [ $1 = "progress" ]

    then

        DRIVE=`$MegaCli -PDlist -aALL -NoLog | egrep 'Slot|state' | awk '/Slot/{if (x)print
x;x="";}{x=(!x)?$0:x" -"$0;}END{print x;}' | sed 's/Firmware state://g' | egrep build | awk '{print
$3}`

        $MegaCli -PDRbld -ShowProg -PhysDrv [$ENCLOSURE:$DRIVE] -a0 -NoLog

    exit

fi

# Use to check the status of the raid. If the raid is degraded or faulty the
# script will send email to the address in the $EMAIL variable. We normally add
# this method to a cron job to be run every few hours so we are notified of any
# issues.

if [ $1 = "checkNemail" ]

    then

        EMAIL="tecnicos-doe@upv.es"

        # Check if raid is in good condition

        STATUS=`$MegaCli -LDInfo -Lall -aALL -NoLog | egrep -i 'fail|degrad|error'`

        #STATUS=`$MegaCli -LDInfo -Lall -aALL -NoLog`

        # On bad raid status send email with basic drive information

        if [ "$STATUS" ]; then

            $MegaCli -PDlist -aALL -NoLog | egrep 'Slot|state' | awk '/Slot/{if (x)print x;x="";}{x=(!x)?$0:x"
-"$0;}END{print x;}' | sed 's/Firmware state://g' | mail -r tecnicos@doe.upv.es -s `hostname` -
RAID Notification' $EMAIL

        fi

    fi

```

```

fi

# Use to print all information about the LSI raid card. Check default options,
# firmware version (FW Package Build), battery back-up unit presence, installed
# cache memory and the capabilities of the adapter. Pipe to grep to find the
# term you need.

if [ $1 = "allinfo" ]

    then

        $MegaCli -AdpAllInfo -aAll -NoLog

    exit

fi

# Update the LSI card's time with the current operating system time. You may
# want to setup a cron job to call this method once a day or whenever you
# think the raid card's time might drift too much.

if [ $1 = "settime" ]

    then

        $MegaCli -AdpGetTime -aALL -NoLog

        $MegaCli -AdpSetTime `date +%Y%m%d` `date +%H:%M:%S` -aALL -NoLog

        $MegaCli -AdpGetTime -aALL -NoLog

    exit

fi

# These are the defaults we like to use on the hundreds of raids we manage. You
# will want to go through each option here and make sure you want to use them
# too. These options are for speed optimization, build rate tweaks and PATROL
# options. When setting up a new machine we simply execute the "setdefaults"

```

method and the raid is configured. You can use this on live raids too.

if [\$1 = "setdefaults"]

then

Read Cache enabled specifies that all reads are buffered in cache memory.

\$MegaCli -LDSetProp -Cached -LAll -aAll -NoLog

Adaptive Read-Ahead if the controller receives several requests to sequential sectors

\$MegaCli -LDSetProp ADRA -LALL -aALL -NoLog

Hard Disk cache policy enabled allowing the drive to use internal caching too

\$MegaCli -LDSetProp EnDskCache -LAll -aAll -NoLog

Write-Back cache enabled

\$MegaCli -LDSetProp WB -LALL -aALL -NoLog

Continue booting with data stuck in cache. Set Boot with Pinned Cache Enabled.

\$MegaCli -AdpSetProp -BootWithPinnedCache -1 -aALL -NoLog

PATROL run every 672 hours or monthly (RAID6 77TB @60% rebuild takes 21 hours)

\$MegaCli -AdpPR -SetDelay 672 -aALL -NoLog

Check Consistency every 672 hours or monthly

\$MegaCli -AdpCcSched -SetDelay 672 -aALL -NoLog

Enable autobuild when a new Unconfigured(good) drive is inserted or set to hot spare

\$MegaCli -AdpAutoRbld -Enbl -a0 -NoLog

RAID rebuild rate to 60% (build quick before another failure)

\$MegaCli -AdpSetProp \{RebuildRate -60\} -aALL -NoLog

RAID check consistency rate to 60% (fast parity checks)

\$MegaCli -AdpSetProp \{CCRate -60\} -aALL -NoLog

Enable Native Command Queue (NCQ) on all drives

```
$MegaCli -AdpSetProp NCQEnbl -aAll -NoLog

# Sound alarm disabled (server room is too loud anyways)

$MegaCli -AdpSetProp AlarmDsbl -aALL -NoLog

# Use write-back cache mode even if BBU is bad. Make sure your machine is on UPS too.

$MegaCli -LDSetProp CachedBadBBU -LAll -aAll -NoLog

# Disable auto learn BBU check which can severely affect raid speeds

OUTBBU=$(mktemp /tmp/output.XXXXXXXXXXX)

echo "autoLearnMode=1" > $OUTBBU

$MegaCli -AdpBbuCmd -SetBbuProperties -f $OUTBBU -a0 -NoLog

rm -rf $OUTBBU

exit

fi

### EOF ###
```

3.2 ANEXO II: Manejo de unidades USB. Auto-montaje USB:

Al coexistir en el mismo sistema operativo de los equipos físicos, tanto el sistema de virtualización local (Oracle VirtualBox) como el cliente de escritorio remoto, tendremos conflicto a la hora de montar los dispositivos USB, ya que cada uno de ambos sistemas lo realiza de maneras diferentes.

Para dar una solución a este conflicto, hemos mantenido las reglas de montaje de VirtualBox (/etc/udev/rules.d/90-usb-virtualbox.rules) creando el script: /sbin/vbox-automount-usb.sh

El cual añade al original de VirtualBox las reglas necesarias para que funcione también cuando utilizamos el cliente “ligero” de conexión a escritorio remoto. Combinando este script con el uso de variables en el arranque (/boot/grub/grub.conf) obtenemos el resultado esperado.

El contenido del script se muestra a continuación:

```

echo ID_VENDOR:ID_MODEL ${ID_VENDOR}:${ID_MODEL} > /tmp/udev.log
echo ID_VENDOR_ID:ID_MODEL_ID ${ID_VENDOR_ID}:${ID_MODEL_ID} >> /tmp/udev.log
echo DEVNAME: $DEVNAME >> /tmp/udev.log

#extraemos el parámetro que se ha usado en el menú grub
vm=$(awk -F "vm=" '{print $2}' /proc/cmdline | awk '{print $1}')

attach_storage()
{
    RUNNINGVM=$(/usr/bin/vboxmanage list runningvms | sed -e 's/^".*".*{\(.*\)}\1/')
    set `usr/bin/lusb -d ${ID_VENDOR_ID}:${ID_MODEL_ID} | sed 's/:.*//g`

    while [ ! -z "$1" ]; do

        case $1 in

            Bus) shift

                busdevice="$1"

                ;;

            Device) shift

                busdevice=${busdevice}"/$1"

                ;;

        esac

        shift

    done

    if [ ! -z "$busdevice" ]; then

        address=$(/usr/bin/vboxmanage list usbhost | grep "Address:" | grep $busdevice

        | sed -e 's/Address:/' -e 's/^[ \t]*//')

        if [ ! -z "$address" ]; then

```

```

        /usr/bin/vboxmanage controlvm $RUNNINGVM usbattach $address
    fi
fi
}

#A continuación la parte que da solución al cliente ligero rdesktop, desmontamos lo que haya por
si acaso

case $DEVNAME in
dev/sd[a-z][1-9])
    if [ "$vm" == "thin" ]; then
        umount -l /media/usb
        mount -o sync $DEVNAME /media/usb
    else
        attach_storage;
    fi
;;

#Para el caso de que sea un usb sin particiones (sda, sdb, sdc...) y estemos en modo conexión
escritorio remoto. Desmontamos todo lo que haya por si acaso.

/dev/sd[a-z]|)
    if [ "$vm" == "thin" ]; then
        umount -l /media/usb
        mount -o sync $DEVNAME /media/usb
    else
        attach_storage;
    fi
;;

```


<pre> fi esac </pre>

Tabla 19 Contenido del script manejo unidades USB: *vbox-automount-usb.sh*

3.3 ANEXO III: Desarrollo de una Interfaz gráfica “Web-app” en “actualizador.dominio.compañia.es”.

INTRODUCCIÓN:

Con el fin de facilitar la administración de los equipos físicos de los laboratorios, se desarrolló una interfaz gráfica desarrollada con lenguaje “php” que se ejecuta vía Navegador WEB. Esta interfaz realiza una serie de funciones y utilidades como la generación de ficheros en formato CSV que luego como hemos visto usaremos en distintos scripts, copia de archivos mediante protocolo “SCP” de un equipo a otro o a varios equipos, comprobación del estado de los equipos de los laboratorios (encendidos o apagados), reiniciar y apagar los equipos, etc.

ORGANIZACIÓN DE LA APLICACIÓN:

Al iniciar el navegador web y acceder a la dirección: <http://actualizador.dominio.compañia.es>, se nos mostrará la pantalla que podemos observar en la siguiente figura, en la cual debemos introducir el usuario y la contraseña para iniciar sesión:



Figura 21 Pantalla de inicio de sesión.

Una vez, hemos iniciado sesión en la aplicación podremos ver la siguiente pantalla:

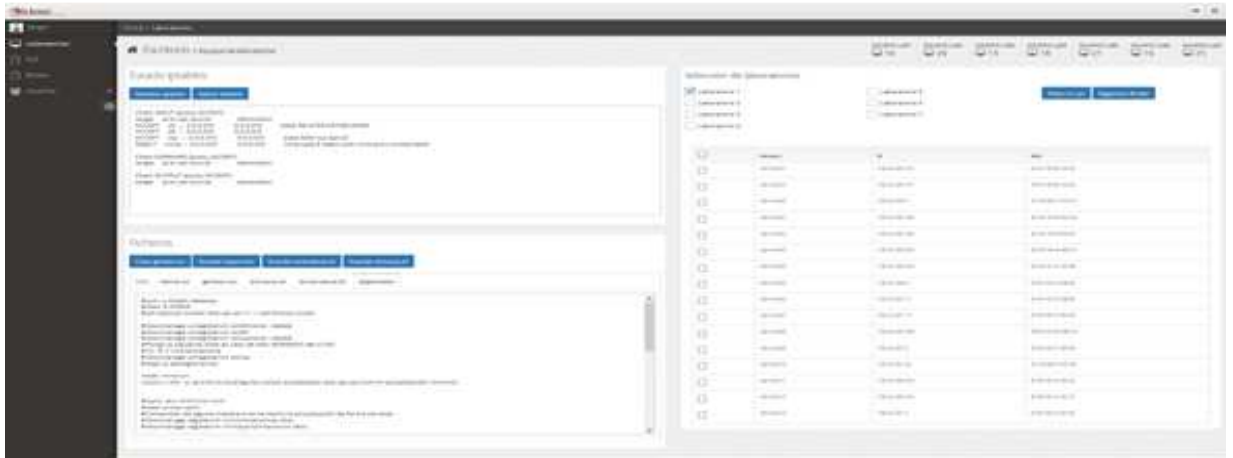


Figura 22 Pantalla gestión de laboratorios

Como podemos observar en la figura, esta pantalla consta un menú vertical en el lado izquierdo donde podremos ir a los distintos apartados de la aplicación:

- **Laboratorios.** Donde se recogen opciones orientadas al trabajo con los equipos de los laboratorios, se divide en tres secciones: “Estado iptables”, “Selección de laboratorios” y “Ficheros”. A continuación, explicaremos la funcionalidad de cada una de estas secciones.

Sección iptables:



Figura 23 Sección iptables.

Esta sección consta de dos botones y un área de texto. El botón, **Aplicar ipTables**, aplicará las reglas creadas para el firewall con los ordenadores que previamente habremos seleccionado en la selección de equipos. El botón, **Resetear ipTables**, restaura la configuración inicial de firewall. Para ver si los cambios han surtido efecto, una vez que hemos pulsado sobre los botones anteriores, disponemos del área de texto que hay debajo de ellos.

Sección Selección de laboratorios:

	Nombre	IP	MAC
<input type="checkbox"/>	lab3doe20	158.42.200.49	00-1F-F3-46-88-46
<input checked="" type="checkbox"/>	lab3doe21	158.42.200.56	00-1F-F3-45-4F-19
<input type="checkbox"/>	lab3doe22	158.42.200.155	00-1F-F3-46-6C-93

Figura 24. Sección Selección de equipos.

Esta sección contiene unas casillas de selección, que nos permitirán mostrar los equipos que tienen los laboratorios. Una vez mostrados los equipos, los podremos seleccionar para ejecutar acciones sobre ellos. Esta sección contiene dos botones, “**Wake on Lan**”, que permite encender ordenadores apagados, y “**Regenerar bróker**”, que creará un fichero de intercambio de datos tipo “csv” con la base de datos de equipos, para ser utilizado posteriormente en las tareas de administración que hemos visto que realizamos en "broker.dominio.compañia.es".

Sección Ficheros:

```
#yum -y install rdesktop
#read -p PARAR
#ssh-keyscan broker.doe.upv.es >> ~/.ssh/known_hosts

#vboxmanage unregistervm win81 clone --delete
#vboxmanage unregistervm win81
#vboxmanage unregistervm winxpclone --delete
#Pongo la siguiente linea en caso de MAL BORRADO del CLON
#rm -R -f /virt/winxpclone
#vboxmanage unregistervm winxp
#read -p desregistramos

mkdir /mnt/virt
mount -t nfs -o vers=3,rw,hard,bg,intr,nolock actualizador.doe.upv.es:/virt
/vm-actualizando/ /mnt/virt

#rsync -avz /mnt/virt/ /virt/
#read -p tras rsync
#Comprobar de alguna manera si se ha hecho la actualización de forma correcta
```

Figura 25. Sección Ficheros

Como vemos en esta tenemos varios botones y un cuadro de texto que nos permite visualizar y editar los ficheros que seleccionamos en las pestañas, así como, ver el contenido del directorio /virt, lo cual nos resultará de ayuda a la hora de distribuir nuevas máquinas virtuales en el modo local.

- **SCP.** En este menú disponemos de opción para copia de archivos usando el protocolo seguro SCP.

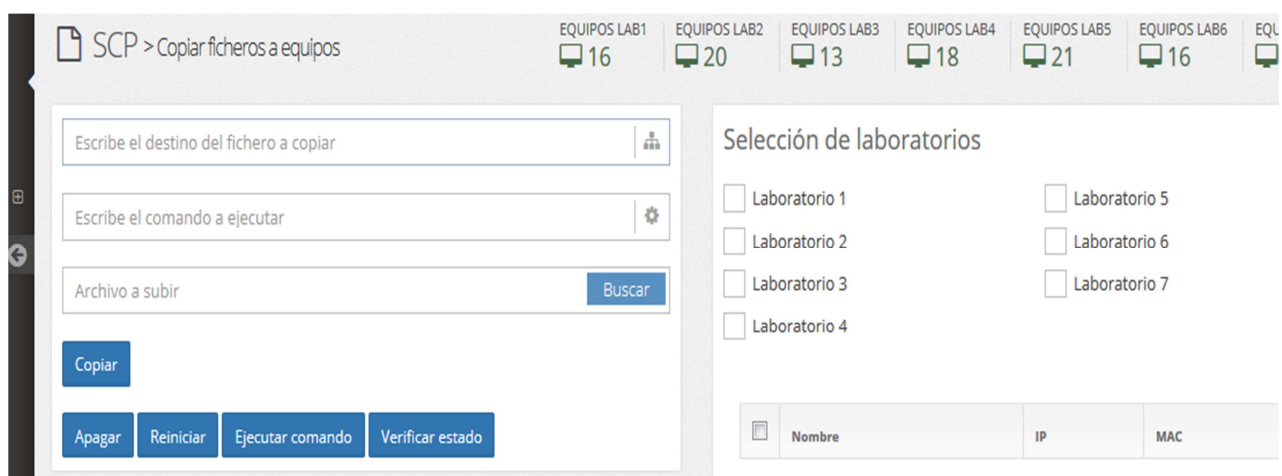


Figura 26. Pantalla SCP

En la figura 26, vemos el volcado de la sección de control de los equipos en remoto y la función de copia de ficheros mediante la orden scp. El funcionamiento es similar al apartado anterior, tenemos una sección para realizar la selección de equipos y otra donde ejecutamos las acciones, en este caso, copia de ficheros.



Figura 27 Detalle de la Sección SCP, copia de ficheros a equipos.

En esta figura vemos que disponemos de varios botones, cada uno para realizar una determinada acción. El botón, **Copiar**, copiará un archivo que habremos buscado en nuestro equipo, al destino que le digamos en los ordenadores seleccionados previamente desde la sección, **selección de equipos**. El botón, **Ejecutar comando**, ejecutara una orden, como por ejemplo, ls -la, etc... En los equipos seleccionados. El botón, **Apagar**, apagara los equipos seleccionados, el botón, **Reiniciar**, reinicia los equipos seleccionados y por último el botón, **Verificar estado**, nos indicará qué equipos están encendidos o apagados de la selección que hayamos hecho.

- **Bróker**. En este menú encontramos un listado o informe sobre la creación del fichero broker.csv que tenemos programada todos los días a las 03:00 AM y donde se recoge la BBDD de equipos físicos de los laboratorios.

ID	Fecha	Hora	Estado
1	2015-03-02	10:59:06	OK
2	2015-03-02	11:15:01	OK
3	2015-03-03	03:00:03	OK
4	2015-03-03	11:30:01	OK
5	2015-03-03	11:40:01	OK
6	2015-03-03	11:43:01	OK
7	2015-03-04	03:00:01	OK
8	2015-03-04	11:34:34	OK

Figura 28 Pantalla Bróker.

- **Usuarios.** Se ha añadido este menú con la finalidad de implementar una gestión de usuarios que pueden acceder a la herramienta, así como realizar auditoría de las tareas que realizan.

Home / Usuarios / Nuevo usuario

Nuevo usuario

Nombre Apellidos

Correo electrónico

Login TIPO DE USUARIO

Contraseña Confirme contraseña

Listar usuarios

Login	Nombre	Apellidos	Email	Tipo usuario	Estado
fvillar	Paco	Villar		Administrador	ACTIVO
juchaful	Julio		juchaful	Operador	ACTIVO
raublahe	Raul		raublahe	Administrador	ACTIVO

Figura 29 Pantalla de gestión de usuarios.

La figura anterior nos muestra la pantalla de usuarios, a través de ella, se nos permitirá crear usuarios nuevos y ver cuáles están activos en la aplicación.

Listar accesos de usuarios

Login	fecha	hora	ip
fvillar	06-07-2015	14:23:58	158.42.98.108
fvillar	22-06-2015	11:23:44	158.42.132.181
fvillar	30-05-2015	11:43:36	158.42.98.209
fvillar	25-05-2015	12:55:08	158.42.98.209

Figura 30. Pantalla listar accesos de usuarios.

Por último, como podemos ver en esta figura, podemos obtener un listado en el que se nos mostrará los distintos accesos para cada uno de los usuarios de la aplicación.

3.4 ANEXO IV: Modificar el servidor “maestro” del pool de XenServer.

Puede sucedernos, que el servidor que tenemos asignado con el rol de maestro del “pool”, en un momento dado no esté accesible. Cuando esto sucede, XenServer automáticamente designará y le dará el rol a otro de los servidores miembros del pool.

Si queremos volver a tener como maestro del pool el servidor que nosotros habíamos configurado como tal, deberemos seguir los siguientes pasos:

- 1.- Deshabilitar temporalmente la “Alta disponibilidad” (desde la interfaz gráfica de XenCenter - Pestaña HA del Pool - Opción “Disable HA...”).
- 2.- En Consola debemos ejecutar este comando en cualquier miembro del pool para conseguir el identificador uuid del servidor que queremos volver a convertir en maestro:

xe host-list

```
uuid ( RO)          : cd68df25-a412-4d92-bd15-4d0c784623d0
```

```
name-label ( RW): pegasus
```

```
name-description ( RW): Default install of XenServer
```

```
uuid ( RO)          : d25a84fe-6224-4585-8af4-2c7e56e98445
```

```
name-label ( RW): perseus
```

```
name-description ( RW): Default install of XenServer
```

```
uuid ( RO)          : 92514601-4ea1-4ae7-a461-194558a37e74
```

```
name-label ( RW): cetus
```

```
name-description ( RW): Default install of XenServer
```

- 3.- Una vez sabemos el identificador de nuestro servidor ejecutamos el comando:

```
xe pool-designate-new-master host-uuid=cd68df25-a412-4d92-bd15-4d0c784623d0
```

4.- Habilitar de nuevo la alta disponibilidad (XenCenter - Pestaña HA del Pool - Opción “Enable HA...”). Seleccionando el mismo recurso de almacenamiento (SR) donde antes lo teníamos configurado (recordemos, en nuestro caso “HA-LV01”) veremos que todos los ajustes de configuración siguen estando en la misma situación, con lo que no tenemos que volver a configurarlo todo.