

Renovación del software de virtualización de servidores en una empresa

Óscar Ruiz Gallardo

Tutor TFG: Josep Lluís Berral García

Tutora GEP: Olga Pons Peregort

Especialidad: Tecnologías de la Información

24 de Enero de 2022

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Índice

1. Introducción	4
1.1. Contexto	4
1.2. Motivación inicial	5
1.2.1. Objetivo	5
1.2.2. Resultado	5
2. Análisis de requisitos	7
2.1. Situación actual	7
2.2. Problemática	7
2.3. Objetivo	8
3. Opciones actuales	9
3.1. Tecnologías disponibles	9
3.1.1. VMware vSphere	9
3.1.2. Proxmox	9
3.1.3. Red Hat Virtualization	10
3.1.4. Microsoft Hyper-V	10
3.1.5. Citrix	10
3.1.6. Oracle VM Server	10
3.1.7. Virtuozzo	11
3.2. Tecnologías candidatas	11
3.2.1. Citrix	11
3.2.2. Oracle VM Server	12
3.2.3. Proxmox	12
4. Desarrollo	13
4.1. Instalación de los nodos	13
4.2. Redundancia	14
4.2.1. Live migration	14
4.2.2. Shared storage	15
4.2.3. High-Availability	17
4.3. Migración desde VMware	20
5. Contenedores	21
6. Conclusiones técnicas	22
7. Gestión del proyecto	23
7.1. Planificación temporal	23
7.2. Gestión económica	23
8. Sostenibilidad y compromiso social	24
8.1. Matriz de sostenibilidad	24
8.1.1. Ambiental	24
8.1.2. Social	24
9. Conclusiones	25
9.1. Largo plazo	25

Glosario

- **Aprovisionamiento:** Al crear una máquina virtual se reserva una cierta cantidad de espacio de almacenamiento de un almacén de datos a archivos de disco virtual. Se puede asignar de forma gruesa, en el que el disco virtual inmediatamente ocupa todo el espacio que se le asigna y la forma fina solo ocupa lo que necesita, hasta llegar a lo que se le ha provisionado.[1]
- **Contenedor:** Un contenedor agrupa el código de una aplicación con las bibliotecas y los archivos de configuración asociados, junto con las dependencias necesarias para que a aplicación se ejecute.[2]
- **CPD:** Centro de Procesamiento de Datos, es un espacio físico donde se concentran los recursos necesarios para la gestión de la información de una organización. Incluye componentes como routers, switches, cortafuegos, sistemas de almacenamiento y servidores.[3]
- **Downtime:** El tiempo de *downtime* es el tiempo en el que un servicio en producción no está operativo.
- **ftp:** Protocolo para compartir ficheros.
- **hipervisor:** software que crea y ejecuta máquinas virtuales. Permite un ordenador anfitrión soportar distintas máquinas invitadas a partir de compartir sus recursos como la memoria y el procesamiento.[4]
- **IP:** Dirección que sirve para identificar un interfaz de un dispositivo dentro de una red.
- **iso:** Formato de archivo que contiene una copia de los datos de un disco óptico, por ejemplo un disco de instalación de un sistema operativo.
- **RAID:** *Redundant Array of Inexpensive Disks*[5] o *Redundant Array of Independent Disks*, es una tecnología de virtualización de almacenamiento que combina distintas configuraciones de discos físicos en uno o varios virtuales con el propósito de crear redundancia.
- **SAN:** *Storage Area Network*, es una red dedicada independiente que interconecta y distribuye distintos dispositivos de almacenamiento a múltiples servidores.[6]
- **SMB:** Protocolo para acceder a ficheros en red.
- **Socket:** Zócalo donde se inserta el procesador de un ordenador.
- **Spare:** Un disco de *spare* es un disco vacío que se usa para sustituir automáticamente otro disco en caso de fallo.
- **Virtualización:** La virtualización crea un entorno informático simulado, o virtual, en lugar de un entorno físico. A menudo, incluye versiones de hardware, sistemas operativos, dispositivos de almacenamiento, etc., generadas por un equipo. Esto permite a las organizaciones particionar un equipo o servidor físico en varias máquinas virtuales.[7] Se trata de una abstracción de un hardware físico
- **VMWare vSphere:** Software de virtualización de servidores de la empresa VMware.[8]
- **VPN:** Protocolo para establecer un canal seguro en conexiones en internet, se usa principalmente para conexiones remotas.

Abstract

La **virtualización** es una tecnología usada para sacar un mejor provecho del hardware y optimizar sus recursos, ahorrando costes, aumentando la flexibilidad y la portabilidad de las máquinas. En este proyecto se presenta una solución para renovar por completo el sistema de virtualización de la empresa donde trabajo actualmente como administrador de sistemas. Visto que el sistema instalado actualmente y que se ha ido usando durante los últimos 7 años ha quedado anticuado, después de valorar el mercado actual y las soluciones disponibles me he decantado por Proxmox, un sistema de código abierto y gratuito que ofrece una serie de funcionalidades muy acorde a lo que necesitamos actualmente y de cara al futuro.

1. Introducción

La virtualización es una tecnología que se cree que tiene su origen a finales de la década de 1960 y a principios de la década de 1970, cuando IBM invirtió tiempo y esfuerzo en desarrollar soluciones para la compartición de recursos de computación entre un gran grupo de usuarios, intentando así aumentar la eficiencia de los usuarios y de los recursos que comparten. Actualmente la motivación para usar esta tecnología se basa en el mismo conjunto de razones. La capacidad en un único servidor es tan alta que es imposible para la mayoría de cargas de trabajo usarlo efectivamente[9]. Primeramente, antes del resurgimiento de la virtualización el hardware estaba infrautilizado y muchos centros de datos podían tener máquinas corriendo al 10 % de su capacidad, es decir, un 90 % del potencial se estaba desperdiciando. Con la virtualización podemos aumentar ese 10 o 15 % a más del 70 % ya que permitimos asignar los recursos dinámicamente y usarlos como necesitemos. Los centros de datos además cada vez requieren de más servidores para funcionar correctamente, se estima que se pueda llegar a albergar del orden de *yottabytes* de información en la próxima década[10]. Sin virtualización, cada vez se requeriría más hardware lo que aumentaría el coste económico y el espacio a usarse, lo que también conlleva un impacto energético mayor, tanto económico como ambiental. Por último el coste administrativo también se ve afectado, reduciéndolo drásticamente cuando se aplica una solución de virtualización.[11]

Como se ve en la figura 1 un entorno virtualizado puede hacer mejor uso del hardware ya que añade una capa extra de virtualización sobre la cual se ejecutan las distintas máquinas virtuales. El objetivo principal de este proyecto es renovar por completo el entorno de virtualización del CPD de la empresa donde trabajo actualmente, por una parte actualizándolo para poder hacer uso de nuevas herramientas y por otra solucionando ciertos problemas de diseño.

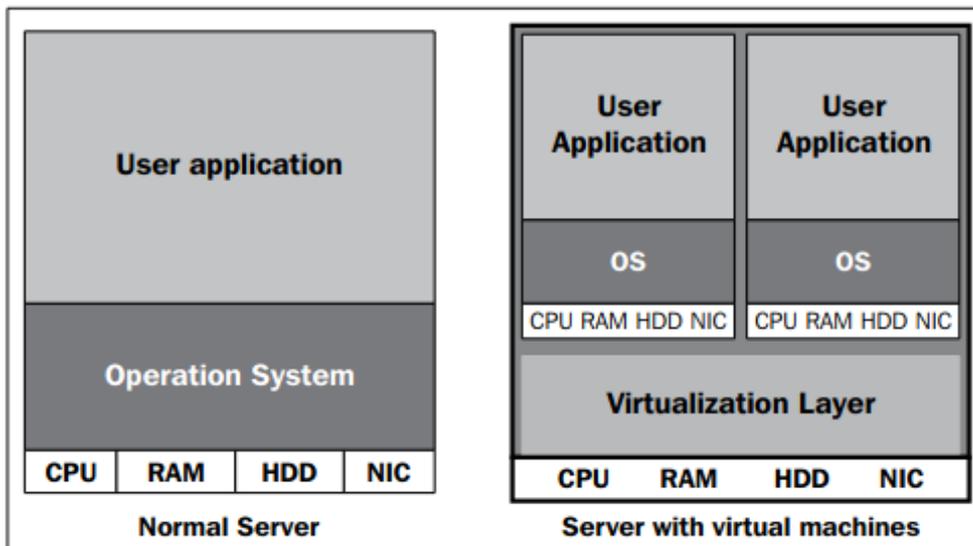


Figura 1: Diferencia entre un sistema no virtualizado (izquierda) donde el sistema operativo se ejecuta directamente sobre el hardware y un sistema virtualizado (derecha) donde se añade una capa de abstracción y el sistema operativo se ejecuta sobre hardware virtual. Fuente: Proxmox High Availability[12, p.7]

1.1. Contexto

Trabajo como administrador de sistemas en una pequeña empresa de unos 100 trabajadores. Mi trabajo consiste en mantener un centro de datos operativo, el cual mantiene los servicios de la empresa, los cuales a su vez son usados por sus empleados y además doy el soporte informático correspondiente. Entre los servicios se encuentran correo electrónico, webs, repositorios de código y de proyectos, **ftp**, **SMB**, **VPN**, servidores de cálculo, entre otros. Tenemos un pequeño centro de datos con una cierta cantidad de máquinas físicas, una parte de ellas está destinada a virtualizar los servidores que dan la mayoría de los servicios anteriormente mencionados.

Actualmente estos servidores virtuales corren encima de un sistema **VMware vSphere**, todos ellos instalados entre 2014 y 2016.

Estas instalaciones se hicieron en una época en la que no se contaban con los recursos necesarios

para hacer un estudio riguroso y simplemente se instaló un sistema popular con buen soporte y documentación. Con el tiempo ha ido quedando descuidado y presenta distintos problemas, principalmente de diseño y por el software en sí mismo.

El objetivo principal es renovar completamente el entorno de virtualización ya sea con el mismo software o uno distinto. La motivación no viene simplemente dada por la antigüedad del software actual, que pese a ser un punto a tener en cuenta porque tiene unos cinco años, otros aspectos y problemas presentes en la instalación actual hacen que vea necesaria esta renovación.

En cuanto a mi contexto como estudiante, he cursado el grado en ingeniería informática con mención en tecnologías de la información. Durante esta etapa me ha interesado especialmente el mundo de los centros de datos y la seguridad informática, así pues asignaturas que podría destacar y han tenido un impacto mayor en este proyecto y en mí como profesional son Arquitectura de Computadores(AC), Redes de Computadores(XC), Administración de Sistemas Operativos(ASO), Seguridad Informática(SI), Centros de Procesamiento de Datos(CPD) y Conceptos Avanzados de Sistemas Operativos(CASO).

1.2. Motivación inicial

Como ya he mencionado, con el paso de los años y el bajo mantenimiento, el entorno de virtualización no está a la altura de las necesidades actuales y presenta varios problemas.

Lo que más salta a la vista a la hora de gestionarlo es la baja maleabilidad que presenta. La versión gratuita de VMware que lleva instalada limita la cantidad de funciones que poder llevar a cabo y tareas como migrar máquinas virtuales no se pueden hacer con tanta facilidad y hay que buscar alternativas, como copiar las máquinas manualmente.

Cada servidor físico tiene ligado un volumen de una **SAN**. La asignación es desigual y no sigue ninguna lógica concreta a día de hoy. Es un acierto mantener el volumen separado del propio servidor físico pero es limitante tener separación entre volúmenes.

Cada servidor lógico es creado tomando espacios desiguales y sin una lógica o metodología aparente y mezclando distintos tipos de **aprovisionamiento**, desperdiciando así mucho espacio.

Pese a ser un *cluster* de máquinas, entre ellas no se comunican como tal y su único punto de conexión es el SAN, el cual internamente también separa las máquinas. Es decir, no hay un agente que permita gestionar el *cluster* como tal, son máquinas independientes que simplemente están en la misma red.

Debido al anterior punto, el *downtime* puede aumentar mucho, porque pese a que la redundancia es buena debido a que la cabina tiene un **RAID6** y un disco de *spare* y cada volumen puede ser reasignado con lo que no se perdería información; la reasignación se tiene que hacer de forma manual, lo que supone un gran downtime que puede llegar a ser problemático, sin contar con la posterior puesta en marcha de la máquina virtual en sí, que también debe hacerse de forma manual. En resumen, un fallo de un nodo supone una tarea manual por parte de un técnico que deberá estar varios minutos u horas volviendo a poner en línea esa máquina. En la figura 2 podemos ver la distribución de volúmenes virtuales. Cada máquina física, llamadas Vmwx, tiene asignada uno o varios volúmenes virtuales. Puede observarse como con los años el resultado ha acabado siendo caótico.

Además hay muchos servicios que se podrían *contenerizar*, ahorrando aún más recursos. Más adelante se introducirá el término.

1.2.1. Objetivo

Con este proyecto espero mejorar gratamente la eficiencia de la gestión del CPD, obtener una rápida respuesta ante desastres i el aprovechamiento más óptimo del hardware. Esto mejorará la calidad general del centro y será más tolerante a problemas.

Para llevar a cabo esta mejora, una vez identificado el problema he optado por decidir si se mantenía el software o se decidía optar por otro con mejores funcionalidades. Para ello he consultado las actuales soluciones disponibles en el mercado y he hecho una pequeña comparación de sus características, precio y estado. He marcado ciertos requisitos personales como que no sea software propietario, tenga un precio lo más bajo posible y tenga un buen soporte a largo plazo.

1.2.2. Resultado

El sistema de virtualización queda actualizado y tenemos una mejor respuesta ante desastres. Las personas que se verán beneficiadas de este proyecto serán los usuarios porque percibirán el servicio como más fiable y robusto, el equipo de sistemas al hacer la gestión mucho menos compleja

y ahorrar tiempo y la empresa ya que se podrán usar los equipos aún más eficientemente, reduciendo costes.

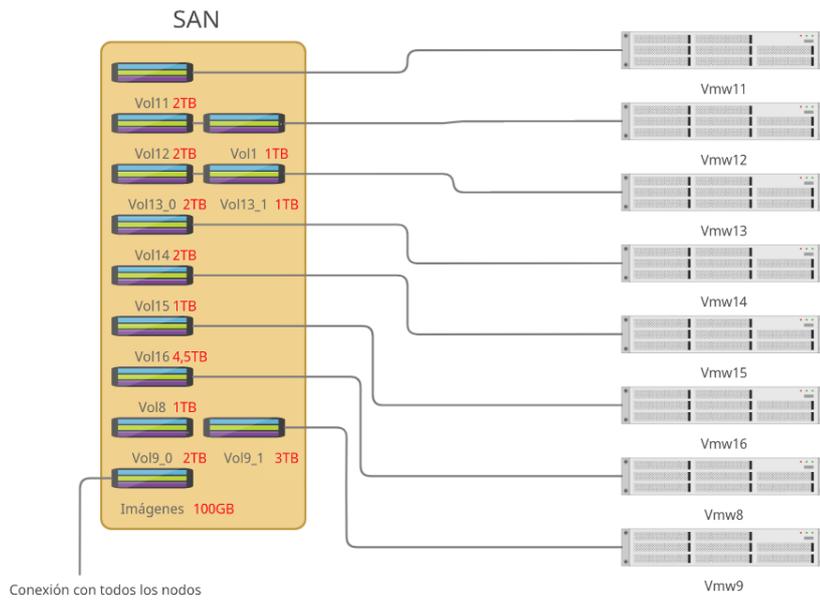


Figura 2: Distribución actual del almacenamiento. Fuente: propia

Nombre	Modelo	CPU	Memoria	SO
VMware ES-Xi 8	DELL PowerEdge R515	AMD Opteron 4334 6c 2 x 6 CPUs x 3,1GHz	32GB	VMware vSphere 5
VMware ES-Xi 9	DELL PowerEdge R715	AMD Opteron 6344 2 x 12 CPUs x 2,6GHz	64 GB	VMware vSphere 5
VMware ES-Xi 11	DELL PowerEdge R730 ST: D55KL82	Intel Xeon CPU E5-2609 v3 @1.90Ghz 2 x 6 CPU x 1,9 GHz	64 GB	VMware vSphere 6
VMware ES-Xi 12	DELL PowerEdge R630 ST: DQPGL82	Intel Xeon CPU E5-2609 v3 @1.90Ghz 2 x 6 CPU x 1,9 GHz	64 GB	VMware vSphere 6
VMware ES-Xi 13	DELL PowerEdge R630 ST: DQQPL82	Intel Xeon CPU E5-2609 v3 @1.90Ghz 2 x 6 CPU x 1,9 GHz	64 GB	VMware vSphere 6
VMware ES-Xi 14	DELL PowerEdge R630 ST: DQNML82	Intel Xeon CPU E5-2609 v3 @1.90Ghz 2 x 6 CPU x 1,9 GHz	64 GB	VMware vSphere 6
VMware ES-Xi 15	DELL PowerEdge R630 ST: DQRHL82	Intel Xeon CPU E5-2609 v3 @1.90Ghz 2 x 6 CPU x 1,9 GHz	64 GB	VMware vSphere 6
VMware ES-Xi 16	DELL PowerEdge R730 ST: D55RL82	Intel Xeon CPU E5-2609 v3 @1.90Ghz 2 x 6 CPU x 1,9 GHz	64 GB	VMware vSphere 6

Cuadro 1: Información básica de los nodos actuales

2. Análisis de requisitos

Sabemos el conjunto de problemas que nos causa el sistema actual, en esta sección cubro todos los problemas a solucionarse y como debe quedar el sistema finalmente.

2.1. Situación actual

La parte donde se almacenan y se ejecutan las máquinas virtuales gestionadas por nosotros en el CPD está formada actualmente por 8 máquinas físicas, tal como se ve en el cuadro 1.

Cada una de éstas máquinas físicas tiene dos tipos de almacenamiento, el almacenamiento interno, donde tiene instalado el **hipervisor** y un almacenamiento montado por red de una cabina de discos con las siguientes características:

Modelo: Dell Equallogic PS4100E 70-0120

ST: DNTNL82

Discos: 12 (11 + 1 Spare) x 3 TB en Raid 6 (Capacidad: 24,15 TB)

Año: 2015

Esta cabina mediante su software (Dell EqualLogic SAN Headquarters Version 3.1) está dividida en volúmenes lógicos (Figura 3), cada uno o varios de ellos ligados a una o más máquinas físicas. De esta forma tenemos el almacenamiento centralizado y la integridad de los datos no depende del estado de cada uno de los nodos del CPD.

2.2. Problemática

Cada instancia Vmware, es decir, cada nodo del CPD es independiente a los demás. Actualmente no existe un objeto *cluster* o *datacenter* que permita gestionarlo todo de una manera centralizada. Además, pese a que los volúmenes provienen de la misma cabina, son independientes entre ellos, lo que en otros escenarios nos aventaja pero en este caso nos entorpece al no haber un sistema automático de recuperación de desastres.

La gestión de los volúmenes de la cabina no es homogénea, se mezclan tipos de aprovisionamiento y no hay uniformidad en el tamaño. A esto se le añade la falta de redundancia de servidores.

Actualmente el *downtime* asociado a la caída de una máquina física es bastante alto, puede ir de una hora para servidores simples a varios días al depender de una jornada laboral, ya que hay que

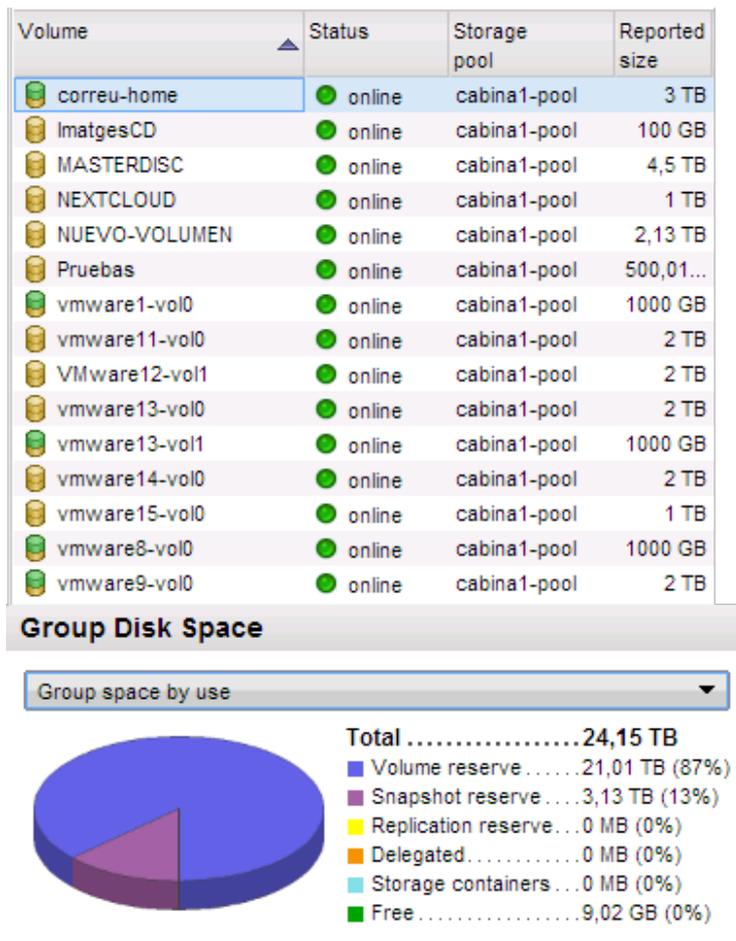


Figura 3: Volúmenes actuales de la cabina. En la figura 2 se puede ver su relación con las máquinas físicas, además se puede observar que hay un conjunto de volúmenes no asignados. Fuente: propia

manualmente asignar el volumen a otra máquina física y debido a las limitaciones del software, se deberían recrear nuevas máquinas virtuales a las que asignar los discos.

Una migración actual de máquina virtual se realiza con el software VMware vCenter Converter especificando el servidor a desplazar y la máquina virtual de destino (la cual tiene que ser creada con anterioridad).

El software más reciente instalado actualmente es vSphere 6.0, su soporte terminó en marzo de 2020.[13][14]

2.3. Objetivo

El nuevo sistema debe poder solucionar todas las problemáticas que nos presenta el actual. Primero deberíamos contar con un objeto *cluster* virtual que permita la gestión centralizada de todas las máquinas virtuales. Esto permitiría además instaurar un buen sistema de redundancia para reducir mucho más el tiempo de *downtime* en caso de fallo de un nodo. Homogeneizar el reparto de volúmenes según la estrategia a utilizar. A priori la estrategia de separar por volúmenes lógicos la veo obsoleta y es mucho más útil mantener un volumen solo para máquinas virtuales las cuales repartirán sus ejecuciones a través de los nodos. Solamente habría que mantener separación en aquellos recursos que no necesiten comunicación entre sí.

La cabina debe contar con un buen sistema de RAID además de los propios *backups* de ciertos tipos de datos.

Además la solución se debe adaptar al nuevo modelo de servidor general:

Dell PowerEdge R540
 2 x Intel Xeon Gold 6230 2,1 GHz
 2 x 32 GB RDIMM, 3200MT/s

Lo cual no es un gran problema debido a que es un modelo de servidor similar a los actuales.

3. Opciones actuales

Con tal de encontrar una mejor opción he hecho un pequeño estudio sobre las opciones más populares en el mercado actualmente para ver cual se podría ajustar mejor a nuestras necesidades, de éstas escogería las más convenientes y las probaría para acabar de elegir una única. El criterio para escoger una tecnología está basado en distintos puntos. Primero que pueda cubrir perfectamente los objetivos del proyecto, segundo que tenga un coste lo más bajo posible de licencia y priorizar una opción de código abierto frente a otra cerrada.

3.1. Tecnologías disponibles

Aquí se muestra un conjunto de tecnologías mostradas en distintos artículos divulgativos sobre virtualización[15].

3.1.1. VMware vSphere

Como punto fuerte tenemos que VMware es el líder del mercado en virtualización de servidores[16], por otro lado, su coste puede llegar a ser mucho más alto que el equivalente en otra empresa.

Cuenta con un gran número de componentes. El hipervisor ESXi, vCenter para gestión de servidores, vMotion para migración de máquinas y más.

En el cuadro 2 podemos ver las limitaciones del software en las tres últimas versiones. Vemos

	6.5	6.7	7u1
Host por cluster	64	64	96
MV por cluster	8000	8000	8000
Hosts por VMware vCenter Server	2000	2000	2500
MV encendidas por vCenter Server	25K	25K	30K
CPU por host	576	768	768
RAM por host	12 TB	16 TB	16 TB
MV por host	1024	1024	1024
CPU virtuales por MV	128	128	128
RAM virtual por MV	6 TB	6 TB	6 TB

Cuadro 2: Límites de distintas versiones de VMware ESXi[17].

como las capacidades del software superan ampliamente a nuestras necesidades, de hecho cualquier solución de las presentadas ofrecen unas capacidades muy por encima de las requeridas por lo que no es necesario entrar en detalle en ese aspecto.

Tenemos dos versiones, la pensada para pequeños negocios y la versión para empresas. El precio más barato que se puede conseguir es el paquete VMware vSphere Essentials Kit[18], de la versión para pequeños negocios. Es una licencia perpetua para 6 procesadores, es decir 3 servidores con 2 cada uno y una de vCenter, además de actualizaciones durante 1 año. Su precio es de 576,96 dólares estadounidenses.

3.1.2. Proxmox

Proxmox VE es un entorno de gestión de servidores completo de código abierto, publicado bajo la licencia GNU AGPLv3, lo que significa que es completamente gratuito para descargar, usarse y compartirse. La suscripción es un servicio adicional diseñado para ayudar a negocios y profesionales a mantener sus instalaciones actualizadas ya que da acceso a un repositorio estable entre otras cosas.[19] Integra el hipervisor KVM y Linux Containers (LXC), además incorpora una interfaz web para gestionar los clusters y usar todo el conjunto de herramientas.[20] Permite hasta 32 nodos por *cluster*. [21]

Ya que KVM es de código abierto y está disponible como un módulo del kernel, viene con cero coste y listo para usarse, lo que abarata el coste final de la solución.

Como permite su uso gratuitamente podríamos considerar esta opción como coste cero, aunque por comparar, hay una suscripción de comunidad que cuesta 90€ anuales por *socket* de CPU. Esto te da acceso al repositorio empresarial, todas las características y soporte de la comunidad.

3.1.3. Red Hat Virtualization

Red Hat también está basado en KVM y tiene herramientas de gestión para virtualizar recursos, procesos y aplicaciones. El Red Hat Virtualization Manager (su software de gestión) proporciona un sistema de gestión centralizado con interfaz gráfica.

En el cuadro 3 se pueden ver sus limitaciones. Como he mencionado con anterioridad, supera con creces las necesidades.

Cada suscripción tiene un precio de 999 dólares estadounidenses por par de *sockets* por año.[22]

	Red Hat Enterprise Linux 5 Server with KVM	Red Hat Enterprise Linux 6 Server with KVM	Red Hat Enterprise Linux 7 Server with KVM	Red Hat Enterprise Linux 8 Server with KVM
Maximum number of virtual CPUs in virtualized guest	16	240	240	384
Maximum memory in virtualized guest	512 GB	4 TB	6 TB	6 TB
Minimum memory in virtualized guest	512 MB	512 MB	512 MB	512 MB

Cuadro 3: Límites de virtualización para Red Hat Enterprise Linux con KVM[23]

3.1.4. Microsoft Hyper-V

Consiste en varias partes: el hipervisor Windows, Hyper-V Virtual Machine Management Service, el Windows Management Instrumentation (WMI) provider, el virtual machine bus (VMbus), virtualization service provider (VSP) and virtual infrastructure driver (VID). Además de herramientas y módulos para la gestión[24]. La preferencia por soluciones de código abierto nos hace descartar esta opción rápidamente.

Ofrecen dos planes:

- Standard: \$1323 hasta 16 núcleos.
- Datacenter: \$3607 hasta 16 núcleos.

Los 16 núcleos asumen que cada procesador cuenta con 8 núcleos. Una máquina física tiene dos procesadores así que se necesitaría una licencia por máquina.

3.1.5. Citrix

El hipervisor Citrix está basado en el hipervisor de XEN project pero con funcionalidades adicionales. Permite hacer una instalación sin necesidad de comprar una licencia (Express Edition). En la figura 4 podemos ver los componentes del hipervisor. Tenemos una capa de hardware y justo encima el hipervisor XEN. Luego tenemos el plano de control que es una máquina virtual Linux privilegiada basada en el sistema operativo CentOS 7.5 que ejecuta XAPI, un conjunto de herramientas que permiten gestionar el hipervisor en si[25], además también ejecuta los controladores de dispositivo. El plano de control puede comunicarse con el hipervisor para iniciar o parar máquinas virtuales invitadas. El plano de invitado o *guest domain* son las máquinas virtuales creadas por el usuario, las cuales piden recursos al plano de dominio.[26]

Ofrecen dos planes:

- Standard: \$800
- Premium: \$1525

3.1.6. Oracle VM Server

Se trata de un software gratuito, por lo que cumple uno de los requisitos marcados. La descarga del software requiere de registro.

Incorpora el hipervisor Xen[27] y es de código abierto bajo una licencia GPLv2[28].

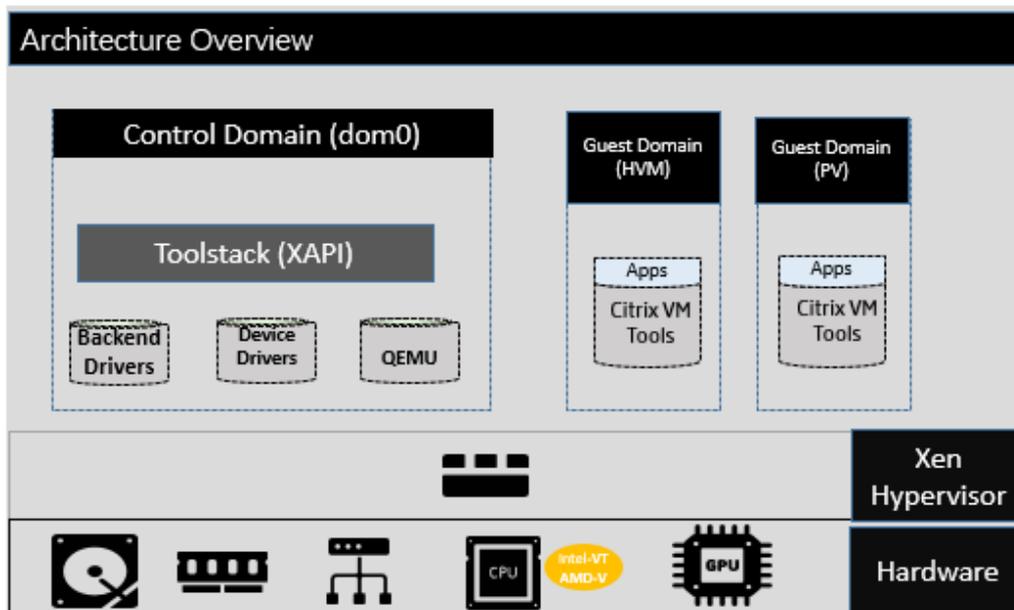


Figura 4: Componentes del hipervisor Citrix[26]

3.1.7. Virtuozzo

No es uno de los grandes líderes en virtualización pero ofrece una buena funcionalidad. Compite directamente con red-hat virtualization y plataformas *open-source* como Proxmox y KVM. Está basado en CentOS 7 KVM y una versión moderna de Qemu, que permite a máquinas virtuales ejecutarse sin cambios específicos del vendedor que puedan causar incompatibilidades. Permite unas 50 instancias de virtualización por servidor y su precio es de unos 990 dólares estadounidenses por mes por negocio.

3.2. Tecnologías candidatas

Entre las soluciones mencionadas hay tres de ellas que cumplen los requisitos impuestos para ser probadas. Citrix, Oracle VM server y Proxmox.

3.2.1. Citrix

Instalo el hipervisor en un equipo antiguo y una vez finalizada la instalación procedo a acceder por la interfaz web. Como se puede ver en la figura 5, aparece una página con información pero no permite la gestión directamente. Nos sugiere instalar XenCenter para ese propósito, programa únicamente disponible en el sistema operativo Windows.

Este requisito ya me causa un ligero rechazo hacia la solución porque implica uso de software propietario como es Windows. Procedo a la instalación de XenCenter en un sistema Windows y añadimos el nodo instalado en él para su gestión.

Una vez añadido nos aparece un aviso de que el nodo esta sin licencia, habría que ver si eso afecta al conjunto de funcionalidades de la herramienta.

Procedemos a crear una máquina virtual, es aquí donde aparece el primer problema importante, sólo permite obtener la imagen de instalación del sistema operativo de un DVD o de la red.

Para poder añadir una unidad de imágenes es necesario desde el host del hipervisor lanzar el siguiente comando:

```
# xe sr-create name-label=<NAME> type=iso device-config:legacy_mode=true
device-config:location=<ISODIR> content-type=iso
```

Donde <NAME> es el nombre que queremos dar a la unidad y <ISODIR> el directorio que montará esta unidad.

De esta forma ya nos permite elegir la unidad creada como repositorio de imágenes. Como última prueba intento hacer una migración en vivo de la máquina virtual que acabo de crear pero aparece un mensaje como el de la figura 6. En definitiva, las funciones reservadas para la versión sin licencia del entorno son demasiado limitadas y no cumple con los requisitos, tal y como ocurría con VMware ESXi.

Welcome to Citrix Hypervisor ® 8.2.0

Thank you for using Citrix Hypervisor. This page contains information about the Citrix Hypervisor 8.2.0 release.

Citrix Hypervisor

For information about Citrix Hypervisor 8.2.0 release, see the [Citrix Hypervisor Product Download](#) page.

For information about getting started with Citrix Hypervisor and for links to product documentation and other useful resources, see [Citrix Hypervisor Product Documentation](#).

To access Citrix Hypervisor Developer documentation, see [Citrix Developer Documentation](#).

XenCenter

XenCenter is the graphical Windows-based user interface for Citrix Hypervisor.

To download and install XenCenter, see the [Citrix Hypervisor 8.2.0 Product Download](#) page.

To access XenCenter documentation, see [XenCenter Product Documentation](#).

Copyright © 2009-2020 Citrix Systems, Inc. All rights reserved.

Citrix and Xen are registered trademarks. Citrix Hypervisor and XenCenter are trademarks of Citrix Systems, Inc. in the United States and other countries. All other product names, company names, marks, logos, and symbols are trademarks of their respective owners.

Figura 5: Página que se muestra al acceder por navegador al servidor con Citrix. Fuente: propia

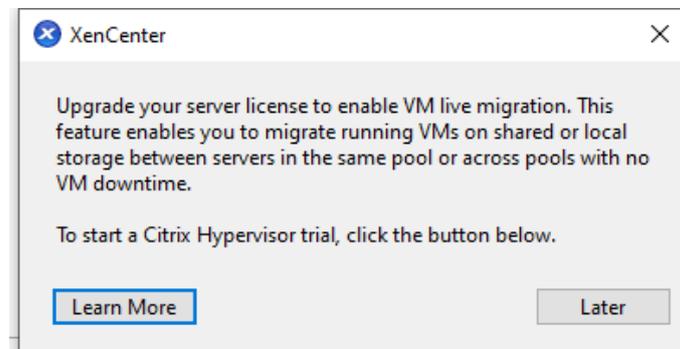


Figura 6: Mensaje de aviso. Fuente: propia

3.2.2. Oracle VM Server

He procedido a hacer una instalación para probarlo. Para poder acceder a su repositorio es necesario tener una cuenta Oracle. Primeramente he instalado Oracle VM server en una máquina virtual, lo que es un proceso bastante sencillo y rápido. Seguidamente he procedido a instalar Oracle VM manager, que controla el entorno de virtualización pero muestra algunos errores al iniciar el instalador. Debido a esto descarté esta opción al ser algo más compleja de instalar que el resto y por falta de tiempo.

3.2.3. Proxmox

Entre las tres opciones Proxmox es la que cumple todos los requisitos mencionados para poder ser instalado como entorno de virtualización en todo el CPD, por lo que es la tecnología escogida para ello. Procederé a entrar más en detalle sobre su instalación en la próxima sección.

4. Desarrollo

Entre las siete soluciones vistas la que más destaca es Proxmox. Sus funcionalidades gratuitas son suficientes para nuestras necesidades, a diferencia de las demás soluciones que vienen limitadas. Contamos con 8 nodos en producción y 3 en pruebas, en total 12. Lejos de los 32 que permite por *cluster*, dato que además nos implica un futuro crecimiento.

El hipervisor KVM es el líder en virtualización *open-source* para Linux y viene sin licencias. Creado en 2006 y con desarrollo activo desde entonces, cuenta con suficiente madurez para considerarlo un sistema estable. Es un hipervisor de tipo 1, es decir, se ejecuta directamente sobre la máquina sin ningún sistema operativo en medio[29]. Un hipervisor de tipo 2 se ejecutaría sobre un sistema operativo anfitrión. En la figura 7 se puede ver como se relaciona cada uno con el hardware.

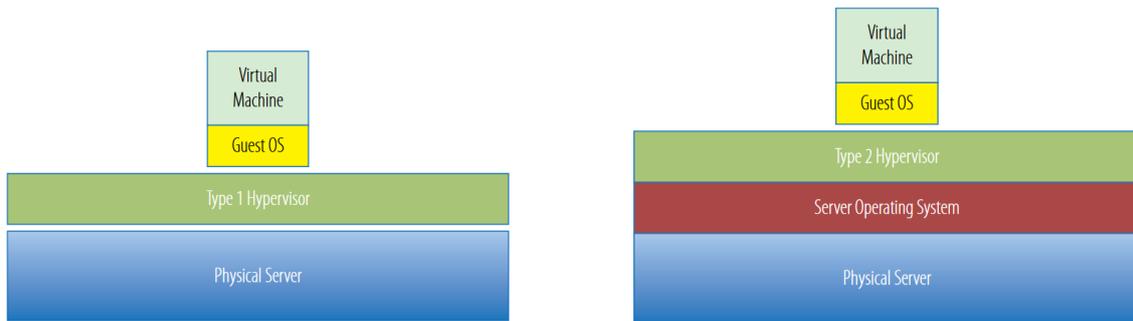


Figura 7: hipervisores de tipo 1 y 2. Fuente: Virtualization essentials (2016)[30, p.24-25]

4.1. Instalación de los nodos

Las pruebas las realizo en 3 máquinas distintas que actualmente se encuentran fuera de producción, las he bautizado como PXx, donde la última x es el número de nodo. Sus especificaciones se muestran en el cuadro 4.

La instalación del software es sencilla y fácil de seguir según la guía[31]. Se instala cada nodo independientemente y luego se forma el *cluster*. La gestión de los nodos se puede realizar mediante web mediante la dirección **IP** del nodo y el puerto especificado, por defecto se usa el puerto 8006:

```
https://<IP_NODO>:8006
```

Además se puede hacer una gestión más específica mediante comandos conectando al servidor por protocolo SSH (puerto 22). El siguiente paso es crear un *cluster* a partir de todos los nodos instalados. Se añaden mediante claves, las cuales se encuentran en cada uno de los nodos, en la figura 8 se puede ver la información de unión de px1, la cual se usa para añadirlo mediante la interfaz web, cada nodo tiene una información distinta.

Nodo	Modelo	CPU	Memoria
PX1	Dell PowerEdge R610	2 Intel Xeon Quad E5504 2.00Ghz	24GB
PX2	Dell PowerEdge R610	2 Intel Xeon Quad E5420 2.50Ghz	32GB
PX3	Dell PowerEdge R630	2 Intel® Xeon® E5-2609 v3 1,9GHz	64GB

Cuadro 4: Características de los nodos de pruebas.

También es posible crear la unión mediante la interfaz de comandos mediante el comando:

```
# pvecm add IP-NODO-DEL-CLUSTER
```

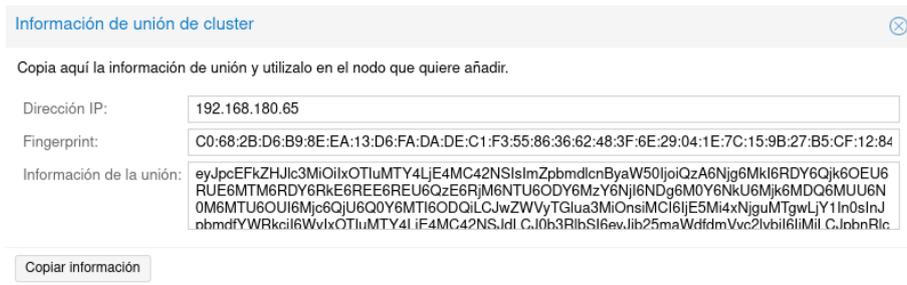


Figura 8: Información de unión de px1. Fuente: propia

Podemos ver la información de los nodos:

```
root@px1:~# pvecm nodes
Membership information
```

Nodeid	Votes	Name
1	1	px1 (local)
2	1	px2
3	1	px3

Una vez formado el *cluster* se puede gestionar cada nodo indistintamente desde cualquier otro nodo de éste mediante la interfaz web. Con esto ya tenemos dos objetivos cubiertos, por una parte conseguimos la configuración de un *cluster* virtual y por otra la facilidad de la gestión. Hasta ahora ésta también era por web pero solo permite gestionar el nodo al que te conectas. El último detalle es cambiar el repositorio de *enterprise* por el de *community* en cada uno de los nodos. Es importante para poder tener el sistema actualizado sin necesidad de contratar un plan específico. En cada uno de los nodos debemos comentar o eliminar la siguiente línea en la ruta `/etc/apt/sources.list.d/pve-enterprise.list` :

```
deb https://enterprise.proxmox.com/debian/pve bullseye pve-enterprise
```

y en `/etc/apt/sources.list` tenemos que habilitar el repositorio de no suscripción:

```
deb http://ftp.debian.org/debian bullseye main contrib
deb http://ftp.debian.org/debian bullseye-updates main contrib
# PVE pve-no-subscription repository provided by proxmox.com,
deb http://download.proxmox.com/debian/pve bullseye pve-no-subscription
# security updates
deb http://security.debian.org/debian-security bullseye-security main contrib
```

4.2. Redundancia

El sistema actual tiene un cierto grado de redundancia pero no es suficiente porque como hemos explicado el *downtime* asociado a un desastre es demasiado alto y no es un buen diseño en general. En esta sección se discuten ciertas funcionalidades a usar en el nuevo sistema.

4.2.1. Live migration

Live migration o migración en vivo permite cambiar el nodo de ejecución de una máquina virtual sin que haya prácticamente *downtime* del servicio en cuestión. Es una herramienta útil para hacer paradas programadas o reorganizaciones. La instalación actual de VMware permite migrar máquinas virtuales, aunque siguiendo un método distinto que nos añade tiempo de caída. Para las máquinas virtuales basadas en KVM se utiliza la técnica de *pre-copy memory migration*, su funcionamiento se puede ver en la figura 9.

En el primer paso se intentan copiar todas las páginas de memoria desde la fuente hasta el destino mientras la máquina está corriendo. Si hay algún cambio en las páginas de origen éstas se marcan como no válidas y se volverán a copiar al final tantas veces hasta que el ratio de páginas copiadas dos o más veces es mayor a páginas no válidas (asegurando solamente cambios menores). Luego cambia de fase apagando la máquina original y copiando las últimas páginas no válidas, este pequeño *downtime* es del orden de milisegundos o segundos. Una vez copiadas las páginas la máquina reanuda la ejecución en el nuevo nodo[12, p.37-39].

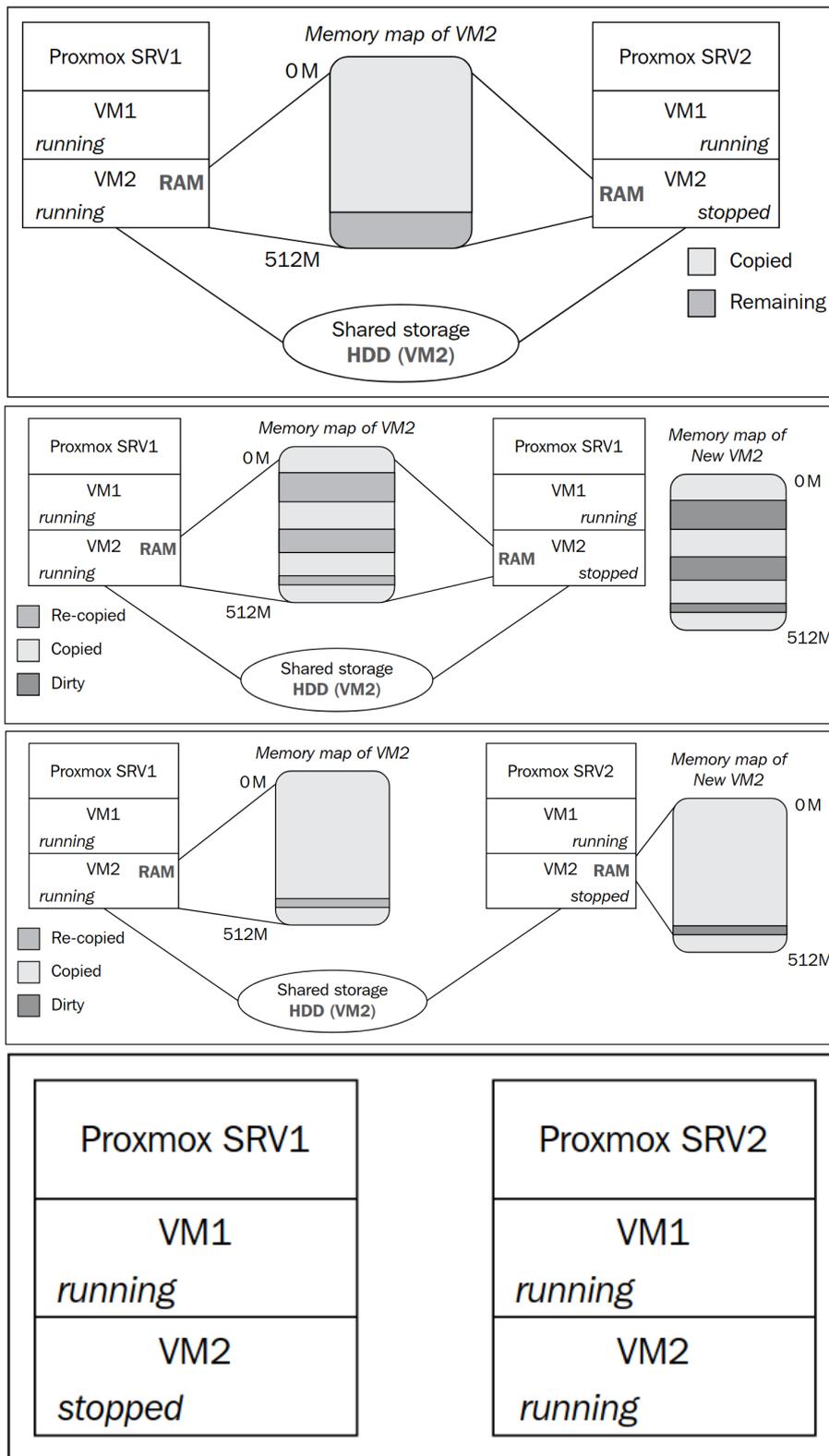


Figura 9: Pasos en una migración *pre-copy*. Fuente: Proxmox High-Availability

4.2.2. Shared storage

Con la instalación principal cada uno de los nodos tiene su propio almacenamiento y cada máquina virtual se almacena en el propio nodo. Podría encajar como solución si cada uno de los nodos contase con un sistema de RAID el cual pudiese mantener los datos intactos en caso de fallo de disco, por otra parte para ajustarlo mejor al paradigma actual decido aplicar la solución de almacenamiento compartido tal y como ha estado hasta ahora excepto que ya no habrá separación en volúmenes virtuales sino que todo el conjunto de máquinas virtuales irá en el mismo. Hacerlo

de esta manera comporta poder tener acceso a todas las máquinas virtuales desde cualquier nodo de forma casi inmediata sin necesidad de copiar información entre volúmenes. En la figura 10 se ve como funciona este principio, cada máquina física mediante red conecta con un almacenamiento en el cual están almacenadas las máquinas virtuales. Pongo en marcha una cabina de discos casera

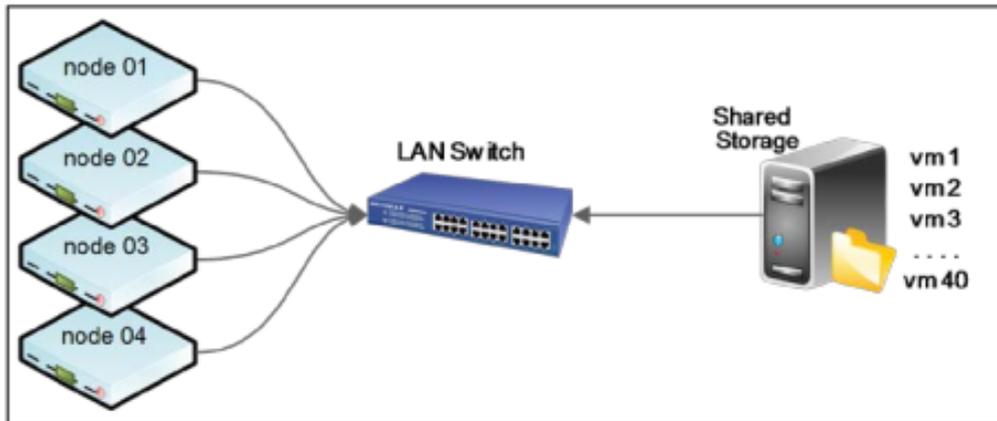


Figura 10: Almacenamiento compartido entre los nodos Fuente: Mastering Proxmox[32]

hecha con un PC antiguo con un procesador Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz y 12 GB de memoria RAM, además lo he equipado con 4 discos de 1 TB para el almacenamiento y un disco de 320 GB para el SO, el cual será un TrueNAS-12.0-U5.1, que también es de código abierto. Bautizo esta cabina como el nodo 0, "px0". Los 4 discos de almacenamiento están configurados con RAIDZ1, similar a RAID5, con un bloque de paridad por franja. Ambos obtienen el mismo nivel de redundancia, tolerando un disco fallido. El sistema de RAID escogido no es importante para esta prueba.

Ahora solo queda añadir el almacenamiento al *cluster* en el centro de datos apartado almacenamiento. Añadiremos el almacenamiento por red, siguiendo el protocolo NFS. Luego en el *cluster* lo añadimos como una unidad NFS. Tenemos que configurarlo de tal forma que nos permita guardar discos virtuales, también podemos añadir otros volúmenes para dar usabilidad que nos permita guardar imágenes **iso** por ejemplo para hacer instalaciones, de esta forma podemos acceder a ellas desde cualquier nodo.

Estado del almacenamiento en el primer nodo:

```
root@px1:~# pvesm status
Name          Type      Status    Total      Used      Available  %
CabinaTN      nfs       active    2734440576 15576064 2718864512 0.57%
Images        nfs       active    2723902208 5037696  2718864512 0.18%
local         dir       active    71154840   9831008  57663648  13.82%
local-lvm     lvmthin  active    189976576 0        189976576 0.00%
```

Vemos dos unidades por NFS, CabinaTN e Images. Ambas unidades pertenecen a la cabina de discos, la primera almacena las máquinas virtuales y la segunda imágenes de instalación.

4.2.3. High-Availability

Proxmox HA(High-Availability), literalmente alta disponibilidad, permite mover o migrar máquinas virtuales de un nodo a otro sin necesidad de una interacción por parte del usuario. Esto nos reduce el tiempo de *downtime* drásticamente al no requerir de una presencia humana. Actualmente un fallo de servidor nos puede dar un caída del servicio de más de 10 horas si no se actúa a tiempo, por ejemplo en el caso que un servicio caiga justo después de terminar una jornada laboral. La fórmula para calcular la disponibilidad es la siguiente:[12, p.30]

$$A = 100 * (EU - DD)/EU$$

Donde A es la disponibilidad, EU el tiempo de funcionamiento esperado y DD el tiempo de *downtime*. Simplemente con el apagón de mantenimiento anual que dura aproximadamente 24 horas tenemos una disponibilidad del 99,72%. Teniendo una caída de servidor de 10 horas la disponibilidad bajaría al 99,61%.

Tenemos una serie de requisitos antes de activar la herramienta de HA:[33, p.313]

- Al menos 3 nodos en el *cluster*
 - Para tener un *quorum* fiable. *Quorum* es el mínimo número de votos requeridos para realizar una operación en el *cluster*. Con solo dos nodos el voto se mantiene 1:1 así que no es posible el *quorum*. Este problema y su solución se puede ver en la figura 11.
- Almacenamiento compartido para las MV.
- Redundancia de Hardware y componentes fiables
 - El HA se debe usar como último recurso, primeramente hay que eliminar cualquier *single point of failure* de la instalación.

La unidad principal manejada por ha-manager se llama recurso, éste está únicamente identificado por un identificador de servicio que consiste en el tipo de recurso y su ID específico, por ejemplo vm:100 es la máquina virtual con ID 100[33, p.314]. Con el siguiente comando podemos ver el estado actual del HA:

```
root@px1:~# ha-manager status
quorum OK
master px1 (active, Thu Dec 16 09:03:47 2021)
lrm px1 (active, Thu Dec 16 09:03:46 2021)
lrm px2 (idle, Thu Dec 16 09:03:50 2021)
lrm px3 (idle, Thu Dec 16 09:03:51 2021)
service vm:100 (px1, started)
```

Se puede habilitar el HA para un recurso tanto desde la interfaz web como en la línea de comandos con el comando:

```
# ha-manager add vm:100
```

La pila de HA entonces intenta empezar los recursos y mantenerlos ejecutándose. Esta pila funciona de manera asíncrona y tarda algo de tiempo en ver los resultados de las acciones ya que debe comunicarse con los demás nodos. Se pueden definir grupos de nodos, de esta forma puedes restringir algunos recursos a ejecutarse solamente en miembros de ese grupo. Vamos a probar la herramienta con una máquina virtual la cual aloja una web operativa sobre un sistema operativo Ubuntu 20 server. Ahora configuramos el HA para proteger esta máquina virtual; primero de todo se crea un grupo con el *cluster* de nodos al completo, ya que no nos interesa en este momento separarlo. Este grupo nos define el *lance* de esta migración. Después añadimos el recurso (la máquina virtual con id 100) y lo asociamos al grupo que acabamos de crear. El siguiente paso es simular un fallo en el nodo px1 apagándolo repentinamente. El resultado esperado es que la máquina virtual debería pasarse a ejecutar en otro nodo. Tras unos pocos minutos vemos como la máquina se migra correctamente y se inicia. Con este procedimiento tenemos un tiempo de *downtime* de unos pocos minutos.

El funcionamiento es sencillo [34], para proveer el HA dos **daemons** se ejecutan en cada nodo:

- *Local resource manager* (pve-ha-lrm), controla los servicios que se ejecutan en el nodo local.
- *Cluster resource manager* (pve-ha-crm), hace las decisiones del *cluster*. Envía comandos al LRM, procesa los resultados y mueve recursos a otros nodos si algo falla. El CRM también se encarga del *fencing*.

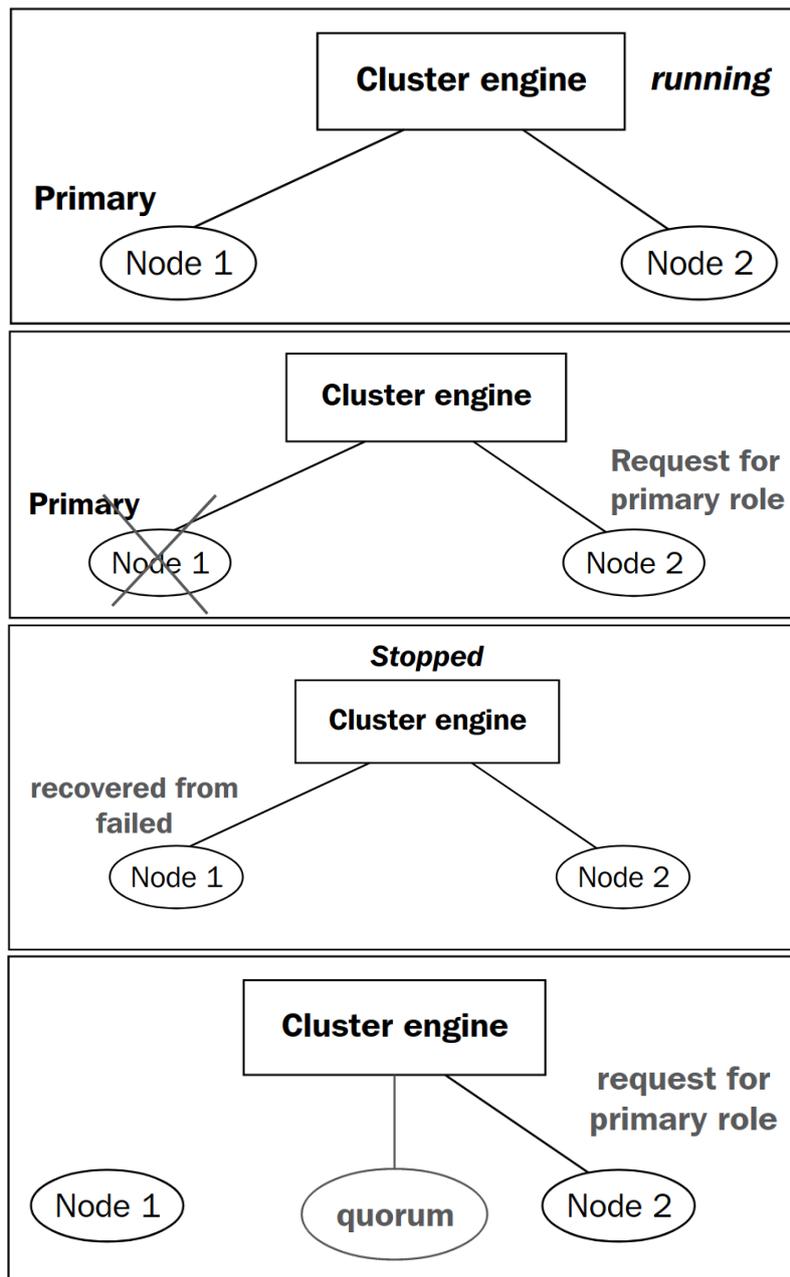


Figura 11: Con un *cluster* de dos nodos, si uno cae, el otro tomara su posición para ser el líder, aunque no tiene éxito ya que se necesitan dos nodos para cambiar de rol así que el *cluster* no será funcional hasta que el primer nodo vuelva a estar operativo. Cuando éste vuelve intenta unirse al *cluster* falla debido a que el *cluster* no está funcionando. Para resolver esto se recomienda añadir el tercer nodo. Fuente: Proxmox High Availability

El *fencing* asegura que en el caso de fallo de nodo, el nodo fallido es garantizado que realmente está *offline*. Esto es requerido para asegurarse que ningún recurso se ejecuta por duplicado al ser recuperado en otro nodo.

El CRM mantiene un estado en los servicios:

```
service vm:100 (px1, started)
```

Los estados más comunes:

- **Started:** El servicio está activo y el LRM debería iniciarlo.
- **Starting:** El CRM todavía no tiene confirmación por parte del LRM de que el servicio se está ejecutando.
- **Stopping:** El CRM todavía no tiene confirmación por parte del LRM de que el servicio se debe parar.

- Stopped: El servicio está parado y confirmado por LRM.
- Fence: Esperar a que el nodo haga *fencing* ya que ahora el nodo del servicio no se encuentra en el *cluster*, en cuanto el nodo termine el proceso, el servicio será puesto en modo *recovery*.
- Recovery: Esperar a que el servicio se recupere, el HA manager intenta buscar un nuevo nodo donde ejecutar el servicio.

Un ejemplo, imaginemos que en el nodo px3 tenemos una máquina virtual con id 100 y este nodo cae. Tenemos px1 y px2 disponibles y finalmente se elige px1 para migrarla, seguiría estos estados:

```
service vm:100 (px3, started) -> service vm:100 (px3, fence) ->  
-> service vm:100 (px3, recovery) -> service vm:100 (px1, starting) ->  
-> service vm:100 (px1, started)
```

4.3. Migración desde VMware

El propósito de este proyecto es renovar todo el sistema de virtualización así que es necesario conservar las antiguas máquinas. El método utilizado consiste en obtener la máquina virtual original en formato OVF y el disco o los discos virtuales. El formato OVF es un estándar abierto, seguro, portable, eficiente y extensible para el empaquetamiento y la distribución de software para ejecutarse en máquinas virtuales que no depende de un sistema propietario[30, p.51-52]. Con este archivo podemos definir la nueva máquina virtual a crear en el sistema Proxmox y enlazar con el disco correspondiente.

Procedo a migrar una máquina virtual de uno de los nodos de VMware. Esta contiene dos discos virtuales, de 16 y 10 GB, 2 GB de RAM y 1 vCPU. Obtenemos el OVF y los discos mediante la interfaz web de VMware. Desde el menú de máquinas virtuales elegimos la máquina virtual y le damos a exportar.

Esto descarga 3 ficheros; <nombre_mv>.ovf, disk-0.vmdk y disk-1.vmdk, dónde <nombre_mv>es el nombre de la máquina virtual. Esto lleva un tiempo según el tamaño de los discos y cuantos tenga.

Una vez lo tenemos hay que moverlo al nodo Proxmox donde lo queramos ejecutar o directamente a la cabina. Lo más importante es que desde ese mismo nodo ejecutemos el comando siguiente:

```
# qm importovf <id_mv> ./<nombre_ovf>.ovf <storage> --format <format>
```

Dónde:

<id_mv>es el número identificador que tendrá la nueva máquina virtual.

<nombre_ovf>es el archivo que nos hemos descargado con extensión ovf.

<storage>es el almacenamiento dónde queremos que se guarde.

<format>es el formato de disco que queremos, puede ser qcow2, vmdk o raw si no se especifica nada.

En el formato raw el espacio de disco es asignado durante la creación y usará el tamaño especificado[12, p.24]. El formato qcow2 es una versión mejorada del qcow, el cual permite crecer según se vayan añadiendo datos. Permite tamaños más pequeños que el formato raw al no asignar todo el disco al inicio[35]. El formato vmdk se usa generalmente en productos de VMware.

He procedido a migrar distintas máquinas virtuales con distintos sistemas operativos. Con una máquina Ubuntu el resultado es satisfactorio al seguir los pasos anteriormente mencionados. En nuestro entorno también contamos con máquinas CentOS y Windows. Al seguir los mismos pasos, el sistema operativo no arranca en el nodo Proxmox. Para solucionarlo, previamente desde el anterior hipervisor hay que cambiar la controladora de disco a IDE[36].

De cara a una migración masiva, se puede realizar nodo por nodo y a medida que un nodo VMware tenga todas las máquinas migradas al *cluster* de pruebas, el cual acabará entrando en producción, se instalaría Proxmox VE y se uniría al *cluster* como un nodo más. Esta migración conlleva tiempo y hacerlo todo a la vez supondría un tiempo de *downtime* muy alto así que es más conveniente parar los servicios de manera escalada.

5. Contenedores

Un contenedor, a diferencia de una máquina virtual que virtualiza recursos del host, sólo virtualiza las capas de software por encima del sistema operativo. Son paquetes de software ligeros que contienen todas las dependencias para ejecutar una aplicación o servicio, en la figura 12 se puede ver la diferencia con una máquina virtual convencional. Dentro de los contenedores de Proxmox solamente se pueden ejecutar distribuciones de Linux ya que usa Linux Containers (LXC) como tecnología. Hay el Proxmox container toolkit (pct) que simplifica el uso y manejo de LXC y hay un conjunto de imágenes (*templates*) disponibles. La única prueba que he podido hacer ha sido descargar e iniciar una imagen de Debian 11 y funciona exitosamente.

Esta es una tecnología a tener en cuenta de cara al futuro. Actualmente el foco de atención está en las máquinas virtuales y su correcta migración al nuevo sistema. Se podría considerar como el siguiente paso a la hora de actualizar algunos de los servicios ofrecidos, manteniendo un sistema mixto ya que no todas las máquinas virtuales cuentan con un sistema Linux instalado. El coste de tiempo sería demasiado alto si hacemos ambas tareas a la vez.

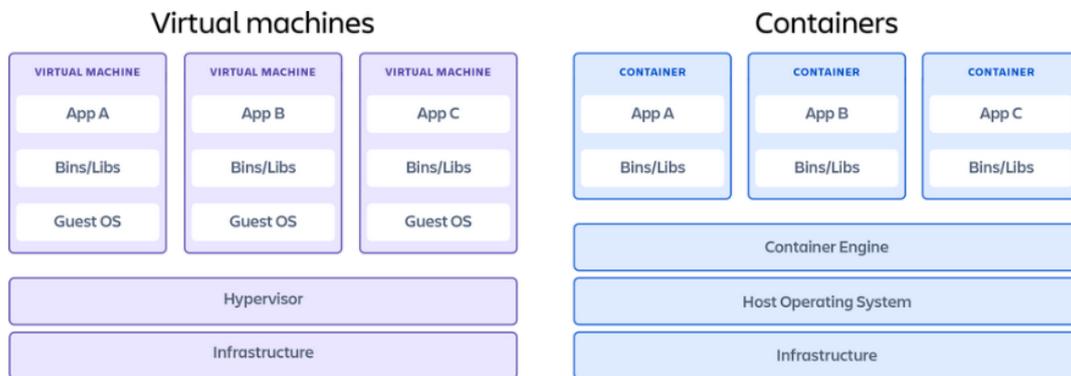


Figura 12: Diferencia entre máquina virtual (izquierda) y contenedor (derecha). Fuente: [37]

6. Conclusiones técnicas

Las soluciones tratadas en este proyecto van a permitir tener un sistema lo suficiente capaz y robusto para funcionar con una alta disponibilidad y con mucha tolerancia a fallos. Hemos cumplido gran parte de los objetivos: Primero hemos solucionado el problema del almacenamiento, con el nuevo paradigma, eliminamos la separación lógica de volúmenes y no se desaprovecha espacio, todo va almacenado en un único volumen. Segundo hemos solucionado el problema del *downtime*, reduciéndolo drásticamente mediante la solución de HA que nos automatiza el proceso de migración en caso de caída de nodo y con el almacenamiento compartido y la facilidad de la migración de máquinas virtuales para mantenimientos programados. Además de estas soluciones hay muchas más capacidades de la tecnología Proxmox que podemos explotar en nuestro beneficio como los contenedores, los cuales nos podrían brindar ventajas en según que tipos de servicios.

En cuanto a la solución escogida, se adapta perfectamente a nuestras necesidades y requisitos. La envergadura del CPD es suficientemente pequeña como para optar por una solución de código abierto y no gastar extra en licencias pensadas para instalaciones a gran escala o más específicas. Además ha supuesto una instalación y configuración más simple que otras tecnologías candidatas. La migración total de servidores, uno de los objetivos marcados, no se ha podido realizar pero se ha planteado para poderse hacer en el corto plazo.

7. Gestión del proyecto

7.1. Planificación temporal

La planificación temporal de mi trabajo de fin de grado comenzó el verano pasado. Primeramente me planteé varios temas sobre los que trabajar. Todos ellos eran tareas y problemas que subsanar en la empresa donde trabajo como administrador de sistemas. Finalmente me decidí sobre la renovación del sistema de virtualización que tenemos instalado en nuestro pequeño centro de datos. Así pues el reparto de tareas está pensado sobre mi jornada laboral, la cual dedico una parte del tiempo en el proyecto. Más allá de mi jornada dedico tiempo a la parte de documentación. En el gráfico Gantt presentado expongo las tareas por intervalos de tiempo de dedicación y las clasifico con dos colores según sean tareas principales o sub-tareas de éstas. Las tareas programadas para finales de octubre cuentan con gran riesgo debido a que se depende enteramente de la situación de la empresa. Finalmente he acabado por describir un plan de acción a llevar a cabo debido a que no se ha podido realizar a tiempo, este plan consiste en migrar de forma escalada las máquinas virtuales, ganando nodos en el *cluster* poco a poco. Como era de esperar

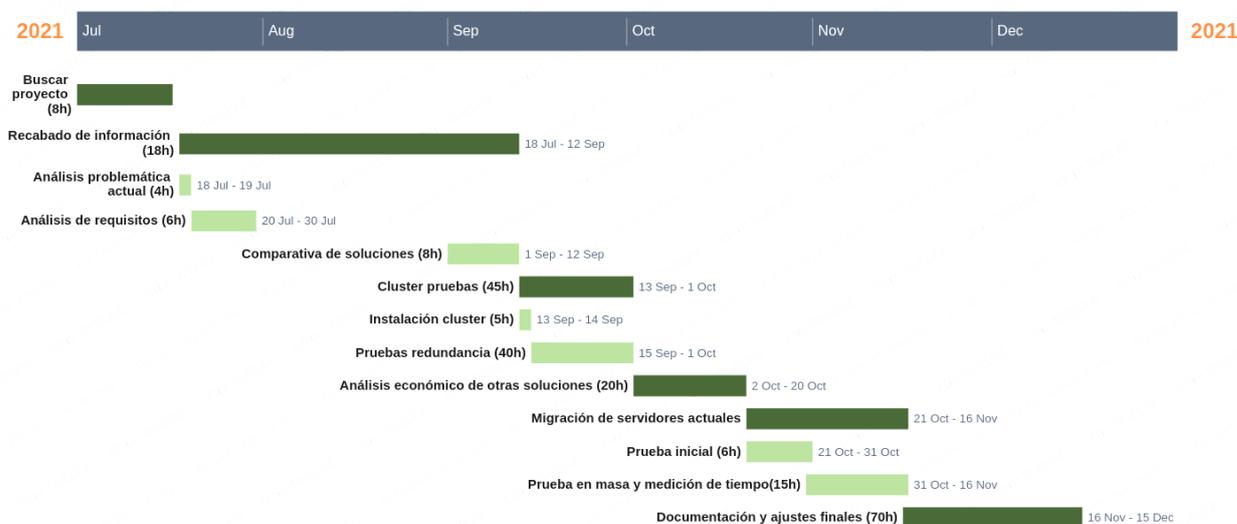


Figura 13: Gantt del proyecto. Fuente: Propia

esta planificación ha ido sufriendo cambios de última hora como el mencionado anteriormente o tareas adicionales que se han ido tratando que no están reflejadas. Además muchas de las tareas que sí fueron previstas han podido ir variando en tiempo de dedicación.

7.2. Gestión económica

Hay diversos agentes a tener en cuenta en el estudio económico de este proyecto. Como gasto humano estoy yo, ya que una parte de mi jornada va dedicada a esto. Supondría un gasto de 161 horas de trabajo, unos 2000€.

El gasto en equipo, pese a que cuento con un conjunto de máquinas ya instaladas tanto en producción como en pruebas, tomaré como referencia el modelo actual de compra, el cual es un Dell PowerEdge R540 con un precio de 5354,12 €. Este es el equipo que se comprará para futuras ampliaciones así que en el caso que se necesitase crear un CPD desde cero, es la estimación más exacta.

Para el *cluster* de prueba estoy usando 3 equipos, un total de 16062,36 € y una cabina de discos rudimentaria de 4 TB de capacidad con RAID1, un NAS (similar a SAN) como el Synology DS220+NAS tiene un precio de unos 400 €. Una cabina de discos que sí que entraría en producción como DELL Unity 380 tiene un coste de 41146,97€ Para interconectar los servidores se están usando 3 switches con un precio de 2922,50€ cada uno. El cluster de servidores finalmente ascenderá a un valor de 42832,96 €.

Coste total: 110.809,79 €

Por fortuna, gran parte del hardware está siendo reutilizado y no es necesario renovarlo a corto plazo, el precio es orientativo y da una idea del coste de un proyecto de esta escala desde cero.

8. Sostenibilidad y compromiso social

Pese a ser una persona bastante concienciada con la crisis climática, todo lo que sé es para uso fuera de mi ámbito profesional y aunque puedo reconocer fácilmente en el día a día que soluciones pueden suponer menos gasto energético o menos cantidad de hardware siendo usado, mi desconocimiento de leyes, estándares y prácticas me impiden pensar a una escala global o estimar exactamente el impacto de mis acciones.

Como he comentado en el punto anterior, el desempeño profesional que llevo a cabo actualmente no supone un gran impacto ambiental más allá de poder influir en pequeñas fluctuaciones sobre el gasto energético que se utiliza o la cantidad de hardware que se está utilizando. Lo más probable es que una parte de nuestros servidores esté siendo infrautilizada y se podría dar el mismo servicio con un gasto energético y de hardware menor en algunos aspectos. De ahí que vea tan necesaria la renovación de al menos uno de los sistemas del centro de datos. También se podría mejorar otros aspectos como la refrigeración pero ahí no entraré.

Intentaré en la medida de lo posible usar las métricas oportunas y los estándares disponibles para medir y cumplir con las normativas necesarias en las decisiones que tome.

8.1. Matriz de sostenibilidad

8.1.1. Ambiental

El impacto ambiental de mi proyecto probablemente sea invariable. La única forma de reducirlo sería disminuir el gasto energético de las máquinas. Por otra parte, al tener que reutilizar hardware, probablemente consiga mejores resultados con el mismo impacto anterior, lo que se podría traducir en un menor impacto en general ya que tenemos un margen de crecimiento sin tener que aumentar la huella.

Actualmente el problema se resuelve usando las máquinas disponibles a mayor capacidad. Una sola máquina puede hacer perfectamente el trabajo que actualmente pueden hacer dos. De esta forma se obtiene una máquina perfectamente operativa para reutilizarse en cualquier otra tarea.

8.1.2. Social

La necesidad del proyecto viene dada por la escasez de material y el derroche de éste hasta ahora, esto comportaba que hubiese un sistema algo caótico y difícil de gestionar. La simplicidad que comporta este proyecto ayudará a ahorrar muchas horas de gestión al departamento de sistemas y evitar problemas inesperados.

Principalmente este proyecto me va a aportar experiencia y una gran influencia en mi puesto de trabajo por lo que a parte de suponer otro paso más en el grado que curso también es un paso más en mi carrera profesional.

9. Conclusiones

El objetivo principal de este proyecto era poder plantear una alternativa al entorno de virtualización actual el cual recordemos presenta una serie de problemas como su antigüedad, problemas con el almacenamiento y la redundancia. Se ha podido plantear satisfactoriamente una alternativa válida a estos problemas, en definitiva, el sistema propuesto sienta las bases para poder sustituirlo completamente, resolviendo los problemas que acarrea y añadiendo nuevas funcionalidades que facilitan la gestión.

9.1. Largo plazo

Uno de los problemas más notorios de la instalación anterior es el bajo mantenimiento que ha sufrido con los años. La separación por volúmenes virtuales ha forzado a cada nodo a trabajar con un espacio concreto y lo que podía tener sentido al principio, debido al crecimiento del CPD se ha perdido. Ahora con una sola unidad almacenando todos los discos virtuales no tenemos que preocuparnos por ese aspecto, ya que el *cluster* puede ir transformándose de muchas maneras sin verse afectada la gestión del almacenamiento.

El hipervisor instalado es el 6.5 de VMware, con la última actualización en 2016. Para el nuevo sistema necesitamos tener programada una fecha en la que actualizar el hipervisor, como podría ser una vez al año durante el mantenimiento general. La actualización de Proxmox es sencilla, una vez editados los repositorios solo hay que ejecutar:

```
root@px1 ~# apt-get update
root@px1 ~# apt-get dist-upgrade
```

El mayor aprovechamiento del hardware nos da margen de crecimiento suficiente. Además se nos brinda la oportunidad de añadir nuevas tecnologías como contenedores.

Referencias

- [1] VMware. *Aprovisionamiento fino de discos virtuales*. URL: <https://docs.vmware.com/es/VMware-vSphere/7.0/com.vmware.vsphere.storage.doc/GUID-8204A8D7-25B6-4DE2-A227-408C158A31DE.html> (visitado 15-09-2021).
- [2] Microsoft Azure. *What is a container?* URL: <https://azure.microsoft.com/es-es/overview/what-is-a-container/#overview> (visitado 15-09-2021).
- [3] Cisco. *What Is a Data Center?* URL: <https://www.cisco.com/c/en/us/solutions/data-center-virtualization/what-is-a-data-center.html> (visitado 15-09-2021).
- [4] VMware. *What is a hypervisor?* URL: <https://www.vmware.com/topics/glossary/content/hypervisor> (visitado 15-09-2021).
- [5] Randy H. Katz David A. Patterson Garth Gibson. "A Case for Redundant Arrays of In-expensive Disks". En: (1988). URL: <https://www2.eecs.berkeley.edu/Pubs/TechRpts/1987/CSD-87-391.pdf>.
- [6] VMware. *What is SAN and how does it work?* URL: <https://www.vmware.com/topics/glossary/content/storage-area-network-san> (visitado 15-09-2021).
- [7] Microsoft Azure. *What is virtualization?* URL: <https://azure.microsoft.com/es-es/overview/what-is-virtualization/> (visitado 10-2021).
- [8] VMware. *vSphere*. URL: <https://www.vmware.com/products/vsphere.html>.
- [9] Oracle. *Brief History of Virtualization*. 2011. URL: https://docs.oracle.com/cd/E26996_01/E18549/html/VMUSG1010.html (visitado 11-2021).
- [10] Dr. Anders S.G. Andrae. "Total Consumer Power Consumption Forecast". En: 2017. URL: https://www.researchgate.net/publication/320225452_Total_Consumer_Power_Consumption_Forecast/link/59d5ee16aca2725954c78d15/download.
- [11] Bernard Golden. *Virtualization for dummies 3rd edition*. 2011. ISBN: 978-0-470-94331-1.
- [12] Simon M.C. Cheng. *Proxmox High Availability*. 2014. ISBN: 978-1-78398-088-8.
- [13] VMware. "End of General Support for vSphere 6.0 (66977)". En: (). URL: <https://kb.vmware.com/s/article/66977> (visitado 08-2021).
- [14] VMware. *VMware ESXi 6.0, Patch Release ESXi600-202002001*. 2020. URL: <https://docs.vmware.com/en/VMware-vSphere/6.0/rn/esxi600-202002001.html> (visitado 08-2021).
- [15] Sam Ingalls. *Best Server Virtualization Software for 2021*. 2021. URL: <https://www.serverwatch.com/best-server-virtualization-software/> (visitado 21-09-2021).
- [16] Drew Robb. *VMware vSphere: Software Features Price*. 2019. URL: <https://www.serverwatch.com/virtualization/vmware-vsphere-virtualization-software-overview-and-insight/> (visitado 21-09-2021).
- [17] VMware. *VMware vSphere Version Comparison*. URL: <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/vsphere/vmw-version-comparison.pdf> (visitado 20-10-2021).
- [18] VMware. *VMware vSphere Essentials Kit*. URL: <https://store-us.vmware.com/vmware-vsphere-essentials-kit-282883900.html> (visitado 10-2021).
- [19] Proxmox. *Proxmox Virtual Environment. Compute, network, and storage in a single solution*. URL: <https://www.proxmox.com/en/proxmox-ve>.
- [20] Proxmox. *Support Subscriptions*. URL: <https://www.proxmox.com/en/proxmox-ve/pricing> (visitado 10-2021).
- [21] Drew Robb. *Proxmox VE*. 2019. URL: <https://www.serverwatch.com/virtualization/proxmox-ve/> (visitado 21-09-2021).
- [22] Drew Robb. *Red Hat Virtualization: Features Pricing*. 2019. URL: <https://www.serverwatch.com/guides/red-hat-virtualization-product-overview-and-insight/> (visitado 22-10-2021).
- [23] Red Hat. *Virtualization limits for Red Hat Enterprise Linux with KVM*. URL: <https://access.redhat.com/articles/rhel-kvm-limits> (visitado 22-10-2021).
- [24] Drew Robb. *Microsoft Hyper V Review*. 2020. URL: <https://www.serverwatch.com/virtualization/microsoft-virtualization/> (visitado 22-10-2021).

- [25] Citrix. *XAPI*. URL: <https://xenproject.org/developers/teams/xen-api/> (visitado 01-12-2021).
- [26] Citrix. *Technical overview*. 2021. URL: <https://docs.citrix.com/en-us/citrix-hypervisor/technical-overview.html> (visitado 02-12-2021).
- [27] Oracle. *Oracle Concepts Guide for Release 3.3*. URL: https://docs.oracle.com/cd/E50245_01/E50249/html/vmcon-ovm-server-hypervisor-xen.html (visitado 27-10-2021).
- [28] Kyle Guercio. *Oracle VM VirtualBox Review*. 2020. URL: <https://www.serverwatch.com/server-reviews/oracle-vm-virtualbox-review/> (visitado 07-01-2022).
- [29] Tytus Kurek. *KVM hypervisor: a beginners' guide*. 2021.
- [30] Matthew Portnoy. *Virtualization essentials Second Edition*. 2016. ISBN: 978-1-119-26772-0.
- [31] Proxmox. *Installation*. 2021. URL: <https://pve.proxmox.com/wiki/Installation> (visitado 28-10-2021).
- [32] Wasim Ahmed. "Mastering Proxmox". En: 2014. ISBN: 978-1-78398-082-6.
- [33] Proxmox Server Solutions Gmbh. *Proxmox VE Administration Guide 7.1*. 2021. Cap. 15.
- [34] Proxmox. https://pve.proxmox.com/wiki/High_Availability. 2021. (Visitado 02-09-2021).
- [35] Cisco. *The QCOW Image Format*. 2006. URL: <https://people.gnome.org/~markmc/qcow-image-format-version-1.html> (visitado 02-12-2021).
- [36] Proxmox. *Migration of servers to Proxmox*. 2021. URL: https://pve.proxmox.com/wiki/Migration_of_servers_to_Proxmox_VE#VMware (visitado 29-09-2021).
- [37] Ian Buchanan. *Comparación de contenedores y máquinas virtuales*. URL: <https://www.atlassian.com/es/continuous-delivery/microservices/containers-vs-vms> (visitado 01-07-2022).